Published in final edited form as:
Control Conf ECC Eur. 2020 May ; 2020: 527–533. doi:10.23919/ecc51009.2020.9143818.

Estimation of general time-varying single particle tracking linear models using local likelihood

Boris I. Godoy[1], **Nicholas A. Vickers**[1], **Y. Lin**[2], **Sean B. Andersson**[1,2]

[1]Department of Mechanical Engineering, Boston University, Boston, MA, 02215 USA

[2]Division of Systems Engineering, Boston University, Boston, MA, 02215 USA

Abstract

In this work, we study a general approach to the estimation of single particle tracking models with time-varying parameters. The main idea is to use local Maximum Likelihood (ML), applying a sliding window over the data and estimating the model parameters in each window. We combine local ML with Expectation Maximization to iteratively find the ML estimate in each window, an approach that is amenable to generalization to nonlinear models. Results using controlled-experimental data generated in our lab show that our proposed algorithm is able to track changes in the parameters as they evolve during a trajectory under real-world experimental conditions, outperforming other algorithms of similar nature.

I. INTRODUCTION

Single Particle Tracking (SPT) refers to a class of experimental techniques and mathematical algorithms for following small particles (less than 100 nm) moving inside living cells, including viruses, proteins, and strands of RNA [1]. By labeling the particles with a fluorescent reporter such as a fluorescent protein or quantum dot, the motion of the tag and, by extension, the motion of the particle can be observed. While there are many different schemes, the general paradigm in SPT involves capturing a series of widefield fluorescence images, localizing the fluorescent particle in each frame to form a trajectory, and then analyzing the trajectory to estimate motion model parameters. Knowing these parameters provides vital information to understanding the underlying biological processes and has been used to understand processes from viral infection [2], intracellular communication between neurons [3], and other applications.

There are many models that are relevant to the biophysical application domain including (among others) free diffusion, confined diffusion, directed motion, and combinations of these, such as joint diffusion and directed motion [4]. Among the variety of motion models, diffusion is arguably the simplest and one of the most commonly used models when describing biomolecular motion. Because it is a fairly straightforward model, it is amenable to analysis and the use of optimal estimation. This was recognized and developed in [5]–[8]. Using optimal techniques such as Maximum Likelihood (ML) estimation places the

bgodoy@bu.edu .

problem on firm theoretical grounding, ensures that the analysis method is both consistent and asymptotically efficient, and provides a rigorous understanding of the accuracy and performance of the estimator. A similar approach has also been developed for the analysis of confined diffusion [9].

Existing ML schemes in SPT assume that model parameters, while unknown, are fixed (time invariant). There have been efforts on extending the analysis to determine the most likely model among a given set [4] but these also assumed fixed parameters [10]. One of the authors has considered time-varying parameters using a jump Markov model [11] but such models do not allow for continuously varying parameters, assume that the number of levels to which the parameter can jump is known, and impose a probabilistic structure on the changing parameter values that may be non-physical. In [12], the authors considered for the first time a windowed approach for the pure diffusion setting, combined with a strategy of change detection to refine the estimation of the diffusion constants after initial estimates were generated.

In this work, we develop an alternative estimation algorithm based on *local* likelihood estimation, extending beyond simple diffusion to general linear models typically found in SPT with time-varying parameters. The main idea is to take a window of data and fit a time-invariant parameter model in the window. Then the window is moved along by a fixed amount and the process repeated to trace out time-varying parameters. One of the fundamental insights from the statistical literature [13], [14] is that rectangular windows induce a kind of Gibbs-ringing (familiar from Fourier analysis) which can produce bias in the estimates. Thus, only smooth windows should be used. Unfortunately, in a lot of the applied literature this basic insight is unknown and rectangular windows are still very common.

The problem of estimating diffusion constants can be translated to one of estimating the process noise covariance matrix. This is a well-studied problem with a variety of approaches in the literature, e.g., maximum likelihood [15], [16], Bayesian estimation [17], covariance matching [15], and correlation techniques [18]. Some of these works are designed for adaptive filtering, including [15], [19]–[21]. A fair comparison against these adaptive innovation-based algorithms is carried out in [22] where it is shown that the algorithm introduced in that paper outperforms the others. In the present work, then, we compare our method to that of [22]. It is important to note, however, that our method is more general than that of [22], allowing for estimation of any of the model parameters and not just the process covariance.

II. SPT MODEL

SPT refers to a broad collection of experimental techniques and data analysis algorithms for studying the motion of single, labeled biological macromolecules moving in their native environments, including inside living cells. The basic paradigm is to collect a series of images from a CCD camera, localize the particle in each of the images, link these positions into a trajectory, and then analyze the trajectory for motion model parameters using, typically, a curve fit to an MSD curve. Localization can be done using a variety of different algorithms, ranging from simple centroid calculations through full ML estimation [23]–[25]. One of the most common techniques is a fit of the intensity in the image to a simple Gaussian profile, yielding an accuracy on the order of 10 nm, well below the diffraction limit of light.

Given a trajectory, analysis typically proceeds by selecting a parameterized model and using the trajectory to identify those parameters. In this work we consider the following quite general model (expressed here in the *x*-axis only with similar models holding in the other axes):

$$\begin{aligned} x_{k+1} &= a_t x_k + b_t + w_k, \;\; w_k \sim \mathcal{N}(0, q_t), \\ y_k &= x_k + v_k, \;\; v_k \sim \mathcal{N}(0, r_t), \end{aligned} \tag{1}$$

where $x_k, y_k, w_k, v_k \in \mathbb{R}$, $q_t = 2D_t\Delta t$ is the variance of the process noise defined by the diffusion coefficient D_t and the sampling time Δt, and r_t is the variance of the measurement noise as generated by a variety of processes, including shot noise due to the physics of photon generation in fluorescence and read-out noise in the camera. System (1) can be used to represent several models important in the SPT application. For example, setting $a_t = 1$, $b_t = 0$ describes pure diffusion; if we choose $a_t < 1$, $b_t = 0$, we have the Ornstein-Uhlenbeck model that can capture tethered motion of a biomolecule or be used to approximate confined diffusion [9], [26]. We note that the restriction to one dimension in (1) is for simplicity of presentation only. The extension to the multi-dimensional setting is straightforward since motion in the different axes is generally considered to be independent.

