

FAST DYNAMIC NONPARAMETRIC DISTRIBUTION TRACKING IN ELECTRON MICROSCOPIC DATA

BY YANJUN QIAN^{*}, JIANHUA Z. HUANG[†], CHIWOO PARK[‡] AND YU DING[†]

Virginia Commonwealth University^{*}

Texas A & M University[†]

Florida State University[‡]

In situ transmission electron microscope (TEM) adds a promising instrument to the exploration of the nanoscale world, allowing motion pictures to be taken while nano objects are initiating, crystalizing and morphing into different sizes and shapes. To enable in-process control of nanocrystal production, this technology innovation hinges upon a solution addressing a statistical problem, which is the capability of online tracking a dynamic, time-varying probability distribution reflecting the nanocrystal growth. Because no known parametric density functions can adequately describe the evolving distribution, a nonparametric approach is inevitable. Towards this objective, we propose to incorporate the dynamic evolution of the normalized particle size distribution into a state space model, in which the density function is represented by a linear combination of B-splines and the spline coefficients are treated as states. The closed-form algorithm runs online updates faster than the frame rate of the *in situ* TEM video, making it suitable for in-process control purpose. Imposing the constraints of curve smoothness and temporal continuity improves the accuracy and robustness while tracking the probability distribution. We test our method on three published TEM videos. For all of them, the proposed method is able to outperform several alternative approaches.

1. Introduction. The nanoparticle self-assembly process produces nanocrystals from small building blocks such as atoms and molecules that are spontaneously arranged into order structures at the nanoscale. It is considered a promising method of producing nanocrystals in large quantities (Li et al., 1999; Boal et al., 2000). To produce nanocrystals with desired sizes and shapes, its growth process should be monitored and controlled (Grzelczak et al., 2010), but accomplishing this goal is rather challenging, due to the existence of multiple growth mechanisms (Zheng et al., 2009), complex interactions among hundreds of nanoscale particles (Park et al., 2015), and after all, the stochastic nature of the growth processes. Critical to the mission of achieving in-process control is a recent technology innovation in nanoscale metrology, the *in situ* transmission electron microscope (TEM) (Zheng et al., 2009). An *in situ* TEM uses a special sample holder in which a nanocrystal growth process takes place, allowing motion pictures to be taken while the nanocrystals in the sample holder are initiating, crystalizing, and morphing into different sizes and shapes.

The morphological features extracted from a TEM video are the sizes and shapes of the nanocrystals and their evolving trajectories over the time. In this study, we focus primarily

Keywords and phrases: Kalman filter, nanotechnology, nonparametric density, online density estimation, state space model

on particle size, because all the TEM videos we have at hand contain nanocrystals of rather uniformly round shape throughout their growth process. We note that the current progress by research communities in handling *dynamic* TEM images (i.e., videos) is still at the stage of dealing with size, rather than both size and shape.

When an image frame of the process is recorded, an image processing tool is used to extract the contours of the nanocrystals in the frame, count the quantity, and calculate the particle sizes. After that, an estimate of the normalized particle size distribution (NPSD) is created and used as the observational input to the subsequent modeling. Here, the NPSD is the original particle size distribution normalized by the average radius of the nanocrystals at a given moment. Studies show that NPSD provides a better indicator than the average absolute size to anticipate and detect phase change point in nanocrystal growth (Zheng et al., 2009; Woehl et al., 2013; Qian et al., 2017). Research has been conducted by domain experts to shed insights on asymptotic solutions of NPSD under certain growth mechanisms (Lifshitz and Slyozov, 1961; Aldous, 1999).

Understanding and modeling the evolution of NPSD appears to be an enabling prerequisite to process control of a nanocrystal growth process. Estimating NPSD using particle size data extracted from TEM images is a statistical problem of probability density estimation. Empirical analyses in Zheng et al. (2009) and Woehl et al. (2013) demonstrate that the density function of NPSD can change from a multi-modal, asymmetric function in the early stages of growth to a uni-modal, symmetric one in the late stages. Since it is hard to specify a parametric function of NPSD to adequately describe different growth mechanisms in a multi-stage growth processes (Lifshitz and Slyozov, 1961; Aldous, 1999), a nonparametric approach appears inevitable.

Direct application of standard nonparametric methods, however, does not produce good density estimation, due to the fact that too few nanocrystals are available in the image at a single time frame of the TEM video. To overcome the small sample size problem, Qian et al. (2017) observe that the NPSD changes gradually over time during each nanocrystal growth stage, so one can borrow information across time frames to obtain a more reliable density estimation. Following Eilers and Marx (1996), they model the log density function at each time frame as a linear combination of B-spline basis functions and employ the penalized Poisson likelihood of binned data with a smoothness penalty. The formulation of Eilers and Marx (1996) is further extended in Qian et al. (2017) by pooling the log-likelihoods from all time frames together and including a second penalty to ensure that the estimated density functions have certain degree of “temporal continuity.” As TEM videos are captured at discrete times, here the temporal continuity means that the NPSD evolves gradually from one image frame to the next, so that the estimated NPSDs at neighboring image frames should be close to each other, especially so when two frames are in the same growth stage. Using the extended formulation embedding two smoothness penalties, Qian et al. (2017) estimate the time varying NPSD density functions, via a modified alternating direction method of multipliers (ADMM) algorithm that efficiently solves the resulting optimization problem.

