2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 4-7 October 2016, Alcalá de Henares, Spain

Fitting for Smoothing: A Methodology for Continuous-Time Target Track Estimation

Tiancheng Li[†], Javier Prieto[†‡], Juan Manuel Corchado[†]
[†]BISITE Research Group, University of Salamanca (USAL), Salamanca, Spain
[‡]R&D Department, StageMotion, Palencia, Spain
{t.c.li, javierp, corchado}@usal.es

***Abstract*—A preliminary framework for inferring *continuous-time* target trajectory (namely "track") is given for a class of target tracking problems in which the target is subject to a rather smooth evolving process in time series, such as tracking passenger aircrafts or ships that have scheduled routes. As the core idea, the distant estimates given over time by a recursive estimator are 'fitted' by using a *function of continuous-time*, which can be then used to infer the state for any time instants in the effective fitting period, either the past (like conventional smoothing, but curried out online) or the future (including long-term prediction). This regression analysis methodology, referred to as *fitting for smoothing* (F4S), also facilitates** combating misdetection **and** outliers **from which most existing tracking systems suffer. Simulations are provided to illustrate how it works and benefits in either cluttered or non-cluttered environments, with either a single target or multiple targets.**

Keywords—Track estimation; target tracking; regression; time series fitting; long-term prediction

I. INTRODUCTION

Dynamic state estimation is concerned with the sequential process of estimating the state(s) (of one or multiple targets of interest) evolving over time that is/are periodically observed by sensor(s), where false alarms and misdetection may occur. It sits at the core of considerable scientific and engineering problems regarding, e.g., positioning, tracking and forecast. Due to the periodical working nature of physical sensors, the observation function $h_k(\cdot)$ that explains the generation of the observation is commonly formulated in discrete time series as

$$\boldsymbol{y}_k = h_k(\boldsymbol{x}_k, \boldsymbol{v}_k) \tag{1}$$

where $k \in \mathbb{N}$ indicates the distant time-instant, $\boldsymbol{x}_k \in \mathbb{R}^{D_1}$ denotes the D_1-dimensional state, $\boldsymbol{y}_k \in \mathbb{R}^{D_2}$ denotes the D_2-dimensional observation (also called measurement), and $\boldsymbol{v}_k \in \mathbb{R}^{D_2}$ denotes the observation noise. It is necessary to note that, as a rare case, continuously observed processes have also been investigated, for which the reader is referred to [1].

There are two major classes of estimation solutions that differ from each other regarding whether the hidden state process will be modelled and thereby used. The first solution, related to 'frequentist' or more widely "data-driven", is to infer the state directly from the sensor data based on maximum-likelihood rules or direct observation-state converting [2], without regard to knowledge a priori or the state process model. This data-driven solution is preferable when the sensor data are of high quality while little is known about the state process model [2, 3]. The other solution, commonly referred to as 'Bayesian' or more generally "model-driven", is to establish a Markov state transition model to link the states over time (so that to use knowledge a priori). The best known is the sequential Bayesian inference (SBI), for which a filter that consists of 'prediction' and 'correction' two steps are applied iteratively or a model-based optimization object function is developed [4]. For this, the state process (also called *dynamics*) needs to be modeled as a hidden Markov transition process in either the discrete-time or continuous-time format, as given by difference equation (2a) and differential equation (2b) respectively.

$$\boldsymbol{x}_k = g_k(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) \tag{2a}$$

$$\frac{d}{dt}\boldsymbol{x}_t = g_t(\boldsymbol{x}_t, \boldsymbol{u}_t) \tag{2b}$$

where $t \in \mathbb{R}^+$ indicates continuous time, $\boldsymbol{u}_k \in \mathbb{R}^{D_1}$ and $\boldsymbol{u}_t \in \mathbb{R}^{D_1}$ denote noises affecting the discrete/continuous time state processes g_k and g_t, respectively.

It is worth noting that: the state updating frequency must be set properly to match the sensor scanning frequency so that (2) and (1) will work recursively and *tightly* in time as well as to meet constraints [5]. In particular, the noise $\boldsymbol{u}_k$ or $\boldsymbol{u}_t$ has to be "simulated" according to the iteration period of the sensor [6]. Given that the model information about (1) and (2) is not given a priori, a general estimation framework shall estimate the model, $\mathfrak{m}_k$, jointly with the state from sensor data, i.e.,

$$p(\boldsymbol{x}_{1:k}, \mathfrak{m}_{1:k}|\boldsymbol{y}_{1:k}) = p(\boldsymbol{x}_{1:k}|\boldsymbol{y}_{1:k}, \mathfrak{m}_{1:k})p(\mathfrak{m}_{1:k}|\boldsymbol{y}_{1:k}). \tag{3}$$

Given the complete model-candidate set $\mathcal{M}$ (its parameter space is a finite set a priori), the full probability formula and the Bayes' rule show that,

$$p(\boldsymbol{x}_{1:k}|\boldsymbol{y}_{1:k}) = \sum_{\forall i:\, \mathfrak{m}_i \in \mathcal{M}} p(\boldsymbol{x}_{1:k}|\boldsymbol{y}_{1:k}, \mathfrak{m}_{1:k})p(\mathfrak{m}_{1:k}|\boldsymbol{y}_{1:k})$$

$$= \frac{1}{p(\boldsymbol{y}_{1:k})}\sum_{\forall i:\, \mathfrak{m}_i \in \mathcal{M}} p(\boldsymbol{x}_{1:k}|\boldsymbol{y}_{1:k}, \mathfrak{m}_{1:k})p(\boldsymbol{y}_{1:k}|\mathfrak{m}_{1:k})p(\mathfrak{m}_{1:k}) \tag{4}$$

where $p(\boldsymbol{y}_{1:k}) = \sum_{\forall i:\, \mathfrak{m}_i \in \mathcal{M}} p(\boldsymbol{y}_{1:k}|\mathfrak{m}_{1:k})p(\mathfrak{m}_{1:k})$.

In a typical set-up for target tracking, the observation model (1) depends on the sensor which is usually known a priori, while the state process model (2) is unknown and varies over time if the target maneuvers. Therefore, the terminology "*model*" will henceforth primarily refer to the state process model.