The model in (1) also allows for time-varying parameters. For example, the process noise q_t can vary as the particles move into different regions of the cells with different local environments, interact with different species in their surroundings, or go through biochemical changes due to the natural activity in the cell. This insight will set up the conditions for the examples given in Sec. IV-B. For notation purposes, we add subindex t to a parameter to indicate that it is time-varying, and remove the subindex t when it is presumed to be time-invariant.

Despite the complex reality of measurement noise, it is often modeled as a simple Gaussian white noise process. The Gaussian approximation works well at high signal-to-noise levels but does break down at low intensities, necessitating a more complex description of the data [27]. In general, the measurement noise r_t has reasonably static statistics as it is driven primarily by the experimental equipment. However, r_t may also due to non-uniform background or changes in illumination intensity during the measurements.

Our goal, then, is to develop an algorithm based on local ML estimation that can track the time-varying parameters of motion in (1), namely, a_t, b_t, q_t and r_t. (Of course, if some parameters in the model are known a priori or can be independently measured, one should use that value instead of trying to estimate it from the data; the approach below is easily modified for estimating a subset of these parameters.)

III. Inference problem

Our goal is to use a local likelihood approach to handle time-varying estimation. We begin with a single window where the problem reduces to the time-invariant case.

A. Time-invariant parameter estimation via EM

Consider model (1). We define β, the parameter vector containing all the unknowns, that is, $\beta = [\Gamma \ q \ r]^T$, where $\Gamma = [a \ b]$. The ML estimation problem (for a particular window) is then given by

$$\hat{\beta} = \underset{\beta}{\operatorname{argmax}} \ l(\beta), \tag{2}$$

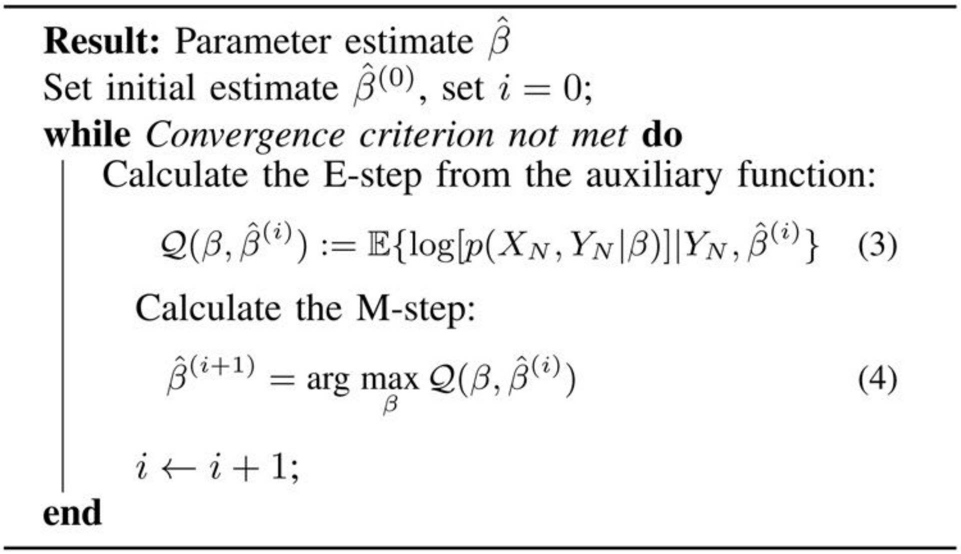
where $l(\beta)$ is the log-likelihood function defined as $l(\beta) = \log p(Y_N|\beta)$ and Y_N is the observed data defined as $Y_N = y_1,...,y_N$. For convenience and future reference, we also define $X_N = \{x_1,...,x_N\}$.

Many authors have worked on problem (2) for model (1) (or on variations) using the EM algorithm, see, e.g., [28]–[32] and references therein. The EM algorithm produces a sequence of estimates $\hat{\beta}^{(i)}, \ i = 1, 2, \cdots$, of the unknown parameter β, which is guaranteed to converge to a local maximum of the log-likelihood function $l(\beta)$ [32]. The basic idea is to use a *hidden*[1] variable, which in our case is taken to be the underlying particle trajectory X_N, to create an auxiliary function $\mathcal{Q}$, which approximates the log-likelihood function. The EM algorithm is summarized in Algorithm 1.

[1] The terminology arises from the statistics literature.

Algorithm 1:

Time invariant EM



```
Result: Parameter estimate $\hat{\beta}$
Set initial estimate $\hat{\beta}^{(0)}$, set $i = 0$;
while Convergence criterion not met do
    Calculate the E-step from the auxiliary function:
        $\mathcal{Q}(\beta, \hat{\beta}^{(i)}) := \mathbb{E}\{\log[p(X_N, Y_N|\beta)]|Y_N, \hat{\beta}^{(i)}\}$    (3)
    Calculate the M-step:
        $\hat{\beta}^{(i+1)} = \arg\max_{\beta} \mathcal{Q}(\beta, \hat{\beta}^{(i)})$    (4)
    $i \leftarrow i + 1$;
end
```

It turns out that, under quite general conditions, if the estimate $\hat{\beta}^{(i+1)}$ is chosen as in (4), then the sequence of numbers $l(\hat{\beta}^{(i)}), i = 1, 2, \cdots$ is monotonically increasing, and therefore the parameter sequence $\hat{\beta}^{(i)}$ converges to a local maximum of the log-likelihood function $l(\beta)$ [32].

It is well-known, see e.g. [33, p.343], that for the system in (1), with time-invariant parameters ($a_t = a$, $b_t = b$, $q_t = q$, $r_t = r$), the associated E-step (ignoring constant terms and the parameters of the initial condition) is given by

$$\mathcal{Q}\left(\beta, \hat{\beta}^{(i)}\right) = \mathbb{E}\left\{-2l(\beta) \mid Y_N, \hat{\beta}^{(i)}\right\} = -N\log q^{-1} - N\log r^{-1} + \sum_{k=1}^{N} \mathbb{E}\left\{(y_k - x_k)^2 r^{-1} \mid Y_N, \hat{\beta}^{(i)}\right\} + \sum_{k=1}^{N} \mathbb{E}\left\{(x_{k+1} - \Gamma z_k)^2 q^{-1} \mid Y_N, \hat{\beta}^{(i)}\right\} \tag{5}$$

where, for convenience, we have defined $z_k = \left[x_k \ u_k\right]^T$. Notice that we have included an input u_k with enough energy (different to the constant 1) that does not exist in the original model (1), with the purpose of improving the estimation of b_t (or b for the time-invariant case). $\mathbb{E}\left\{\cdot \mid Y_N, \hat{\beta}^{(i)}\right\}$ refers to the expected value given the incomplete data Y_N, see e.g. [28], [29]. For details about the optimization of the auxiliary function $\mathcal{Q}$ (the M-step), we refer to [29], where a general robust numerical implementation can be found.