The method developed in Qian et al. (2017) is a retrospective approach conducted offline. It uses the observations from all video frames and minimizes a loss function embodying all available data. Should this retrospective approach be applied to an online application, it ought to solve the

optimization problem whenever a new image frame (or a couple of new images) comes. Doing so takes more time than allowed in a real-time processing. For instance, it takes a couple of minutes to solve the optimization problem using the algorithm in [Qian et al. \(2017\)](#) for the 76-second video clip (Video 1, Section 2), not a big deal for offline analysis but not practical for online applications. The retrospective approach is hence inefficient and ill suited for online applications. The goal of the current paper is to develop a prospective method by furthering the development in [Qian et al. \(2017\)](#).

Before we discuss our contributions in this paper, we would like to emphasize that there is a strong need for a prospective method for real-time processing, because our goal is online monitoring and tracking, and only through this online capability does it enable in-process control of an ongoing process. Since the nanocrystal growth process is stochastic and volatile, it is difficult to foresee control opportunities ahead of time. When a control opportunity presents itself, one needs to react rapidly, as the window of opportunity may not stay long in such stochastic dynamic environment. A retrospective method does not enable process control capability since one waits for the process to complete and by then the control opportunity is long gone. For an online analysis, the model updating to capture the NPSD change needs to be fast enough; how fast is enough is dictated by the imaging speed (in Video 1 of Section 2, about 15 frames per second).

Same as in [Qian et al. \(2017\)](#), we estimate the NPSD density at each time frame by smoothing the histogram data using penalized B-splines. Departing from [Qian et al. \(2017\)](#), we characterize the dynamics of the NPSD using a state space model in which the spline coefficient vector at each time frame is treated as the hidden state and its time evolution is modeled through a random walk. The random walk state equation naturally ensures the temporal continuity among the states. To ensure the estimated density function at each time frame to be a smooth function, we introduce a new state vector that encodes the second order differences of the spline coefficients at each time frame, thus achieving smoothness by controlling the magnitude of time increment of such differences. To deal with the non-Gaussian nature of our state space model, we develop an iterative local Gaussian approximation, in the form of an extended Kalman filter ([Ljung, 1979](#)), and derive a closed form state-updating equation for updating the time-varying NPSD when new observations come. Our algorithm is fast enough to catch up with the imaging rate for the TEM videos that are available to us. The imposition of the two constraints on B-splines alleviates overfitting and reduces the estimation sensitivity to observational variability. [Figure 1](#) presents an overview of the proposed online, prospective analysis.

Related to this work, there are two main branches of research: (1) nano image processing, (2) estimation of time-varying nonparametric density functions.

In the first branch of literature, the vast majority of the existing methods for analyzing TEM measurements, including several of our own, are for handling still images ([Park et al., 2012, 2013](#); [Muneesawang and Sirisathitkul, 2015](#); [Qian et al., 2016](#)). These methods laid the foundation for handling dynamic images in TEM videos. One can even use them to process the images one frame at a time. Of course, processing one frame at a time is inefficient and also overlooks the dynamics and correlation among the adjacent video frames. A few approaches are available for handling dynamic TEM images, including [Qian et al. \(2017\)](#), but the current approach, with exception of

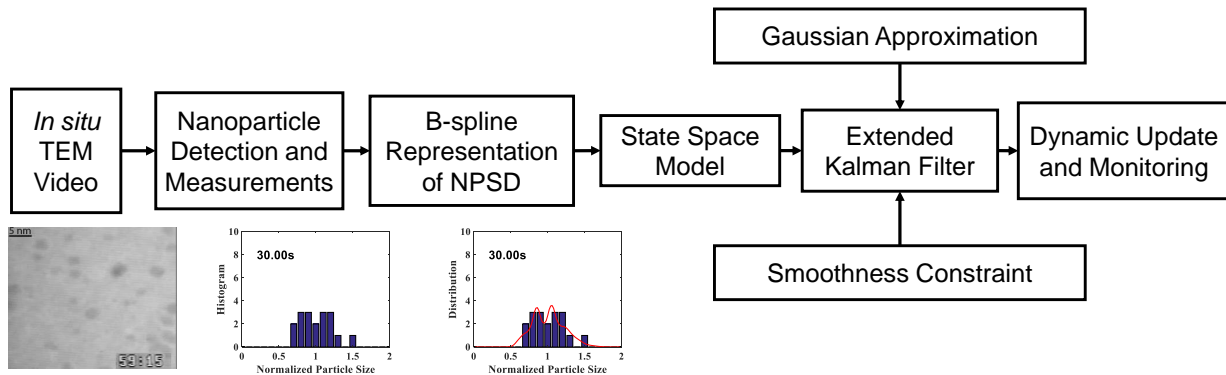


FIG 1. The framework of a prospective analysis of *in situ* TEM videos.