The SBI *modeling* involves a wide range of latent statistical models, uncertainties and constraints regarding target quantity, observability and dynamics, sensor profiles (e.g., misdetection, clutter) and background (e.g. noises can be distorted, correlated, colored and/or multiplicative) etc. Most of the time, they are unknown in realistic applications and can only be approximated on the basis of online or offline data. Even if the model can be well approximated, the full Bayesian computation could be prohibitively expensive and further approximation must be resorted to. Therefore, SBI can easily suffer from inaccurate modeling, over-approximation [2-3] and intractable likelihood [8]. However, they are not a problem for the observation-only (O2) inference [2-3] which requires little model knowledge about the targets (whether birth, death or maneuvering) or clutter. Given a sufficient number of sensors employed, data mining tools such as clustering can be applied on the O2 estimates to deal with misdetections and to avoid false alarms, namely *clustering for filtering* (C4F) [3,7]. It enjoys an accuracy that increases with the quantity and quality of sensors and therefore becomes more favorable with the escalating update of sensors nowadays in both scanning frequency and observation accuracy. This is an appealing and important property which, however, does not always hold for filters; see [9, 10].

However, in most estimation approaches, whether the model-specific SBI or the data-driven O2/C4F inference, the state-estimates are given in time series as a finite set, although there are some studies that aim to estimate this set online [11,12] at the expense of significantly growing computational cost. In multi-target scenarios, a critical issue referred to as estimate-to-track (E2T) association is required to distinguish different estimates and to associate the estimates that belong to the same target across successive time-steps. In this line of thinking, a principled way is adding *labels* to the state vector [13-16] to distinguish targets, where tracks are formed simply by linking the estimates with the same label; this essentially belongs to the theme of data association. While this seems like a reasonable idea, a number of practical and theoretical issues arise, e.g., it is argued that labels have no physical significance [17] and in some scenarios, the track labels are irrelevant, although explicitly ignoring track labels results in a better state estimation [11]. Overall, it is simply more nature and desirable to have the target "track"/state-trajectory in terms of a *function of continuous-time*, which is simply closer to the ground truth. This forms the starting point of this paper.

In this paper, we will show that with the use of *regression analysis* for linking distant estimates given by the SBI or the O2/C4F inference, continuous target tracks can be obtained as state-functions of time (each representing a target's movement over continuous time). This methodology, namely *fitting for smoothing* (F4S), is carried out in time series, facilitating the use of the nontrivial *unstructured model* information: the parameters of the state process model are unknown but the track is known to be a smooth one. It is assumed that the targets evolve *smoothly* at most of the time in the state-time space (although from time to time, maneuver, misdetection and outlier may occur), which is actually the reality of a wide class of real world targets of significance such as aircrafts and maritime ships. For example, the routes of passenger aircrafts are always smoothly planned to render the most safety and flexible operation.

The idea of employing data regression or fitting for target trajectory learning/estimation have been earlier appeared, most of which however are in batch manners and for a single target, where the regression parameters are determined based on e.g. maximum likelihood estimation [39] or Bayesian statistical inference [40-42]. In [39], directional bearing data from one or multiple sensors are investigated, where Cardinal splines (i.e., splines with equally spaced knots) of different dimensions are fit to the data. In [40], the trajectory is approximated by a cubic spline with an unknown number of knots in 2D Euclidean plane and the function estimate is determined from positional measurements which are assumed to be received in batches. For the data drawn from an exponential family, the spline knot configurations (number and locations) are changed by reversible-jump Markov chain Monte Carlo [41]. Further cubic splines was used to represent the trajectories of dynamic objects [42] based on the optimization of sliding window data. Comparably, our approach makes lesser assumption but works for multi-target case in cluttered environments and in real time manner.

The reminder of the paper is organized as follows. Section 2 gives the technical detail of the proposed C4F methodology for track estimation including target maneuver detection. Section 3 provides three simulation demonstrations and further discusses on the results. Finally, Section 4 outlines the conclusions drawn from the research.

II. Fitting-based Track Inference

In this section, we concentrate on how continuous-time target track functions can be established based on regression analysis of the discrete-time point estimates given by a recursive estimator and how they can be beneficial.

For a time series of state-estimates $\{i, \boldsymbol{x}_i\}_{i=k_1,k_1+1,\ldots,k_2}$ given by a sequential discrete-time estimator $\boldsymbol{x}_i \leftarrow p(\boldsymbol{x}_i|\boldsymbol{y}_{k_{1:i}})$ from time k_1 to time k_2, assume there is a function F_k, called regression function, depending on certain parameters to define $c_1, c_2, \ldots . c_m$, satisfying

$$\boldsymbol{x}_i \approx F_k(i; c_1, c_2, \ldots . c_m) \tag{5}$$

which fits these distant estimates over time. Therefore, this function of time F_k, offers an efficient approximate of the real track function f_t for the time interval $k_1 \le t \le k_2$ where $k_1$and k_2 are the starting and ending times of the track, respectively. Given that the function captures the target movement in a wider interval $[K_1, K_2]$, where $K_1 \le k_1, K_2 \ge k_2$, then, the state at time $t \in [K_1, K_2]$ (that does not need to be an integer) can be estimated,

$$\hat{\boldsymbol{x}}_t = F_k(t; c_1, c_2, \ldots, c_m), \forall\, K_1 \le t \le K_2. \tag{6}$$

This regression function will be updated in time series every time a new observation is received i.e., $k_2 \leftarrow k_2 + 1$, enabling sequential filtering (i.e., $t = k_2$). It can also be used for inferring the estimate for any time instants in the past (for smoothing; $t < k_2$) or even in the future (for prediction; $t > k_2$). This F4S approach can also be interpreted from a probabilistic perspective. A Bayesian treatment allows the predictive distribution to be written in the marginalized form

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{k_1:k_2}) \propto F_k(t) \cdot p(F_k\,|\boldsymbol{x}_{k_1:k_2}) \cdot p(\boldsymbol{x}_{k_1:k_2}|\boldsymbol{y}_{k_1:k_2}) \quad (7)$$

where $p(\boldsymbol{x}_{k_1:k_2}|\boldsymbol{y}_{k_1:k_2})$ is given by a recursive estimator by SBI or O2/C4F inference, $p(F_k\,|\boldsymbol{x}_{k_1:k_2})$ describes the fitting process (to get the regression function), and $F_k(t)$ utilizes the obtained regression function to infer new state at time t.