B. Time-varying estimation using local likelihood

In this section, we describe an algorithm to trace time-varying parameters associated with (1). We pose a time-varying likelihood, which is time-invariant in a window of a nominated length *h*, optimize in this window, and continue the process by leaving one sample out and taking a new one in order to keep the same window size. The process finishes when we have included the last available sample. There are alternative ways to local modeling. For example, there is much literature on how to model data using a global polynomial fit. However, this method can potentially have the drawback of needing a large number of components to have a reasonably low bias [14], and, as a consequence, may lead to over-parametrization, which has an effect on the variance of the estimate. The local approximation approach, in general, only requires a small number of parameters.

The local likelihood is defined as:

$$l_t(\beta_t) = \sum_{k=1}^{N} K_{k,t} l(y_k \mid \beta_t), \tag{6}$$

where $K_{k,t} := K\left(\frac{k-t}{h}\right)$ is the *kernel*, h is the window length[2], t is a point within the window, usually the middle point, k indicates any point within the window, and $l(y_k|\beta_t)$ is the likelihood of $y_k|\beta_t$. Different values for h can be used, as well as different kernels, and appropriate selection of them is a nontrivial issue, see e.g. [14]. However, it is important to use smooth windows to minimize the effects caused by the Gibbs ringing phenomenon, which is achieved by using $K_{k,t}$ to create a window with *rounded* edges, see e.g. [34]. Examples of such kernels can be found from the following expression

$$K(v) = \begin{cases} \frac{1}{\beta(1/2, \gamma+1)}\left(1 - v^2\right), & \text{if } |v| \leq 1, \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $K(v) \geq 0$ and $\int_{-\infty}^{\infty} K(v) dv = 1$. For $\gamma = \{0, 1, 2\}$, we obtain the uniform, Epanechnikov, and biweight kernels, respectively. The kernel progressively downweights data points far from the kernel centre. In this way, data points are introduced smoothly into the windowed neighbourhood.

If we now multiply (6) by −2, consider (5), take $\mathbb{E}\left\{\cdot \mid Y_N, \hat{\beta}_t^{(i)}\right\}$ on both sides, then we can find that the time-varying auxiliary function is given by

$$\mathcal{Q}_t\left(\beta_t, \hat{\beta}_t^{(i)}\right) = E\left\{-2l(\beta_t) \mid Y_N, \hat{\beta}_t^{(i)}\right\} \tag{8}$$

where the log-likelihood $l(\beta_t)$ is given by (ignoring constant terms and the initial condition)

[2] or *bandwidth* in the statistics literature

$$-2l(\beta_t) = \sum_{k=1}^{N} K_{k,t} l(y_k \mid \beta_t), \tag{9}$$

with l defined as

$$l(y_k \mid \beta_t) := (x_{k+1} - \Gamma_t z_k)^2 q_t^{-1} + (y_k - x_k)^2 r_t^{-1} . \tag{10}$$

Remark 1: Notice that the difference of $\mathcal{Q}_t$ with respect to the $\mathcal{Q}$ (found in the time-invariant EM) is the inclusion of the kernel function $K(v)$, where $v := (k - t)/h$, and the time dependancy of the parameters of interest. We have added the sub-index t in this function to denote that it depends on the time-varying parameter β_t.

For our scalar case, and discarding the constant terms and the parameters for the initial conditions, we can obtain the following auxiliary function $\mathcal{Q}_t$

$$\begin{aligned} \mathcal{Q}_t\left(\beta_t, \hat{\beta}_t^{(i)}\right) = &-\left(\log q_t^{-1} + \log r_t^{-1}\right) \sum_{k=1}^{N} K_{k,t} + \\ &q_t^{-1}\left[\tilde{S}_{11} - \tilde{S}_{01}^T \Gamma_t^T - \Gamma_t \tilde{S}_{01} + \Gamma_t \tilde{S}_{00} \Gamma_t^T\right] + \\ &r_t^{-1} \sum_{k=1}^{N} K_{k,t}\left[\left(y_k - \hat{x}_{k \mid h}\right)^2 + P_{k \mid h}\right], \qquad \text{with} \end{aligned} \tag{11}$$

$$\begin{aligned} \tilde{S}_{11} &= \sum_{k=1}^{N} E\left\{K_{k,t} x_{k+1}^2 \mid Y_N, \hat{\beta}_t^{(i)}\right\} \\ &= \sum_{k=1}^{N} K_{k,t}\left[\hat{x}_{k+1 \mid h}^2 + P_{k+1 \mid h}\right], \end{aligned} \tag{12}$$

$$\begin{aligned} \tilde{S}_{01} &= \sum_{k=1}^{N} E\left\{K_{k,t}\begin{bmatrix} x_k x_{k+1} \\ u_k x_{k+1} \end{bmatrix} \mid Y_N, \hat{\beta}_t^{(i)}\right\} \\ &= \sum_{k=1}^{N} \begin{bmatrix} K_{k,t}\left[\hat{x}_{k+1 \mid h} \hat{x}_{k \mid h} + P_{k+1,k \mid h}\right]^T \\ K_{k,t} u_k \hat{x}_{k+1 \mid h} \end{bmatrix}, \end{aligned} \tag{13}$$

$$\begin{aligned} \tilde{S}_{00} &= \sum_{k=1}^{N} E\left\{K_{k,t}\begin{bmatrix} x_k^2 & x_k u_k \\ u_k x_k & u_k^2 \end{bmatrix} \mid Y_N, \hat{\beta}_t^{(i)}\right\} \\ &= \sum_{k=1}^{N} \begin{bmatrix} K_{k,t}\left[\hat{x}_{k \mid h}^2 + P_{k \mid h}\right] & K_{k,t} u_k \hat{x}_{k \mid h} \\ K_{k,t} \hat{x}_{k \mid h} u_k & K_{k,t} u_k^2 \end{bmatrix} \end{aligned} \tag{14}$$

and where the values of $\hat{x}_{k \mid h}$, $P_{k,k-1 \mid h}$, and $P_{k/h}$ are defined by the smoothed distributions within the appropriate window. In our setting, these can be calculated from the Kalman smoother equations within the window h, see e.g. [35]. $K_{k,t}$ is taken out of the expected values since it is deterministic.