Qian et al. (2017), is to identify and track individual nanocrystals (Park, 2014; Park et al., 2015) and characterize the growth dynamics by looking at the trajectories of individual nanocrystals. This object-tracking approach assumes traceability of nanocrystals across image frames, which may not be possible in practice.

Among the second branch of literature, Rodriguez and Ter Horst (2008), and Mena and Ruggiero (2016) studied a dynamic hierarchical model for time-varying distributions. Under their framework, the time-varying distribution follows a Dirichlet mixture prior (Lo, 1984), the parameters of the Dirichlet processes change over time according to a time series model, and posterior sampling is used for inference. Another set of papers (Ma et al., 2008; Zhang et al., 2017), which develop the dynamic multivariate count data model, is conceptually useful for our work, too, because of the connection between the estimation of density functions and the estimation of intensities of Poisson counts in a histogram (Eilers and Marx, 1996). However, the sampling-based solution approaches advocated in those works are computationally slow, hence ill-equipped to satisfy the online model updating objective (recall the 15 frames per second imaging rate).

The remaining parts of the article are organized as follows. In Section 2, we discuss the data used in this study. In Section 3, we present the state space model and devise an extended Kalman filter for online updating and tracking of the particle size distribution. In Section 4, we explain how to estimate the parameters used in the state space model. In Section 5, we apply our method to analyze three segments of TEM videos and demonstrate the merits of the proposed method. Finally, we conclude our work in Section 6.

2. Data. As a newly emerged technology and rather expensive, there are not many *in situ* TEMs available yet in the United States. There are a very limited number of TEM videos available in the public domain. In this study, we use three clips of *in situ* TEM video: two clips published by Zheng et al. (2009) and one clip published by Woehl et al. (2013). The three videos clips capture, respectively, 76.6, 42.5, and 112 seconds of a nanocrystal growth process, and there are 1,149, 637, and 112 image frames in the respective clip. We label them as Video 1, Video 2 and Video 3, respectively. Figure 2 presents four frames of Video 1, capturing the growth of platinum nanocrystals.

The data processing works as follows. When an image frame of the nanocrystal growth process is recorded by an *in situ* TEM, we first process the image and extract the nanocrystal information, which is the number and the corresponding size of the nanocrystals in the frame. The specific tool for processing individual images is from Qian et al. (2016), a method particularly potent of handling noisy TEM images with low contrast. The result of one frame from each video clip is shown in Figure 3. Of the three video clips, Video 1 and 2 are of 290×242 pixels in size and Video 3 is of 496×472 pixels. Considering their relatively small image size, the image pre-processing can be done fairly quickly. For Video 1 and Video 2, the image processing takes only 0.04 seconds per frame and for Video 3 it takes 0.2 seconds per frame.

After all nanocrystals in the frame of time t are detected, we calculate $A^\ell(t)$, the area of the ℓ -th nanocrystal at time t , for $\ell = 1, \dots, N_t$, where N_t is the total number of nanocrystals in the frame of time t . Following Woehl et al. (2013), we use $A^\ell(t)$ to compute the average radius of the ℓ -th nanocrystal, namely $r^\ell(t) = \sqrt{A^\ell(t)/\pi}$, to represent the size of each nanocrystal. The mean radius for each image frame, $\bar{r}(t)$, can be readily obtained. Finally, we normalize $r^\ell(t)$ by $\bar{r}(t)$ to obtain the normalized radius $x^\ell(t)$, such that $x^\ell(t) = r^\ell(t)/\bar{r}(t)$. We note that here “size” is represented by the radius, so that the normalized particle size distribution is in fact the normalized particle radius distribution. Such modeling choice follows the domain science’s convention and treatment (Aldous, 1999; Lifshitz and Slyozov, 1961; Woehl et al., 2013).

To facilitate the subsequent computation in estimation and updating, we bin the observations to create a histogram and then use the histogram as the input to the dynamic state space model. We limit the range of $x^\ell(t)$ to $[0, 2.0]$, as the nanocrystals twice as large as the average size are very few at any given time. We divide the range into m intervals of equal size δ . Here we use a constant $m = 21$ throughout the monitoring process and denote by x_i the normalized particle size corresponding to the center of the i th interval, $i = 1, \dots, m$. We will further elaborate in Section 3.1 the reasons behind binning the observations and conduct in Section 5.2 a sensitivity analysis on the number of intervals used in the input histogram. The resulting histogram for the frame of time t is denoted by the vector of $\mathbf{Y}_t = [Y_{1t}; Y_{2t}; \dots; Y_{mt}]$, where Y_{it} is the number of the observed $x^\ell(t)$ ’s falling into the i th interval of the histogram.