The idea of the F4S approach is using the distant estimates to infer the track function in order to estimate the state for any time instants in the valid fitting period (even for the time instants when no direct observation is reported).

It is worth noting that regression/fitting is internally sensitive to dimensionality. Hyper-surface/multi-dimensional fitting is still an intractable problem in math. To avoid this, the regression analysis could be performed in low dimensions, e.g., with respect to each dimension independently, by assuming conditional independence among dimensions.

There are many regression analysis techniques, ranging from parametric (in which the regression function is defined by a finite number of parameters) to nonparametric (where the regression function lies in a specified function bank). While maintaining a general approach, in this paper we will focus on the parametric regression analysis, especially the popular least squares (LS) regression. What follows are given in a higher level manner that is easy for understanding.

A. Least squares regression

In general, the fitting data (which are the point-estimates given by an estimator) may not all strictly work on the function as specified by (5), instead a fitting error $\{\boldsymbol{e}_i\}$ may exist, i.e.,

$$\boldsymbol{x}_i = F_k(i; c_1, c_2, \ldots, c_m) + \boldsymbol{e}_i, \forall i = k_1, k_1 + 1 \ldots, k_2 \quad (8)$$

This is particularly true in the context of target tracking, as observation noises are inevitable. The fitted function given above is therefore generally better than that given by (5) which is in fact over-fitted when noises exist. For more fundamentals about regression analysis, many works are available, e.g., [18,19]. What follows will address briefly the LS regression.

We first consider the single-model fitting, while the more general and also difficult multiple-model function fitting will be described later. For our case, one common and convenient solution is to select a function of time t that depends linearly on the parameters, in the form of

$$F(t; c_1, c_2, \ldots, c_m) = c_1\phi_1(t) + c_2\phi_2(t) + \cdots + c_m\phi_m(t) \quad (9)$$

where $\{\Phi_i(\boldsymbol{x})\}$ are a set of functions specified a priori, such as monomials $\{\boldsymbol{x}^{i-1}\}$ or trigonometric functions $\{\sin\pi i\boldsymbol{x}\}$. We call m the order of the fitting function. In our situations, m is much smaller than the number of fitting data.

In the LS curve fitting, there is an optimal order of the fitting function that controls the number of free parameters in the model and thereby governs the model complexity. To specify the order of the function, m, the best case is when the state evolution function is somewhat known in advance (e.g., the most common approximate models for target movement: nearly constant velocity (NCV), nearly constant turn (NCT), or nearly constant accelerate (NCA), etc.). Otherwise, reasonable assumptions and offline model learning are required. More generally, to determine the parameters $c_1, c_2, \ldots, c_m$ to get a good approximation of the true model f_k, a general idea is to make the residuals $\boldsymbol{e}_i, \forall i = k_1, k_1 + 1 \ldots, k_2$ simultaneously as small as possible. The error is usually defined on some norm of the vector $\boldsymbol{e} = [\boldsymbol{e}_{k_1}, \boldsymbol{e}_{k_1+1}, \ldots, \boldsymbol{e}_{k_2}]^T$. For example, in the 1-dimensional case, the un-weighted 2-norm

$$\|e\|_2 = \left(\sum_{i=k_1}^{k_2} |e_i|^2\right)^{\frac{1}{2}} \quad (10)$$

This leads to a linear system of equations to determine the minimum parameters $\hat{c}_m$'s. The resulting $F_k(t;\, \hat{c}_1, \hat{c}_2, \ldots, \hat{c}_m)$ is known as the ordinary LS approximation to the given data. This however assumes that each data point provides equally precise information. To overcome this assumption, one can assign weights w_i to each data to account for uncertainty, leading to

$$\|e\|_2 = \left(\sum_{i=k_1}^{k_2} \overline{w}_i |e_i|^2\right)^{\frac{1}{2}} \quad (11)$$

where $\overline{w}_i = w_i \left(\sum_{i=k_1}^{k_2} w_i\right)^{-1}$ is the normalized weight of w_i.

The trajectory fitting given by (8) can be interpreted more generally as an **optimization** problem

$$\operatorname*{argmin}_{F(t)} \sum_{t=k_1}^{k_2} \|\boldsymbol{y}_t - h_t(F(t), \boldsymbol{v}_t)\| \quad (12)$$

where $\overline{\boldsymbol{v}}_t$ is used to compensate the observation noise [7] and can be specified as the mean $E(\boldsymbol{v}_t)$.

The formula (12), instead of fitting estimates in the state space, performs fitting on sensor data and is more robust but also difficult. One key idea in this paper is that the state-time function F_t is defined in the continuous time space and can be expressed by some smooth functions.

B. Segmented fitting (maneuver detection)

Often, a single state trajectory may have different models over time (especially for maneuvering target); if so, it will be no more preferable to use a fixed form of function to fit the trajectory of different models. For this, *segmented fitting* (also referred to local regression) is recommended for the goal of using different fitting functions for different target movement models, or even for different stages of the same model. In fact, even for a trajectory function of constant form, segmented fitting can benefit in fitting accuracy.

To carry out segmented fitting, the fitting data are partitioned into consecutive intervals, each of which corresponds to one track function/model of relatively lower complexities. The best case is that the model change is correctly detected in one way (to be explained below), and then a separate segment is fitted to each interval and the boundaries between the segments. The fitting function is thus a sequence of grafted sub-functions

$$F(t; C) = \begin{cases} F_1(t; C_1) & k_1 \le t \le k_2 \\ F_2(t; C_2) & k_2 \le t \le k_3 \\ \vdots & \vdots \\ F_r(t; C_r) & k_r \le t \le k_{r+1} \end{cases} \quad (13)$$

where $k_1, k_2, \ldots, k_{r+1}$ are called change points, which are boundaries between intervals, and C_r is the parameter set of the rth model.