The algorithm also depends upon choosing a window length $h = h_o N$ where $h_o << 1$. The choice of h_o affects the rate of change of the parameters that can be followed as well as the accuracy and robustness to noise. Typically some trial and error, as well as experience, is needed to select an appropriate window size. However, there are data-based methods that can be used based on the Steins Unbiased Risk Estimator (SURE) [36].

C. Optimization of the time-varying E-step

We now optimize the time-varying E-step. Taking the derivatives of $\mathcal{Q}_t$ with respect to Γ_t, Q_t^{-1} and R_t^{-1} and setting them to zero yields:

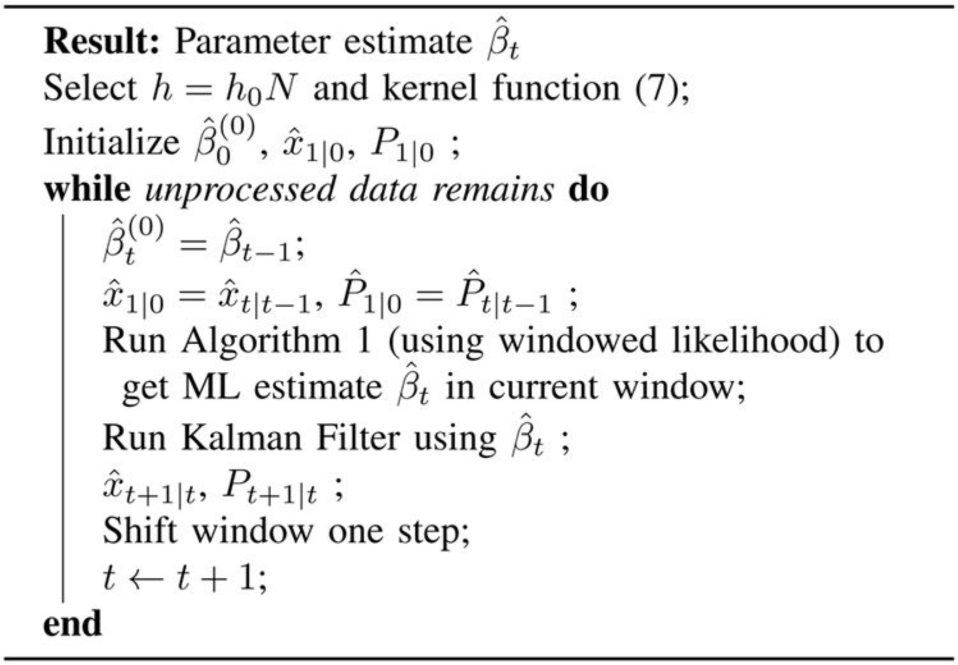
$$\begin{aligned}
\hat{\Gamma}^{(i+1)} &= \tilde{S}_{01}\tilde{S}_{00}^{-1},\\
\hat{Q}_t^{(i+1)} &= \frac{1}{n}\Big(\tilde{S}_{11} - \tilde{S}_{01}^T\big(\Gamma_t^{(i+1)}\big)^T - \Gamma_t^{(i+1)}\tilde{S}_{01} -\\
&\quad \Gamma_t^{(i+1)}\tilde{S}_{00}\big(\Gamma_t^{(i+1)}\big)^T\Big),\\
\hat{R}_t^{(i+1)} &= \frac{1}{n}\sum_{k=1}^{N} K_{k,t}\Big[\big(y_k - \hat{x}_{k\mid h}\big)\big(y_k - \hat{x}_{k\mid h}\big)^T + P_{k\mid h}^T\Big],
\end{aligned} \tag{15}$$

where $n = \sum_{k=1}^{N} K_{k,t}$.

The time-varying estimation algorithm is summarized in Algorithm 2.

Algorithm 2:

Time-varying EM



```
Result: Parameter estimate $\hat{\beta}_t$
Select $h = h_0 N$ and kernel function (7);
Initialize $\hat{\beta}_0^{(0)}$, $\hat{x}_{1|0}$, $P_{1|0}$ ;
while unprocessed data remains do
    $\hat{\beta}_t^{(0)} = \hat{\beta}_{t-1}$;
    $\hat{x}_{1|0} = \hat{x}_{t|t-1}$, $\hat{P}_{1|0} = \hat{P}_{t|t-1}$ ;
    Run Algorithm 1 (using windowed likelihood) to
      get ML estimate $\hat{\beta}_t$ in current window;
    Run Kalman Filter using $\hat{\beta}_t$ ;
    $\hat{x}_{t+1|t}$, $P_{t+1|t}$ ;
    Shift window one step;
    $t \leftarrow t + 1$;
end
```

While not applied here, we note that there are several approaches to improve the numerical robustness of the Kalman filtering and smoothing algorithms such as extending the work in [29] or using an SVD decomposition, see e.g. [37].

D. Convergence

It is well-known that the only requirement needed in the EM algorithm to converge to a stationary point of the likelihood function (which is not necessarily the global maximum) is that $\mathcal{Q}\left(\hat{\beta}^{(i+1)}, \hat{\beta}^{(i)}\right) > \mathcal{Q}\left(\hat{\beta}^{(i)}, \hat{\beta}^{(i)}\right)$. In fact, this is assured to occur when another auxiliary function found in the EM algorithm, known as $\mathcal{H}\left(\beta, \hat{\beta}^{(i)}\right)$, is proven to be non-increasing; for details see e.g. [38].

The next lemma shows that the local likelihood does increase at each step of the EM iteration. Since our algorithm is based on EM, it inherits these same convergence properties. Of course, this also implies we cannot guarantee convergence to the true value because the EM algorithm itself only yields the global maximum under certain special cases [38, ch.3].