3. State Space Modeling and Updating. Our primary objective is to estimate the probability density function, $f_t(x)$, of the normalized particle size using $x^\ell(t)$, available up to time t .

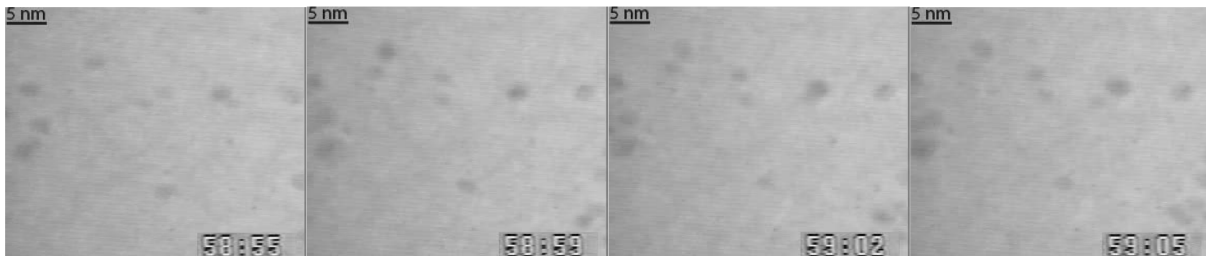


FIG 2. Four frames from the *in situ* TEM video studied by Zheng et al. (2009). The dark spots are nanocrystals.

As mentioned in Section 1, direct application of the retrospective method developed in Qian et al. (2017) is inefficient and does not serve the purpose of real-time processing. In this section, we develop a computationally efficient state space modeling approach that allows us to update the density estimation at time $t + 1$ using the density estimation at time t and the new data observed at time $t + 1$.

3.1. *State Space Model for Normalized Particle Size Distribution.* Since it is difficult to find a parametric method to be adaptive enough to model various types of NPSDs at different growth stages of nanocrystals, we naturally resort to a nonparametric method for density estimation. Among many available nonparametric methods for density estimation, we follow the procedure in Eilers and Marx (1996), which presents a sufficiently flexible spline representation of $f_t(x)$ by fitting a Poisson model to the histogram data \mathbf{Y}_t . This modeling choice is made mainly because doing so allows us to implement a real-time estimation algorithm on the time-varying distribution.

As Y_{it} is the count of observations falling in $[x_i - \delta/2, x_i + \delta/2]$, it is a standard approach to assume that Y_{it} follows the Poisson distribution with expectation of λ_{it} (Bishop et al., 1975):

$$(3.1) \quad Y_{it} \sim \text{Poisson}\{\lambda_{it}\}, \quad i = 1, \dots, m.$$

Following the treatment in Eilers and Marx (1996), we model the count data with B-splines. We adopt a generalized linear model with a log link function to represent λ_{it} as:

$$(3.2) \quad \log \lambda_{it} = \sum_{j=1}^n \alpha_{jt} B_j(x_i),$$

where n is the number of basis functions, $B_j(x)$ is the j th B-spline basis function, and α_{jt} is its coefficient at time t . Using the log link function can guarantee a positive λ_{it} , which is needed in our application. Collectively, $[\alpha_{1t}; \alpha_{2t}; \dots; \alpha_{nt}]$ can be represented as a vector of $\boldsymbol{\alpha}_t$. If we write the B-spline basis functions as a matrix \mathbf{B} , such that $(\mathbf{B})_{ij} = B_j(x_i)$, the Poisson model of Y_{it} is then expressed as:

$$(3.3) \quad Y_{it} \sim \text{Poisson}\{(\exp[\mathbf{B}\boldsymbol{\alpha}_t])_i\}, \quad i = 1, \dots, m.$$

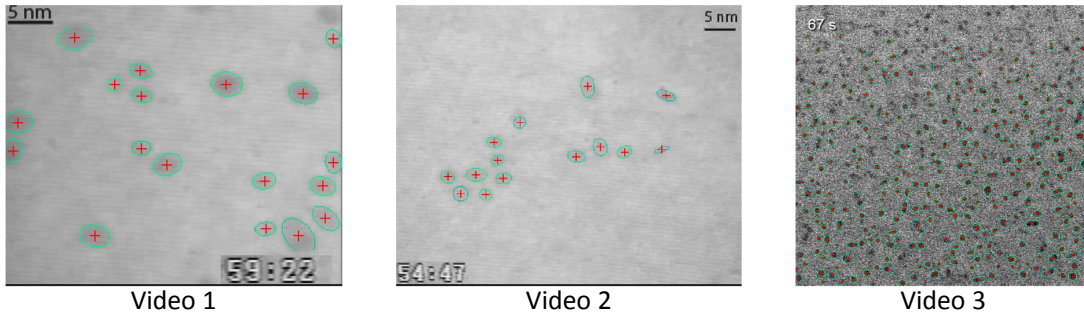


FIG 3. The nanocrystal detection results of a single frame, one each from the three TEM video clips. The green line shows a nanocrystal's edge and the red '+' shows a nanocrystal's center. Videos 1 and 2 were published by Zheng et al. (2009) and Video 3 was published by Woehl et al. (2013).