In online estimation, the newest incoming data will only affect the last segment $F_r(t; C_r)$. The previous $r-1$ segments do not need to change. Therefore, the data size of the last segment on $[k_r, k_{r+1}]$, will increase over time until a new change point is detected and then a new segment is initialized. A key challenge raised here is to detect the model change from the state-estimates flow in time series, where the change point is the desired boundary between segments. This problem is formally known as "change (-point) detection", and also as "maneuver detection" within the context of target tracking [20, 21].

There is a large body of change detection algorithms, ranging from the simple linear to the complex learning (e.g., Markov chain Monte Carlo methods) [22-25]. Here, we advocate formulating it as a hypothesis testing problem with the null (H_0) and alternative (H_1) hypotheses defined as follows:

- H_0: new data does not belong to a new model.
- H_1: new data belongs to a new model.

That is, if the underlying model can clearly explain the newly incoming data, i.e., H_0 holds, no change point is found and the new data-point (received at the current time instant) will be simply added to the underlying segment, otherwise H_1 is true. However, if the new data cannot be explained by the underlying model, it does not certainly mean that the data belongs to a new model, as it can also be an outlier (that is supposed to be singular and does not match the underlying model [26]). To account for this, the testing shall be carried out in a sliding time window on a (small) number, e.g., three, of successive data-points (only the last one is new while the other two are the preceding data). If they all fit the underlying regression function, the test will proceed ahead to the next new data point. When and only when H_1 turns out to be true to all of them, a new model is identified and will be used to re-fit these and other potentially following-up data-points. Otherwise, the "outlier" can be just abandoned (which will neither be added to the underlying segment nor be used to start a new segment). By this, the F4S approach affords an appealing potential to *identify and filter outliers contained in the raw data or the false alarms that do not fit the track.* Better mechanisms to do so remains open. We will now provide some primary insights in this line.

By the Neyman–Pearson lemma [27], when performing a hypothesis test between two hypotheses, H_0 and H_1, the likelihood-ratio test rejects H_0 in favor of H_1 when

$$\Lambda(F_r) = \frac{L(H_0|F_r)}{L(H_1|F_r)} \leq \eta \tag{14}$$

where $\mathrm{P}(\Lambda(F_r) \leq \eta | H_0) = \alpha$ is the most powerful test at the significance level α for a threshold η.

In practice, the likelihood ratio is often used directly to construct tests. However, it can also be used to suggest particular test-criteria that might be of interest; such is the case here where we define the fitting error ratio as follows. Assume that the new incoming data at time $k+1$ fits the underlying segment $\{\boldsymbol{x}_1^r, \ldots, \boldsymbol{x}_n^r\}$ and is denoted as $\boldsymbol{x}_{n+1}^r$, where r denotes the segment. Applying LS fitting on $\{\boldsymbol{x}_1^r, \ldots, \boldsymbol{x}_n^r\}$ results in the track function $F_r(t; C_r)$ with fitting error series $\{\boldsymbol{e}_1^r, \ldots, \boldsymbol{e}_n^r\}$, and applying LS fitting on $\{\boldsymbol{x}_1^r, \ldots, \boldsymbol{x}_n^r, \boldsymbol{x}_{n+1}^r\}$ results in a new fitted track function $F_{r,new}(t; C_{r,new})$ with the new fitting error series turning to be $\{\boldsymbol{e}_{1,new}^r, \ldots, \boldsymbol{e}_{n,new}^r, \boldsymbol{e}_{n+1,new}^r\}$. Here we propose two hypothesis testing criteria. The first criterion is given as

$$\epsilon = \frac{\frac{1}{n+1}\sum_{i=1}^{n+1} e_{i,new}^r - \frac{1}{n}\sum_{i=1}^{n} e_i^r}{\frac{1}{n}\sum_{i=1}^{n} e_i^r} \tag{15}$$

while the other criterion can be given as

$$\epsilon = \frac{|F_r(k+1; C_r) - x_{n+1}^r|}{\frac{1}{n}\sum_{i=1}^{n} e_i^r} \tag{16}$$

where $F_r(k+1; C_r)$ gives the fitted value of $\boldsymbol{x}_{n+1}^r$ based on the preceding model function $F_r(t; C_r)$. Equation (15) measures the change of the average fitting error caused by adding the new data point while (16) measures the fitting error of the new data point by using the underlying function. In both, given a predetermined ratio threshold ϵ_0, H_0 holds when and only when $\epsilon > \epsilon_0$, otherwise H_1 holds. For instance, we set $\epsilon_0 = 10\%$.

Often, one may only need to consider a few candidate models such as NCV, NCT, NCA, etc. most of which can be the same fitted by polynomials (but maybe of different orders). Utilizing this priori information can highly benefit the model-change detection, as argued. In general, either under-fitting or over-fitting can cause severe fitting errors. As a rule of thumb, we may start from a fitting function of a relatively higher (but not too high) order and then check whether the parameters for items of higher order are "significant" enough, i.e., whether it is necessary to keep them – i.e., we employ the principle of Occams' razor [28]. That is, given the fitting function (9), the last item $c_m\phi_m(t)$ will be confirmed to be necessary when and only when the following condition is satisfied:

$$\epsilon = \left|\frac{c_m\phi_m(t)}{c_1\phi_1(t) + \cdots + c_{m-1}\phi_{m-1}(t)}\right| > \epsilon_0 \tag{17}$$

The test will iterate similarly for lower-order item $m \leftarrow m-1$ until $m = 1$ or one item is identified as necessary. This is carried out in the reverse way to [29], which gradually increases the degree of the polynomial till the moderate degree required.

As well as finding the appropriate values for parameters within a given model, e.g., m = 2-order polynomial for NCV, m = 3 for NCT, we may wish to consider a range of different types of models in order to find the best one for a particular problem. If the data are plentiful, then one approach is simply to use the available data to train a range of models, and then to compare them on independent new data, sometimes called a validation set, and select the one having the best predictive performance. This critical issue related to offline learning is beyond the scope of this paper.