Lemma 1: In our scheme, the proposed $l_t(\beta_t)$ defined in (6) has the following property:

$$l_t\left(\hat{\beta}_t^{(i+1)}\right) > l_t\left(\hat{\beta}_t^{(i)}\right).$$

Proof. Following the reasoning in [38], we show that the inclusion of the kernel $K_{k,t}$ does not affect the non-decreasing nature of the function $\mathscr{H}\left(\beta_t, \widehat{\beta}_t^{(i)}\right) : = \mathbb{E}\left\{\sum_{k=1}^{N} K_{k,t} \log p_{\beta}(x_k \mid y_k) \mid Y_N, \widehat{\beta}_t^{(i)}\right\}$. We have for any β_t

$$\begin{aligned}
&\mathscr{H}\left(\beta_t, \widehat{\beta}_t^{(i)}\right) - \mathscr{H}\left(\widehat{\beta}_t^{(i)}, \widehat{\beta}_t^{(i)}\right) = \\
&\mathbb{E}\left\{\sum_{k=1}^{N} K_{k,t} \log p_{\beta_t}(x_k \mid y_k) \mid Y_N, \beta_t^{(i)}\right\} - \mathbb{E}\left\{\sum_{k=1}^{N} K_{k,t}\right. \\
&\left.\log p_{\widehat{\beta}_t^{(i)}}(x_k \mid y_k) \mid Y_N, \widehat{\beta}_t^{(i)}\right\} \\
&= \mathbb{E}\left\{\sum_{k=1}^{N} K_{k,t} \log \frac{p_{\beta_t}(x_k \mid y_k)}{p_{\widehat{\beta}_t^{(i)}}(x_k \mid y_k)} \mid Y_N, \widehat{\beta}_t^{(i)}\right\} \\
&= \sum_{k=1}^{N} K_{k,t} \mathbb{E}\left\{\log \frac{p_{\beta_t}(x_k \mid y_k)}{p_{\widehat{\beta}(i)}(x_k \mid y_k)} \mid Y_N, \widehat{\beta}_t^{(i)}\right\} \\
&\leq \sum_{k=1}^{N} K_{k,t} \log\left[\mathbb{E}\left\{\frac{p_{\beta_t}(x_k \mid y_k)}{p_{\widehat{\beta}_t^{(i)}}(x_k \mid y_k)} \mid Y_N, \widehat{\beta}_t^{(i)}\right\}\right] = 0.
\end{aligned} \tag{16}$$

Using Jenssen's inequality in (16) yields that the expected value is 1, since the expectation is taken with respect to $(x_k | y_k)$. This completes the proof that $\mathscr{H}\left(\beta_t, \widehat{\beta}_t^{(i)}\right)$ is a non-decreasing function, thus guaranteeing that the likelihood in (6) increases at each iteration of the EM algorithm. □

E. Comparison with other approaches

Adaptive filtering techniques have been used for a long time for the estimation of covariances matrices Q and R using the Kalman filter. Literature goes back to the late 60s and early 70s, see e.g. [15], [39]. Here, we make a comparison with [22]. This method uses a retrospective optimization to update the process noise covariance at each time and is based on the minimization of a quadratic function based on the innovations. We take this paper for reference as it claims to have better accuracy than other methods regarded as innovations-based adaptive Kalman filter [15], [19]–[21].

In simulation, we generated trajectories with a change in the diffusion constant from $D_x = 0.1\ \mu m^2/s$ to $D_x = 0.2\ \mu m^2/s$, holding the parameters $a_t = 1$, $b_t = 0$, and $r_t = 0.1$ constant and known. In Fig. 2, we show the comparison between our method using $K(v) = 1$ (naive approach), $K(v)$ as the Epanechnikov kernel, and the method in [22] for the mean of 5 repetitions, considering two window sizes $h = 180, 500$. We observe that our proposed method outperforms both the method of [22] and the naive EM scheme. The inclusion of the kernel plays an important role in smoothing out the estimate, specially when the window size is smaller.

F. Effect of window sizes

There is a clear trade-off between the length of the window size with more sensitivity coming with shorter windows but at the cost of higher variance. To explore this, we performed simulations using the ground-truth values $a_t = 1$, $\beta_t = 0$, and $r_t = 0.01$, with q_t

taking two values (0.1 and 0.2) during the run. We applied our method using short windows ($h = 75, 100$). (Note that even for estimating only q_t, the method in [22] fails to produce an estimate for such short windows.) In Fig. 2, we show the estimate for all the unknowns, and for two different window sizes. The statistics are taken over 20 runs of the same experiment. As expected, we see that shorter windows (blue) the estimates have greater variance longer ones(red) while reacting somewhat faster.

IV. Controlled experiments

A. Generation of data

Here, we test our algorithm with a more realistic situation based on controlled experiments with known ground truth information to generate data sets. The experimental procedure, described in detail in [40], consists of four steps: 1) generate numerical sample paths using the motion model in (1); 2) control the position of a fluorescent microsphere with a nano-positioning stage to follow the sample paths; 3) observe the motion using a widefield fluorescent microscope; 4) do image processing to extract position information from the resulting stack of images. This method effectively provides data sets that exhibit the phenomenon of interest with motions that are accurate, precise, and repeatable while also being subject to real-world effects. Our specific implementation of the controlled experiment, diagrammed in Fig. 3, uses a typical single particle tracking microscope. Realizations using (1) were generated for specific values of a, b, and q_t. The value of r_t was a result of the experimental conditions that include fluorophore brightness, illumination intensity, integration time of the camera and the amount of background light collected by the microscope. These trajectories were then sent to the amplifier of the microscope stage via a National Instruments compact reconfigurable input-output system (NI 9076 cRIO) programmed through the LabView interface. To observe the particle motion, a Zeiss Axiovert 200 equipped with a 63x 1.2 NA water immersion objective and a photometrics 95B sCMOS camera was used to acquire diffraction-limited images at 10 frames per second with an integration time of 50 ms. The microscope stage would move and settle during the time when the camera was off-loading the previous image and the next image would be taken after the stage settled. This yielded a single position per image period. The single particles were then localized using a least squares fitting of the image data to a Gaussian profile with background to the image data. The time series of localizations were then analyzed with local ML.

B. Results

In this section, we demonstrate the performance of our algorithm using experimental data generated using the procedure described in section IV-A. Two dimensional trajectories were generated, modeled as two independent systems of the form (1). The true values of a_t and b_t for both cases were fixed to $a_t = 1$ and $b_t = 0$. The diffusion coefficients began at 0.9 μm^2/s, then switched to 0.1 μm^2/s at time 25 s, and then again back to 0.9 μm^2/s at time 50 s.