Equation (3.3) is referred to as a Poisson-exponential model.

The maximum likelihood method can be used to estimate the spline coefficient vector α_t . Once α_t is estimated, we can further obtain the normalized particle size distribution $f_t(x)$ as follows. As $\lambda_{it} \propto \int_{x_i-\delta/2}^{x_i+\delta/2} f_t(x)dx$, $\lambda_{it} \propto f_t(x_i)$ when δ is small, so that $f_t(x)$ can be estimated by the continuous form of Equation (3.2) as

$$(3.4) \quad f_t(x) = \frac{1}{C_t(\alpha_t)} \exp \left[\sum_{j=1}^n \alpha_{jt} B_j(x) \right],$$

where $C_t(\alpha_t)$ is a normalizing constant to guarantee $f_t(x)$ integrating to one. To ensure the smoothness of the estimated density function, Eilers and Marx (1996) proposed to use a small δ and use a penalized Poisson likelihood with a smoothness penalty that has the form of sum of squared second-order differences of the spline coefficients.

The above approach adopted from Eilers and Marx (1996) is an approximation and simplification of the standard approach of directly fitting the raw data to the model in (3.4), which is through the maximization of the log-likelihood of

$$\sum_{\ell} \log f_t(x^{\ell}(t)) = \sum_{\ell=1}^{N_t} \sum_{j=1}^n \alpha_{jt} B_j(x^{\ell}(t)) - N_t \log C_t(\alpha_t).$$

By applying the penalized Poisson likelihood to the binned data, the approach of Eilers and Marx (1996) can avoid computing the integral in the normalizing constant,

$$(3.5) \quad C_t(\alpha_t) = \int_x \exp \left[\sum_{j=1}^n \alpha_{jt} B_j(x) \right] dx.$$

This simplification turns out to be critical for us to develop an online updating algorithm. When a small δ is used, the binning action does not cause much information loss as compared to using the original observations of $x^{\ell}(t)$, and the smoothness penalty helps prevent overfitting. To focus on developing an online algorithm, we postpone our discussion regarding the smoothness penalty to Section 3.3.

With the B-spline representation in place, we can use the B-spline coefficient vector, α_t , as the state vector in the proposed state space model, because the change in α_t indicates the change of the underlying normalized particle size distribution $f_t(x)$. Previous studies (Lifshitz and Slyozov, 1961; Aldous, 1999) show that $f_t(x)$ undergoes small fluctuations during a growth stage in which the commanding physical growth mechanism remains the same, while $f_t(x)$ will see a much greater change as a different growth mechanism takes over. Based on this understanding, we assume that the state vector, α_t , follows a random walk model

$$(3.6) \quad \alpha_t = \alpha_{t-1} + \mathbf{w}_t,$$

where the innovation, \mathbf{w}_t , is the disturbance vector of the state and assumed to follow the distribution of normal($\mathbf{0}$, \mathbf{Q}). The covariance matrix \mathbf{Q} will be treated as a constant matrix throughout the process. The state updating equation (3.6) and the observation equation (3.3) constitute our

state space model, which will serve as the basis for developing an online updating algorithm using an extended Kalman filter.

Our use of the random walk model is based on a trade-off between flexibility and tractability. We would like to choose a model that is flexibility enough but is still tractable. The main concern for tractability is fast computation that allows real time online processing. Although using a stationary time series process such as ARMA in the state space model is possible, real time updating of both the ARMA parameters and the state variables is a daunting task for online processing. Moreover, it may be tempting to build the change point/regime transition into our model, but this certainly hurts the tractability.

We found that the simple random walk model can cope very well with the nanocrystal growth process. Within a specific stage of nanocrystal growth, the normalized particle size distribution undergoes small fluctuations, and this is well captured by the random walk model. In a transition period moving from one growth stage to another, the random walk model serves as a prior distribution on the state variables, and the Kalman filter is able to adapt to structure changes by calculating the posterior after receiving new observations. When the Kalman filter outputs a large value in the innovation series, it signals possible growth mechanism changes, providing a simple way of identifying change points; see Figure 7 presented later for an illustration.