C. Long-term prediction and compensating misdetection

The target track function of continuous-time, once obtained as addressed so far, will straightforwardly enable two other key tasks, which are required in modern tracking but are intractable in existing recursive-filtering based tracking systems:

Long-term prediction. Given that the fitted function matches the state trend, the target state for arbitrary time in the effective fitting period can be inferred by the function, including the long-term future. This is of particular significance in reality, and is also an advantage of the F4S framework over existing tracking systems that are only able to predict in one time-step. In fact, not only position but also velocity (or even acceleration) can be inferred based on their physical relationship.

Compensating misdetection. When misdetection occurs in the fitting interval, one may still be able to obtain the track function via the regression analysis based on detections, and then the missed data can be recovered. From a practical perspective, this is not limited to misdetection at a single scan; successive misdetections can also be recovered in the same way. As with the aforementioned outlier detection and removal, this is an important property of the proposed F4S approach and will be partly demonstrated in our simulations.

Due to the space limitation, we omit further discussions on the above issues but we note that, in many situations, we are overwhelmed by rich sensor data for which real time and high quality processing is challenging, considering the constantly escalation and joint-use of modern sensors that have higher and higher precision and scanning frequency. For this, mature technologies and algorithms related to data mining and machine learning should be considered and can be very beneficial. In addition to aforementioned [39-42], fitting has been applied for likelihood calculation to speed up particle filtering [30]. In fact, they have received high attention in the topics of "visual tracking" [31-33] and state space modeling [34].

III. Performance Evaluation

This section demonstrates the use of regression analysis, namely F4S, within the SBI (a recursive filter) or the O2/C4F inference by means of three groups of simulations. The first is a simple 1-dimensional model, the second tracks a maneuvering target in the 2-dimensional space, and the third tracks multiple targets in cluttered environments. The following questions are considered: how does the estimation accuracy of the F4S compare to the optimal Bayesian estimator at distant time instants, and how much can F4S benefit O2/C4F inference or SBI if they work jointly.

A. Linear Gaussian Scalar Estimation

In the first example, we considered a linear and Gaussian system to estimate a scalar state that suffers from a very small process noise. The system was governed by the following state process model and observation model, respectively

$$x_k = x_{k-1} + 10 + u_k,\ \ x_0 \sim \mathcal{N}(0,1) \tag{18}$$

$$y_k = x_k + v_k \tag{19}$$

where the state process was as small as $u_k \sim \mathcal{N}(0,10^{-5})$ and the Gaussian observation noise was $v_k \sim \mathcal{N}(0,10^{-2})$.

Furthermore, our simulation generated morbid observations at time instants $k = 20, 30, 160, 161, 165$ by increasing the observation noise to $v_k \sim \mathcal{N}(0,10^{-1})$ and assumed misdetection at time instants $k = 41, 42, 166$ when no observations were reported. However, both the Kalman filter (KF) and the O2 inference are unaware of these.

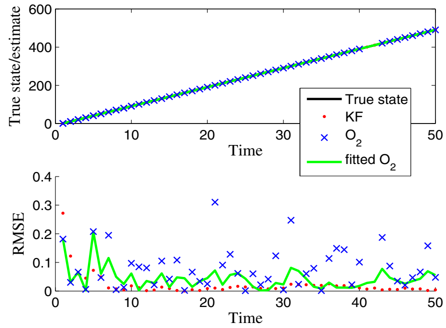

Fig.1. The F4S (fitted O2) significantly improves the time-independent point estimates given by the O2 inference through history data learning, and overcomes the sensitivity of KF to misdetection and outliers. This figure is for the simulation interval $k \in [1, 50]$

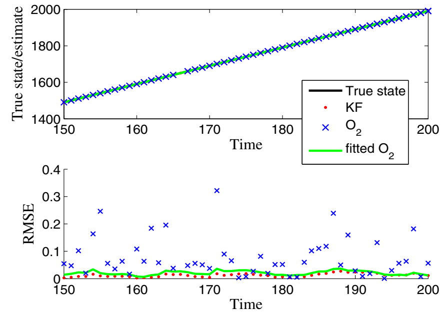

Fig.2. As time goes on (more and more sensor data are available for fitting), the benefit of applying F4S to the O2 inference becomes more significant, resulting an estimate accuracy close to that of the optimal KF. This figure is for the simulation interval $k \in [150, 200]$

The KF is expected to give the minimum mean square error (MMSE) estimate for the linear and Gaussian system, given that the model and noises are accurately assumed (including initialization) to correctly. The filter was initialized with $\mathcal{N}(0,1)$; while this ideal assumption is strong in practice, it does not necessarily indicate the 'best' initialization. In a conditional sense (e.g. the mean square error matrix is conditioned on a specific state sequence which is the case in our simulation), Kalman filter is not efficient [35].

By inversing (19) after dropping v_k, the O2 inference gives

$$\hat{x}_k = y_k \tag{20}$$

We propose to apply online LS fitting on the estimates given by the O2 inference. It turns out that the trajectory of the target was approximately a straight line. Therefore, the fitting function shall be assumed as

$$x_k = c_1 + c_2 k \tag{21}$$

corresponding to a continuous trajectory function of time $x_t = c_1 + c_2 t$ which, as stated, can provide an estimate for any time (i.e., t does not need to be an integer) including prediction.

In the proposed F4S approach, all history data were used for updating the fitting function and thereby estimating the state in time series. However, as opposed to traditional smoothing, the regression function was not used to revise historical estimates, i.e., no backward inference was involved. However, the updated fitting track function could also have been used to improve historical estimates, like a conventional Bayes smoother

Once misdetection occurred, the KF took the one-step prediction from the prior without updating (the estimate given by the naïve O2 inference outputs nothing), while the F4S used the preceding fitted function directly to estimate the state of missing observation. To capture the average result, 100 MC runs were performed (with independent observations), each run consisting of 200 time-steps.

The true state and the estimates and root-mean-square errors (RMSEs) given by the KF, the O2 inference and the fitted O2 inference, respectively, are given in Fig. 1 for the simulation interval $k \in [1, 50]$, and in Fig. 2 for the interval $k \in [150, 200]$. In both figures, distant estimates are given in dots corresponding to integer time instants ('.' for the KF and 'x' for the O2 inference) while the fitted O2 inference gives a continuous track whose RMSE values are given by lines in the figure. The results confirm that the fitting has significantly improved the estimation of the O2 inference and specifically, the more data the better. Given sufficient data, the fitted O2 inference performs closely to the optimal KF that is based on precise system modeling.