The ground truth value used for the creation of the trajectories are shown in Fig. 4 (solid-thin line), as well as the results of the estimation window length $h = 100$ samples. In Fig. 4 (upper-left plot), we show the estimation of a_t which, as expected, moves around the true value of 1; the upper-right plot shows the estimation of b_t, which is around 0; Fig. 4 (lower plot) shows the estimation of the diffusion constant D_t. Plots include the estimation for both x and y-axes.

We also calculate the Mean Square Error (MSE) for each parameter using different window lengths $h = 75, 100, 125$. The rationale behind the selection of these window lengths is that in SPT one typically has a limited number of images due to photobleaching and biological effects, leading to the choice of relatively short windows. In Table I, we can observe that for the fixed parameters MSE values decrease as the window h increases for a and b, which is expected. However, the estimation of the time-varying diffusion coefficient actually gets worse as the window length increases. This is because the change is abrupt and the longer windows imply a longer time for the old data to move out of the window. Thus, there is a clear trade-off between accuracy and capacity to follow the change, having several possibilities for addressing this, such as adaptive windows or combining a forward and a backward pass. Resolving this is left to future work.

V. CONCLUSIONS

In this paper, we have developed a local likelihood estimation algorithm to track time-varying parameters in a general model that captures several common motion paradigms used in SPT. Our approach uses local likelihood in a sliding window, which is well-known in the statistics literature. We generated data using a controlled real experimental setup, and our approach was demonstrated using this generated data.

While focused on pure linear systems with additive Gaussian noise on the trajectory, the basic scheme can represent a wide variety of useful models found in SPT. One way to extend our work is by considering more complex models by replacing the underlying linear filtering and smoothing with nonlinear techniques such as sequential Monte Carlo filtering. There are also clear questions about the optimal way to select window sizes, and the effect of measurements noise and modelling errors.

ACKNOWLEDGMENT

This work was supported in part by the NIH through grant NIGMS 5R01GM117039-02.

References

[1]. Shen H, Tauzin LJ, Baiyasi R, Wang W, Moringo N, Shuang B, and Landes CF, “Single Particle Tracking: From Theory to Biophysical Applications,” Chemical Reviews, vol. 117, no. 11, pp. 7331–7376, 6. 2017. [PubMed: 28520419]

[2]. Babcock HP, Chen C, and Zhuang X, “Using Single-Particle Tracking to Study Nuclear Trafficking of Viral Genes,” Biophysical Journal, vol. 87, no. 4, pp. 2749–2758, 10. 2004. [PubMed: 15454466]

[3]. Vermehren-Schmaedick A, Krueger W, Jacob T, Ramunno-Johnson D, Balkowiec A, Lidke KA, and Vu TQ, “Heterogeneous Intracellular Trafficking Dynamics of Brain-Derived Neurotrophic Factor Complexes in the Neuronal Soma Revealed by Single Quantum Dot Tracking,” PLoS ONE, vol. 9, no. 4, p. e95113, 4. 2014. [PubMed: 24732948]

[4]. Monnier N, Guo S-M, Mori M, He J, Lénárt P, and Bathe M, “Bayesian Approach to MSD-Based Analysis of Particle Motion in Live Cells,” Biophy. Journal, vol. 103, no. 3, pp. 616–626, 2012.

[5]. Berglund AJ, “Statistics of camera-based single-particle tracking,”Physical Review E, vol. 82, no. 1, p. 011917, 7. 2010.

[6]. Michalet X, “Mean square displacement analysis of single-particle trajectories with localization error: Brownian motion in an isotropic medium,” Physical Review E, vol. 82, no. 4, p. 041914, 10. 2010.

[7]. Michalet X. and Berglund AJ, “Optimal diffusion coefficient estimation in single-particle tracking,” Physical Review E, vol. 85, no. 6, p. 061916, 6. 2012.

[8]. Boyer D, Dean DS, Mejía-Monasterio C, and Oshanin G, “Optimal least-squares estimators of the diffusion constant from a single Brownian trajectory,” The European Physical Journal Special Topics, vol. 216, no. 1, pp. 57–71, 1. 2013.

[9]. Calderon CP, “Motion blur filtering: A statistical approach for extracting confinement forces and diffusivity from a single blurred trajectory,” Physical Review E, vol. 93, no. 5, p. 053303, 5 2016. [PubMed: 27301001]

[10]. Vega AR, Freeman S, Grinstein SA, and Jaqaman K, “Multistep track segmentation and motionclassification for transient mobility analysis,” Biophysical Journal, vol. 114, pp. 1018–1025, 2018. [PubMed: 29539390]

[11]. Ashley TT and Andersson SB, “A Sequential Monte Carlo Framework for the System Identification of Jump Markov State Space Models,” in American Control Conference, 2014, pp. 1144–1149.

[12]. Godoy BI, Lin Y, Agüero JC, and Andersson SB, “A 2-step algorithm for the estimation of single particle tracking models using maximum likelihood,” in 12th Asian Control Conf. (ASCC), Kitakyushu, Japan, 2019.

[13]. Loader C, Local Regression and Likelihood. Springer, 1999.

[14]. Fan J. and Gijbels I, Local Polynomial modelling and its Applications. Chapman& Hall, CRC, 1996.

[15]. Mehra R, “Approaches to adaptive filtering,” IEEE Trans. on Aut. Control, vol. 17, no. 5, pp. 693–698, 1972.

[16]. Zagrobelny M. and Rawlings JB, “Identifying the uncertainty structure using maximum likelihood estimation,” in American Control Conf., Boston, MA, USA, 2016.

[17]. Matisko P. and Havlena V, “Noise covariance estimation for Kalman filter tuning using bayesian approach and Monte Carlo,” Int. J. Adaptive Cont. Signal Proc, vol. 27, no. 11, pp. 957–973, 2013.

[18]. Odelson BJ, Rajamani MR, and Rawlings JB, “A new auto-covariance least-squqres methods for estimating noise covariances,” Automatica, vol. 42, no. 2, pp. 303–308, 2006.

[19]. Almagbile A, Wang J, and Ding W, “Evaluating the performances of adaptive Kalman filter methods in GPS/INS integration,” Journal of G.P.S, vol. 9, no. 1, pp. 33–40, 2010.