3.2. Online Updating of State α_t . Updating the estimation of NPSD is thus equivalent to updating the state vector in the state space model. In the dynamic systems and control theory, the Kalman filter (Kalman, 1960) is arguably the most popular method used for conducting such update. For linear state space models with Gaussian observations, a Kalman filter (Kalman, 1960) uses the posterior mean $E(\alpha_t | \mathbf{Y}_1, \dots, \mathbf{Y}_t)$, denoted as $\hat{\alpha}_t$, to iteratively estimate α_t . There are two main steps in a Kalman filter. The first step, known as prediction, is to predict the prior mean, $\hat{\alpha}_t^-$, and the prior covariance matrix, \mathbf{P}_t^- , of the state at time t , based on the observations received up to time $t - 1$. When the new observation of \mathbf{Y}_t arrives, the Kalman filter undertakes a correction step to obtain the posterior mean, $\hat{\alpha}_t$, and the posterior covariance matrix, \mathbf{P}_t . For a Gaussian system, the Kalman filter has a closed-form solution for both prediction and correction steps and can thus run very efficiently.

Unfortunately, our state space model of the time-varying NPSD is not a Gaussian system since \mathbf{Y}_t follows a Poisson distribution with an exponential link function in Equation (3.3). To solve for the posterior mean $E(\alpha_t | \mathbf{Y}_1, \dots, \mathbf{Y}_t)$, one possible solution approach is to use sampling methods, such as particle filtering (Doucet et al., 2001; Ma et al., 2008; Zhang et al., 2017), to simulate the posterior distribution of the state α_t . But the sampling approach is not ideal for online estimation because the approach's computational speed can hardly meet the online updating requirement. After knowing the model set up, the shortcoming of the sampling approach is even more obvious. To estimate the NPSD accurately, both \mathbf{Y}_t and α_t should have a moderate to high dimension; for instance, $m \geq 10$ and $n \geq 10$. To sample from a space of such dimension for approximating a posterior distribution, the sample size are rather large, making its computational efficiency a daunting task to be addressed.

Our solution is to extend the Kalman filter by adopting Durbin and Koopman (1997)'s method

to find a good Gaussian approximation, locally around the current estimation $\hat{\boldsymbol{\alpha}}_t$, of the Poisson observation in Equation (3.3). Rather than approximating the Poisson distribution globally, a Gaussian distribution can have a similar shape as the Poisson within the neighborhood of $\hat{\boldsymbol{\alpha}}_t$. As the Kalman filter usually updates $\boldsymbol{\alpha}_t$ near its current position, such a local approximation can lead to an efficient and accurate estimation. When used in our context, this means that we want to have the following approximation:

$$(3.7) \quad \mathbf{Y}_t \sim \text{normal}(\mathbf{B}\boldsymbol{\alpha}_t + \boldsymbol{\mu}_t, \mathbf{H}_t),$$

so that the probability density functions of Equations (3.3) and (3.7) have the same first and second derivatives with respect to $\boldsymbol{\alpha}_t$ near $\hat{\boldsymbol{\alpha}}_t$. Following this thought, we can derive the following expressions for the mean vector $\boldsymbol{\mu}_t$ and the covariance matrix \mathbf{H}_t (please see the derivation details in Appendix A):

$$(3.8) \quad \begin{aligned} \boldsymbol{\mu}_t &= \mathbf{Y}_t - \mathbf{B}\hat{\boldsymbol{\alpha}}_t - \exp(-\mathbf{B}\hat{\boldsymbol{\alpha}}_t)[\mathbf{Y}_t - \exp(\mathbf{B}\hat{\boldsymbol{\alpha}}_t)], \\ \mathbf{H}_t &= \text{diag}[\exp(-\mathbf{B}\hat{\boldsymbol{\alpha}}_t)]. \end{aligned}$$

As such, the original state space model is converted into an approximated Gaussian state space model, now constituting of Equation (3.7) and Equation (3.6). Technically, a standard Kalman filter can then be devised and applied.

A remaining problem is that $\hat{\boldsymbol{\alpha}}_t$ is unknown when we calculate $\boldsymbol{\mu}_t$ and \mathbf{H}_t in Equation (3.8). To address that issue, we use an iterative strategy to find $\hat{\boldsymbol{\alpha}}_t$: first we use the prior estimator $\hat{\boldsymbol{\alpha}}_t^-$ to calculate $\boldsymbol{\mu}_t$ and \mathbf{H}_t , then update $\hat{\boldsymbol{\alpha}}_t$ by the Kalman filter; and after that, we update $\boldsymbol{\mu}_t$ and \mathbf{H}_t using the newly estimated $\hat{\boldsymbol{\alpha}}_t$. Repeat this process until $\hat{\boldsymbol{\alpha}}_t$ converges. According to both [Durbin and Koopman \(1997\)](#) and our own experiments, this process routinely converges in fewer than five steps.

Algorithm 1 presents the detailed estimation and updating process. We put in Appendix B the basic steps and explanations of the Kalman filter for readers who are not familiar with it.