Given that the system noises are not Gaussian or the model is nonlinear, the KF will hardly get such good performance, but may lose to the O2 inference. In contrast, the proposed F4S approach significantly reduces the sensitivity of the O2 inference to outliers, whether morbid observation or missed observation. As shown next, we iterate that the state trajectory in the context of target tracking is generally of low complexity and is suitable for regression analysis.

B. Maneuvering target tracking

In this simulation, we considered applying segmented fitting to distant estimates given by the O2 inference for tracking a maneuvering target. We would omit the detail of the simulation set-up that can be found in [2]; the sensing scenario, the target trajectory and the distant O2 position estimates in time series are given in Fig.3. The O2 inference was set up exactly the same as given in [2]. There were two significant differences as compared to the last simulation:

- The fitting was conducted in the multi-dimensional space (with conditional independence assumption among dimensions).
- We realized fitting in the planar 2D position space to facilitate the use of curve or straight line fitting.

Once a model change point was detected, no preceding data but only the data generated after detection (until another model change was found) were considered for fitting. For this, we used $\epsilon_0 = 10\%$ in (17) to detect the target model change.

In the general case of polynomial-curve fitting, higher order assumption works well for the model of lower order (though having the risk of over-fitting) but not vice versa. Therefore, given that we assumed a relatively large fitting function order

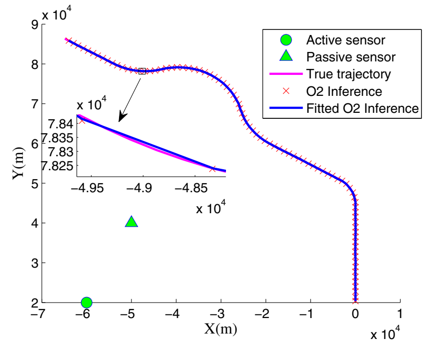

Fig.3. The tracking scenario: the real continuous-time trajectory of the maneuvering target, the naïve O2 inference (given in distant points) and its online fitted outcome (continuous-time fitting curve)

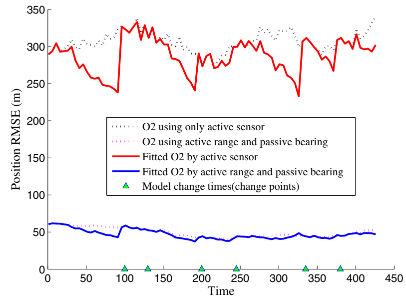

Fig.4. The online fitting (F4S) notably improves the RMSE in position estimates given by the naïve O2 inference for a maneuvering target. Comparably, the benefit increases with time during each segment.

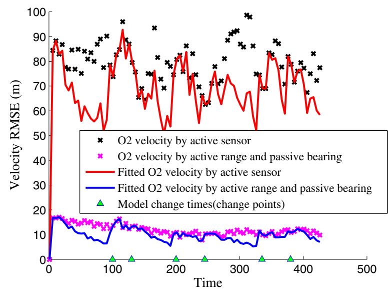

Fig.5. The online fitting (F4S) notably improves the RMSE in velocity estimates of the naïve O2 inference; the benefit increases with time during each segment.

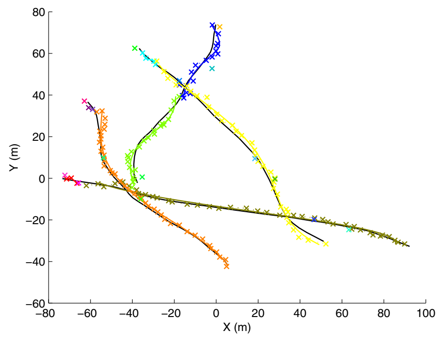

Fig.6. The F4S (given in colored lines) improves the track quality ('x' in different colors) given by the Bayesian-manner dyeing approach [36] to make them smoother and mostly closer to the ground truth (black lines)

(in our case, $m = 3$ for (9)), the F4S approach is not able to handle both straight line fitting and curve fitting. Note that, in contrast to filtering, in fitting the model information does not need to be precise and is in fact unstructured: only knowing that the target moves *smoothly* rather than any exact form which prevents direct SBI unless the model is also estimated, as shown in (4). The resulting F4S approach gets rid of sophisticated system modelling and sensitive parameter setting and is, therefore, at least reliable and computing fast.

As shown in Fig. 4 and 5, the online fitting improved the output of the naïve O2 inference (either position or velocity) where, the velocity was given by a direct difference operation on the position estimation. Specifically, the improvement increases with time (the RMSEs decrease) until the model changes. That is simply because the more data there are, the better the fitting. This example demonstrates again that the fitting is an effective tool to connect data over time series, like *smoothing*.

C. Multi-target tracking in cluttered environments

In this case, we evaluated the performance of the F4S approach applied on the estimates given by a multi-target filter based on the SBI. The tracking was designed in a 2-dimensional scenario over the region [-100,100]×[-100,100] m^2; unit is omitted hereafter. Four moving targets were simulated where the state comprised planar position and velocity $\boldsymbol{x}_k = [p_{x,k}, \dot{p}_{x,k}, p_{y,k}, \dot{p}_{y,k}]$. These targets appeared according to a Poisson point process with intensity function $\gamma_i = 0.1\mathcal{N}(.;B_i,Q)$, where B_1= [-72, 3, 0, 0]$^\mathrm{T}$, B_2=[0, 0, 75, -3]$^\mathrm{T}$, B_3= [-62, 2.6, 36, -1.5]$^\mathrm{T}$, B_4= [-36, 1.5, 62, -2.6]$^\mathrm{T}$, *Q*=*diag*([5, 1, 5, 1]$^\mathrm{T}$), and *diag*(a) gives a diagonal matrix with diagonal a. The target state transition is given as

$$\boldsymbol{x}_k = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_{k-1} + \begin{bmatrix} \Delta^2/2 & 0 \\ \Delta & 0 \\ 0 & \Delta^2/2 \\ 0 & \Delta \end{bmatrix} \begin{bmatrix} w_{1,k} \\ w_{2,k} \end{bmatrix} \tag{22}$$

where the sampling time Δ=1, and the processing Gaussian noise $w_{1,k} \sim \mathcal{N}(0,0.5)$, $w_{2,k} \sim \mathcal{N}(0,0.1)$. Furthermore, the position-only observation equation was given by

$$\boldsymbol{y}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} v_{1,k} \\ v_{2,k} \end{bmatrix} \tag{23}$$

with the Gaussian noise $v_{1,k} \sim \mathcal{N}(0,2.5)$, $v_{2,k} \sim \mathcal{N}(0,2.5)$.