[20]. Hide C, Moore T, and Smith M, “Adaptive filtering for low-cost INS/GPS,” The Journal of Nav, vol. 56, no. 1, pp. 143–152, 2003.

[21]. Mohamed AH and Schwarz KP, “Adaptive Kalman fitering for INS/GPS,” Journal of Geodesy, vol. 73, pp. 193–203, 1999.

[22]. Sobolic F. and Bernstein DS, “Kalman-filter-based time-varying parameter estimation via retrospective optimization of the process noise covariance,” in American Con. Conf., Boston, MA, USA, 2016.

[23]. Cheezum MK, Walker WF, and Guilford WH, “Quantitative comparison of algorithms for tracking single fluorescent particles.” Biophysical Journal, vol. 81, no. 4, p. 2378, 2001. [PubMed: 11566807]

[24]. Andersson SB, “Localization of a fluorescent source without numerical fitting,” Optics Express, vol. 16, no. 23, pp. 18 714–18 724, 1. 2008.

[25]. Small AR and Parthasarathy R, “Superresolution Localization Methods,” dx.doi.org, vol. 65, no. 1, pp. 107–125, 4. 2014.

[26]. van Kampen NG, Stochastic Processes in Physics and Chemistry. Elsevier, 1992.

[27]. Ashley TT and Andersson SB, “Method for simultaneous localization and parameter estimation in particle tracking experiments,” Physical Review E, vol. 92, no. 5, p. 052707, 11. 2015.

[28]. Wills A, Schön T, and Ninness B, "Estimating state-space models in innovations form using the expectation maximisation algorithm," in IEEE Conf. Dec. and Control (CDC), Atlanta, GA, USA, 2010.

[29]. Gibson S. and Ninness B, "Robust maximum-likelihood estimation of multivariable dynamic systems," Automatica, vol. 41, no. 10, pp. 1667–1682, 2005.

[30]. Agüero JC, Yuz JI, Goodwin GC, and Delgado RA, "On the equivalence of time and frequency domain maximum likelihood estimation," Automatica, vol. 46, no. 2, pp. 260–270, 2010.

[31]. Marelli DE, Godoy BI, and Goodwin GC, "A scenario-based approach to parameter estimation in state-space models having quantized output data," in 49th IEEE Conf. on Dec. and Control (CDC), Atlanta, GA, USA, 2010.

[32]. Dempster AP, Laird NM, and Rubin DB, "Maximum likelihood from incomplete data via the EM algorithm," Journal of the Royal Statistical Society. Series B, vol. 39, no. 1, pp. 1–38, 1977.

[33]. Shumway RH and Stoffer DS, Time Series Analysis and Its application, 2nd ed. Springer, 2006.

[34]. Hewitt E. and Hewitt RE, "The Gibbs-Wilbraham phenomenon: An episode in Fourier analysis," Archive of *history of exact sciences*, vol. 21, no. 2, pp. 129–160, 1979.

[35]. Anderson BDO and Moore JB, Optimal Filtering. Prentice-Hall, Inc., 1979.

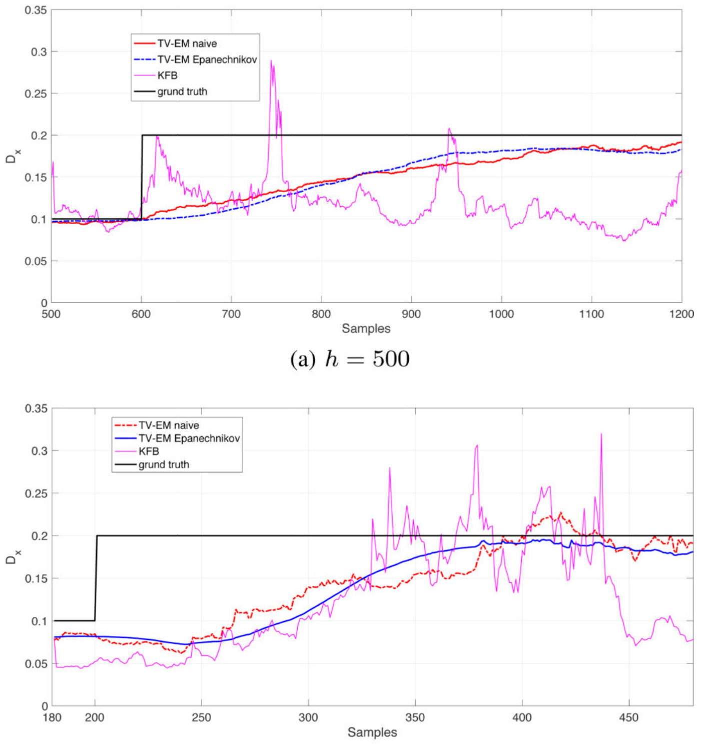
[36]. Long CJ, Brown EN, Triantafyllou C, Wald LL, and Solo V, "Nonstationary noise estimation in functional MRI," Neuroimage, vol. 28, pp. 890–093, 2005. [PubMed: 16129625]

[37]. Kulikova MV and Tsyganova J, "Improved discrete-time Kalman filtering within singular value decomposition," IET Control Theory & Appls, vol. 11, no. 15, pp. 2412–2418, 2017.

[38]. McLachlan GJ and Krishnan T, The EM algorithm and its extensions, 2nd ed. John Wiley& Sons, 2008.

[39]. Mehra RK, "Identification of stochastic linear dynamic systems," in 1969 IEEE Symp. on Adaptive Processes (8th) CDC, Univ. Park, PA, USA, 1969.

[40]. Vickers NA and Andersson SB, "Monte carlo simulation of brownian motion using a piezo-actuated microscope stage," in American Control Conf., Phi, PA, 2019.