After we obtain the posterior estimation of the state $\hat{\boldsymbol{\alpha}}_t$, the corresponding NPSD $\hat{f}_t(x)$ can be represented as:

$$(3.9) \quad \hat{f}_t(x) = \frac{1}{C_t(\boldsymbol{\alpha}_t)} \exp \left[\sum_{j=1}^n \hat{\alpha}_{jt} B_j(x) \right],$$

where $\hat{\alpha}_{jt}$ is the j th element of $\hat{\boldsymbol{\alpha}}_t$. The normalizing constant $C_t(\boldsymbol{\alpha}_t)$ is computed by numerical integration after Algorithm 1 converges. Figure 4 highlights the main online updating steps for tracking the time-varying NPSD.

Remark: The variance stabilizing transformation provides a simple global Gaussian approximation of Poisson distribution. For $Y \sim \text{Poisson}(\lambda)$, $\sqrt{Y + \frac{1}{4}}$ is approximately $N(\sqrt{\lambda}, \frac{1}{4})$ ([Anscombe, 1948](#)). Based on this fact, [Brown et al. \(2010\)](#) studied a root-unroot nonparametric density estimation method. As pointed out by a referee, these work suggest a potentially simple solution to our online density updating problem, as follows—One first models the square-root of the density by a B-spline, i.e., $\sqrt{\lambda_t} = \mathbf{B}\boldsymbol{\alpha}_t$, applies the standard Kalman filter to the square-root transformed bin counts, and then unroots the estimator to obtain a proper density function.

Algorithm 1 Online updating method of the state space model.

1. Set $t = 1$ and initialize $\hat{\boldsymbol{\alpha}}_0$ and \mathbf{P}_0 .
2. Predict the prior estimator of the state as: $\hat{\boldsymbol{\alpha}}_t^- = \hat{\boldsymbol{\alpha}}_{t-1}$.
3. Predict the prior covariance matrix as: $\mathbf{P}_t^- = \mathbf{P}_{t-1} + \mathbf{Q}$.
4. Set $\hat{\boldsymbol{\alpha}}_t = \hat{\boldsymbol{\alpha}}_t^-$.
5. Calculate $\boldsymbol{\mu}_t$ and \mathbf{H}_t as:

$$\begin{aligned}\boldsymbol{\mu}_t &= \mathbf{Y}_t - \mathbf{B}\hat{\boldsymbol{\alpha}}_t - \exp(-\mathbf{B}\hat{\boldsymbol{\alpha}}_t)[\mathbf{Y}_t - \exp(\mathbf{B}\hat{\boldsymbol{\alpha}}_t)], \\ \mathbf{H}_t &= \text{diag}[\exp(-\mathbf{B}\hat{\boldsymbol{\alpha}}_t)].\end{aligned}$$

6. Compute the innovation and its covariance matrix:

$$\boldsymbol{\nu}_t = \mathbf{Y}_t - \mathbf{B}\hat{\boldsymbol{\alpha}}_t^- - \boldsymbol{\mu}_t; \quad \mathbf{F}_t = \mathbf{B}\mathbf{P}_t^-\mathbf{B}^T + \mathbf{H}_t.$$

7. Compute the Kalman gain as: $\mathbf{K}_t = \mathbf{P}_t^-\mathbf{B}^T\mathbf{F}_t^{-1}$.
 8. Update the posterior estimator with measurement \mathbf{Y}_t : $\hat{\boldsymbol{\alpha}}_t = \hat{\boldsymbol{\alpha}}_t^- + \mathbf{K}_t\boldsymbol{\nu}_t$.
 9. Repeat Step 5 to 8 until $\hat{\boldsymbol{\alpha}}_t$ converges.
 10. Update the posterior covariance matrix as: $\mathbf{P}_t = \mathbf{P}_t^-(\mathbf{I} - \mathbf{K}_t\mathbf{B})^T$.
 11. Set $t = t + 1$, repeat from Step 2 until the process ends.
-

We found this root-unroot approach does not work well, for two reasons: (1) The approximation of using the square root transformation has a large bias when λ is close to 0. Both [Anscombe \(1948\)](#) and [Brown et al. \(2010\)](#) noted that the bias is proportional to $\lambda^{-2/3}$. In our context of probability density estimation using binned data, since there are many bins with zero counts (corresponding to small λ 's), this approximation is too crude. (2) If we apply the standard Kalman filter to update $\sqrt{\lambda}_t = \mathbf{B}\boldsymbol{\alpha}_t$, we may end up with a negative value for $\sqrt{\lambda}_t$. Although its square still gives a positive density estimation, there is an upward bias. In fact, when λ_t is near zero, updating the state variables to reduce the value of λ_t may yield a negative updated value of $\mathbf{B}\boldsymbol{\alpha}_t$ and thus can lead to a larger absolute value of λ_t , contrary to the purpose of reducing λ_t , creating the bias.