Each target had a target survival probability $p_S(x)$=0.98 and a detection probability $p_D(x)$ =0.99. Clutter was uniformly distributed over the region with an average rate of 5 points per scan, i.e., $\kappa_k = 5/200^2$. 800 particles per expected target were used and the total number of particles was hard-limited to be not less than 500, for the implementation of the particle dyeing-based sequential Monte Carlo-probability hypothesis density (SMC-PHD) filter [36]. The dyeing approach is a sequential Bayesian solution that dyes/marks particles with respect to their contribution to different estimates; estimates can therefore be associated between two successive scans for track continuity. However, isolated tracks and broken tracks caused by false alarms and misdetection respectively were reserved in the filter output; we apply no ad-hoc solutions to fix them.

The target position estimates that comprise a single track (formed by the dyeing approach) can be further fitted by the proposed F4S approach, resulting in a continuous-time track. As shown in Fig. 6, the real trajectories of the four targets are plotted in black lines, the tracks formed by the dyeing approach are plotted in different colors (all marked by 'x') and each of them is further fitted (given by continuous lines in different colors) by the F4S approach. That is, F4S was realized here based on the dyeing approach, both implemented within the same SMC-PHD filter. The fitting has appealingly improved the track quality to make them smoother, continuous and mostly closer to the ground truth (though sometimes, the opposite occurs). Regardless of the track ID, the accuracy improvement gained by fitting on the position-only estimation was about 6~10% on average in the sense of RMSE.

Furthermore, to identify and remove false tracks (including isolated points) or to connect broken tracks that likely belong to the same target, specific "track refinement" solutions such as [37-38] can be applied for further *smoothing* if *structured* model information is feasible. This is a valuable extension of the F4S approach for dealing with false alarms and misdetections, which remain greatly challenges in the tracking community. Compared to the "model-specific" Bayesian smoothing, the "data−driven" F4S methodology is based on learning the model information from the data rather than taking them as granted/assumptions and therefore has more flexibilities to operate. Given accurate model information available and fully used, they both should converge to the same result. More theoretical and experiment studies are highly desired in this regard.

IV. Conclusion

This paper has presented a framework for estimating the continuous-time target track/trajectory function via regression analysis, resulting in a methodology for continuous-time state estimation called fitting for smoothing (F4S). It can be interpreted as optimization with the purpose of finding a trajectory function that best fits the sensor data, without making Markov assumption but assume the trajectory as a function of time. It is carried out with a recursive estimator, e.g. a model-specific sequential Bayesian estimator or the data-driven observation-only inference, given that the target has a relatively smooth (at least locally for short-time) evolving trajectory that matches the fitting function. The resulting continuous-time track function can be used to refine the estimates given by the recursive estimator for either the past (apparently equal to classic smoothing), the current (for filtering) or the future (for prediction). More appealingly, the F4S approach facilitates identifying and removing outliers (whether morbid observations, false alarms or misdetections), and compensating misdetection (whether a single scan or multiple successive scans) from both of which existing estimators suffer. Long-term prediction is also feasible for smoothly moving targets, which is of high significance in realistic problems such as tracking passenger aircraft or ships. All of these are based on online learning of the sensor data and require little sophisticated target modelling and complicated computation. Simple simulations are provided for illustration. While data fitting based continuous-time track estimation may sound attractive and promising, more insights are to be explored.

ACKNOWLEDGMENT

This work is partly supported by the Marie Skłodowska-Curie Individual Fellowship (H2020-MSCA-IF-2015) and the Spanish Ministry of Economy and Competitiveness under grant PTQ-14-06928. Earlier material of this paper has been partly given in an invited technical talk at the Center for Information Engineering Science Research at Xi'an Jiaotong University, Xi'an China, in December 2015.

REFERENCES

[1] A. Ghoreyshi and T. D. Sanger, "A nonlinear stochastic filter for continuous-time state estimation," IEEE Transactions on Automatic Control, vol. 60, no. 8, pp. 2161-2165, Aug. 2015.

[2] T. Li, J. M. Corchado, J. Bajo, S. Sun, and J. F. Paz, "Effectiveness of Bayesian filters: An information fusion perspective," Information Sciences, 2016, 329, 670-689.

[3] T. Li, J. M. Corchado, J. Bajo, and G. Chen, "Multi-target detection and estimation with the use of massive independent, identical sensors," Proc. SPIE 9469, Sensors and Systems for Space Applications VIII, 94690G, May, 2015.

[4] H. Michalska, D.Q. "Mayne moving horizon observers," Proceeding IFAC Symposium on Nonlinear Systems Design, Bordeaux, France (1992), pp. 576–581.

[5] X. R. Li, "Compatibility and modeling of constrained dynamic systems," 19th IEEE International Conference on Information Fusion, Heidelberg Germany, July 5-8, 2016.

[6] T. Li. A gap between simulation and practice for recursive filters: On the state transition noise. arXiv:1308.1056

[7] T.Li, J.M. Corchado, S. Sun and J. Bajo, Clustering for filtering: Multi-target detection and estimation using multiple/massive sensors, submitted to Information Sciences.

[8] M.N. Tran, D. J. Nott, and R. Kohn. "Variational Bayes with intractable likelihood," arXiv:1503.08621v1.

[9] S. Coraluppi, M. Guerriero, P. Willett, and C. Carthel. "Fuse-before-track in large sensor networks," Journal of advances in Information Fusion, 2010, vol. 5, no.1, pp. 18-31.

[10] M. R. Morelande, and Á. F. García-Fernández, "Analysis of Kalman filter approximations for nonlinear measurements," IEEE Trans. Signal Process. 2013, vol. 61, no. 22, pp. 5477-5484.