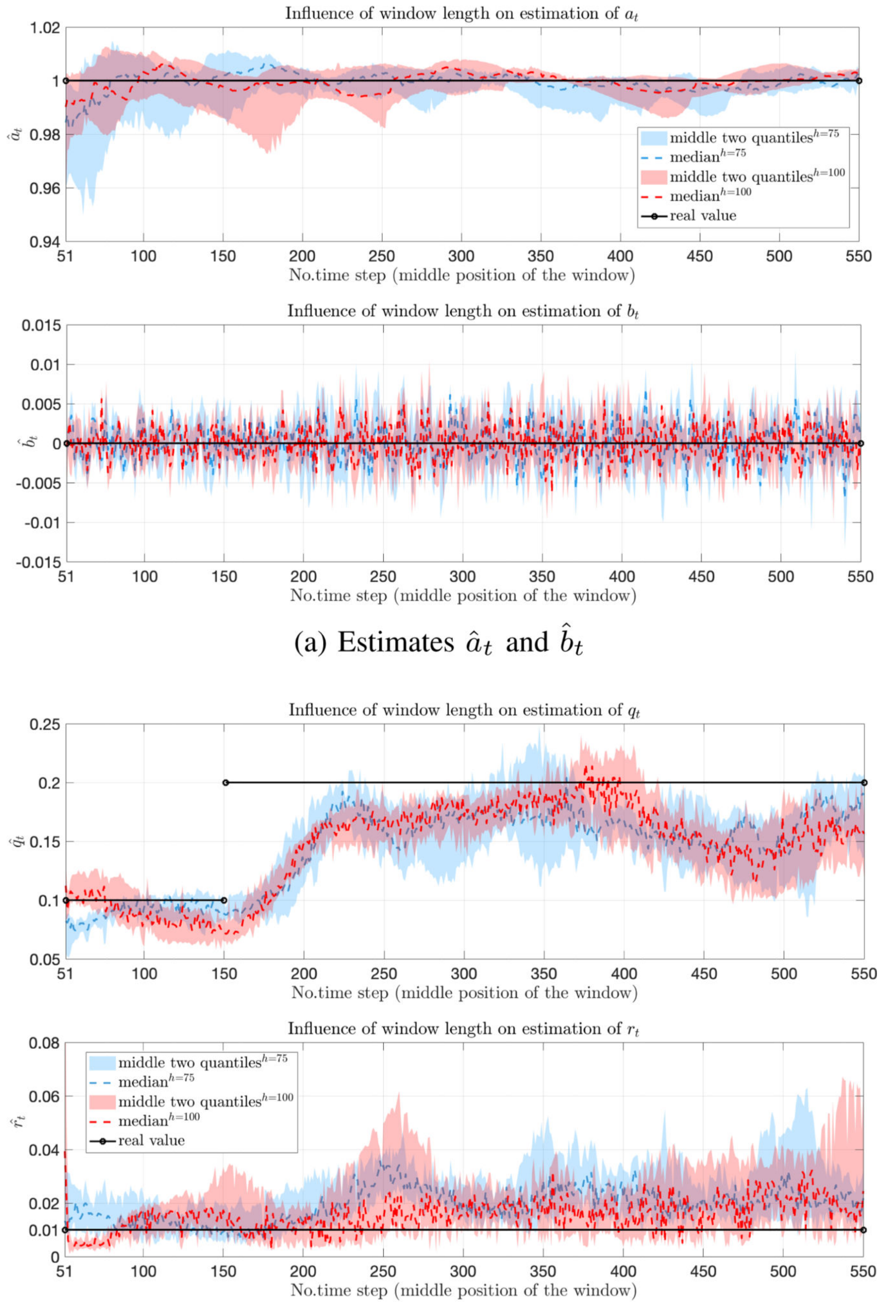


(a) $h = 500$

(b) $h = 180$

Fig. 1.

Comparison of our method with Epanechnikov kernel, no kernel, and innovation-based method [22] (KFB). Measurement noise $r_t = 0.1$



(a) Estimates $\hat{a}_t$ and $\hat{b}_t$

(b) Estimates $\hat{q}_t$ and $\hat{r}_t$

Fig. 2.

Comparison of estimates $\hat{a}_t, \hat{b}_t, \hat{q}_t, \hat{r}_t$ using two different window sizes $h = 75$ (blue) and $h = 100$ (red).

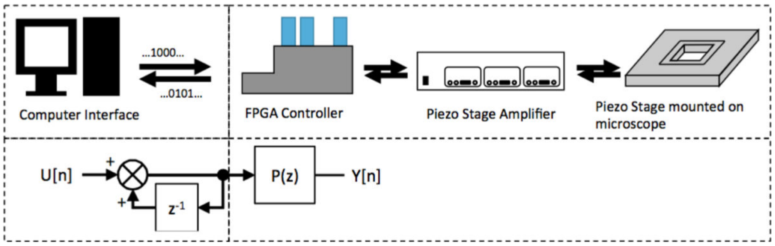
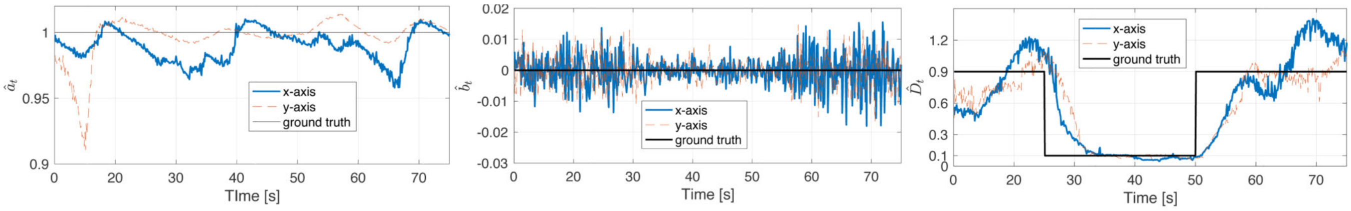


Fig. 3.

A block diagram showing the signal flow. (Upper left) computer generated realizations of our model (bottom) is sent to the controller and is translated to an analog control voltage. This control voltage is sent to the stage amplifier which then generates the high voltage needed to generate motion in the stage.



(a) Estimation of a

(b) Estimation of b

(c) Estimation of D

Fig. 4.

Estimation of the unknown parameters in x (blue-solid) and y (red-dashed) axis of controlled experiments.

TABLE I

MSE VERSUS WINDOW LENGTH h

	h **= 75**		***h*** **= 100**		***h*** **= 125**	
parameter	***x*-axis**	***y*-axis**	***x*-axis**	***y*-axis**	***x*-axis**	***y*-axis**
a_t	.0197	.0297	.0157	.0157	.0133	.0100
b_t	.0048	.0044	.0043	.0042	.0036	.0033
$D_t = \frac{q_t}{2\Delta t}$	.0526	.0507	.0633	.0617	.0703	.0789