[11] M. Guerriero, L. Svensson, D. Svensson and P. Willett, "Shooting two birds with two bullets: How to find minimum mean OSPA estimates," in 13th Conference on Information Fusion, Edinburgh, 2010, pp. 1-8.

[12] Á. F. García-Fernández, L. Svensson, "Trajectory probability hypothesis density filter ," arXiv:1605.06432v1 [stat.ML] 20 May 2016.

[13] E. Aoki, Y. Boers, L. Svensson, P. Mandal, and A. Bagchi, "A Bayesian look at the optimal track labelling problem," in Proceedings of the IET Data Fusion & Target Tracking Conference: Algorithms and Applications, London, UK, 2012.

[14] A. F. García-Fernández, M. R. Morelande, and J. Grajal, "Bayesian sequential track formation," IEEE Trans. Signal Process., vol. 62, no. 24, pp. 6366–6379, 2014.

[15] B.T. Vo and B.N. Vo, "Labeled random finite sets and multi-object conjugate priors," IEEE Trans. Signal Process. vol. 61, no. 13, pp. 3460–3475, 2013.

[16] A.F. García-Fernández, J. Grajal, and M. Morelande, "Two-layer particle filter for multiple target detection and tracking," IEEE Trans. Aerosp. Electron. Syst. vol. 49, no. 3, pp. 1569–1588, 2013.

[17] L. Svensson, and M. Morelande, Target tracking based on estimation of sets of trajectories, 17th International Conference on Information Fusion, Salamanca, Spain, July 8-10, 2014.

[18] M. Bishop, Pattern Recognition and Machine Learning, Springer Science+Business Media, LLC., 2006.

[19] R. W. Hamming, Numerical Methods for Scientists and Engineers, Dover Publications, 1987.

[20] X.R. Li, and V.P. Jilkov. "Survey of maneuvering target tracking. part I: dynamic models," IEEE Trans. Aerosp. Electron. Syst. 2003, vol. 39, no.4, pp. 1333-1364

[21] J. Ru, V. P. Jilkov, X.R. Li, A. Bashi, "Detection of target maneuver onset," IEEE Trans. Aerosp. Electron. Syst. 2009, vol.45, no.2,p. 536-554.

[22] B.E. Brodsky, B.S. Darkhovsky, Nonparametric Methods in Change-Point Problems, Springer-Science + Business Media, B.V., 1993.

[23] B.C. Bates, R. E. Chandler, and A. W. Bowman, "Trend estimation and change point detection in individual climatic series using flexible regression methods," J. Geophys. Res., 2012, 117 (D16106), 1-9.

[24] S. Liu, M. Yamada, N. Collier, and M. Sugiyama, "Change-point detection in time-series data by relative density-ratio estimation," Neural Networks, 2013, 43, 72-83.

[25] C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadic, "Particle methods for change detection, system identification and control," Proceedings of the IEEE, 2004, vol. 92, no. 3, pp. 423–438.

[26] I. Ben-Gal, Outlier detection, In: Maimon O. and Rockach L. (Eds.) Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, Kluwer Academic Publishers, 2005.

[27] E. L. Lehmann, and J.P. Romano, Testing Statistical Hypotheses. Springer Science+Business Media, LLC. 2005.

[28] Occams' razor – Wikipedia, [online]: https://en.wikipedia.org/wiki/Occam%27s_razor, archived on July 12, 2016.

[29] B. Zheng, J. Takamatsu and K. Ikeuchi, "An adaptive and stable method for fitting implicit polynomial curves and surfaces," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 32, no. 3, pp. 561-568, 2010

[30] T. Li, S. Sun, J. M. Corchado1, T. P. Sattar and S. Si, Numerical fitting-based likelihood calculation to speed up the particle filter, International Journal of Adapt Control and Signal Processing, DOI: 10.1002/acs.2656.

[31] B. Yang and R. Nevatia, "Multi-target tracking by online learning of non-linear motion patterns and robust appearance models," 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), , Providence, RI, 2012, pp. 1918-1925

[32] B. Babenko, M. H. Yang and S. Belongie, "Robust Object Tracking with Online Multiple Instance Learning," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 8, pp. 1619-1632, Aug. 2011.

[33] Z. Kalal, K. Mikolajczyk and J. Matas, "Tracking-Learning-Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pp. 1409-1422, July 2012.

[34] M. Karl, M. Soelch, J. Bayer, and P.van der Smagt, Deep variational Bayes filters: unsupervised learning of state space models from raw data, arXiv:1605.06432v1 [stat.ML] 20 May 2016.

[35] C. Fritsche, U. Orguner, F. Gustafsson, On parametric lower bounds for discrete-time filtering, 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, 2016, pp. 4338-4342.

[36] T. Li, S. Sun, J.M. Corchado, M.F. Siyau, "A particle dyeing approach for track continuity for the SMC-PHD filter," In Proc. 17th International Conference on Information Fusion, Salamanca, Spain, 7-10 July, 2014.

[37] L. Jiang, and S. S. Singh, Tracking multiple moving objects in images using Markov Chain Monte Carlo, 2016, arXiv:1603.05522.

[38] T. Vu, B.N. Vo, and R. Evans, "A particle marginal Metropolis-Hastings multi-target tracker," IEEE Trans. Signal Process. vol. 62, no. 15, pp. 3953 – 3964, 2014.

[39] Anderson-Sprecher, R. and Lenth, R.V. (1996) "Spline Estimation of Paths Using Bearings-Only Tracking Data," Journal of the American Statistical Association, 91, 276-283.

[40] M. Hadzagic and H. Michalska, "A Bayesian inference approach for batch trajectory estimation," Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on, Chicago, IL, 2011, pp. 1-8.

[41] I. DiMatteo, C. R. Genovese, and R. E. Kass, "Bayesian curve fitting with free-knot splines," Biometrika, vol. 88, pp. 1055–1071, 2001.

[42] C. Bibby, I. Reid, A Hybrid SLAM Representation for Dynamic Marine Environments, 2010 IEEE International Conference on Robotics and Automation, May 3-8, 2010, Anchorage, Alaska, USA.