As a comparison, the quality of our local Gaussian approximation does not depend on the value of λ . Moreover, it is used in an iterative manner, so that the approximation improves along the iterations. Our numerical results show that the local Gaussian approximation works rather well in the presence of empty bins, especially once the smoothness constraint is imposed.

3.3. Curve Smoothness for Distribution Estimation. While Algorithm 1 can provide an on-line estimation and updating of the time-varying NPSD, it does not impose any requirement on the smoothness of the estimated density function. Without a proper smoothness constraint, the resulting density estimation could be sensitive to choices like the number of intervals in the histogram \mathbf{Y}_t and the number of B-spline basis functions, and could become considerably inaccurate in the cases that some middle intervals in the input histogram turn out empty. So our goal here is to incorporate the curve smoothness constraint and make it work with the state space model.

We plan to impose the curve smoothness constraint for the B-spline density estimation by penalizing the squared norm of the second order difference of the spline coefficient vector $\boldsymbol{\alpha}_t$,

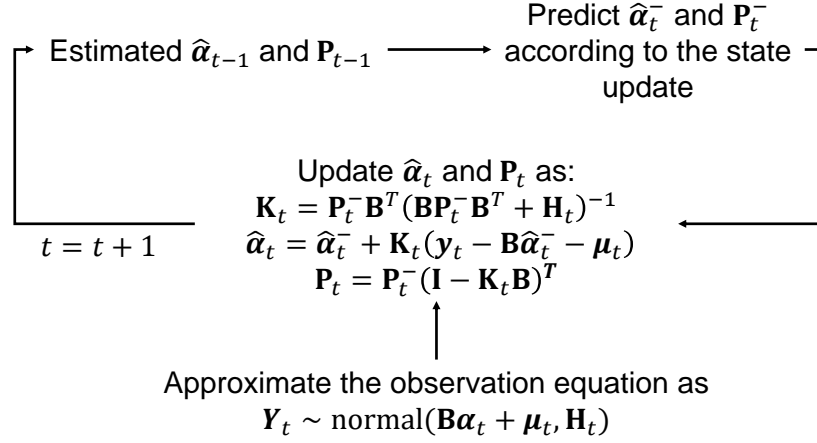


FIG 4. The illustration of main online updating steps.

which is denoted as an $n - 2$ dimension vector $\Delta^2 \boldsymbol{\alpha}_t$ and defined as

$$(3.10) \quad \Delta^2 \boldsymbol{\alpha}_t = \begin{bmatrix} -\alpha_{1t} + 2\alpha_{2t} - \alpha_{3t} \\ -\alpha_{2t} + 2\alpha_{3t} - \alpha_{4t} \\ \cdots \\ -\alpha_{(n-2)t} + 2\alpha_{(n-1)t} - \alpha_{nt} \end{bmatrix} = \begin{bmatrix} -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 2 & -1 \end{bmatrix} \boldsymbol{\alpha}_t.$$

In the smoothing spline literature, the commonly used penalty to enforce smoothness is the squared second derivative penalty $\int_x (f''(x))^2 dx$ (Wahba, 1990). Section 3 of Eilers and Marx (1996) pointed out that the squared second difference penalty $\|\Delta^2 \boldsymbol{\alpha}_t\|^2$ adopted here can be viewed as a convenient approximation with the B-spline representation. The precise relation of the two penalties is given in Equation (11) of Eilers and Marx (1996).

To put a constraint on $\|\Delta^2 \boldsymbol{\alpha}_t\|^2$, we propose to transform linearly the original state $\boldsymbol{\alpha}_t$ into another state $\boldsymbol{\gamma}_t$. The new state $\boldsymbol{\gamma}_t$ includes $\Delta^2 \boldsymbol{\alpha}_t$ but should ideally have the same dimension as $\boldsymbol{\alpha}_t$. We therefore make the last $n - 2$ elements of $\boldsymbol{\gamma}_t$ equal to $\Delta^2 \boldsymbol{\alpha}_t$ and then add something else as its first two elements for the purpose of making the new vector an n dimensional vector.

A straightforward choice for the first two elements is to let γ_{1t} be the summation of all the even numbered coordinates of $\boldsymbol{\alpha}_t$ and γ_{2t} be the summation of all the odd numbered coordinates of $\boldsymbol{\alpha}_t$. This choice ensures these two elements are orthogonal and have similar magnitudes. Because of the later property, we can set the variances of the corresponding innovations, \mathbf{w}_t , at the same value. Consequently, we obtain an invertible transformation from $\boldsymbol{\alpha}_t$ to $\boldsymbol{\gamma}_t$ as:

$$(3.11) \quad \gamma_{1t} = \sum_{j=1}^{[n/2]} \alpha_{(2j)t}, \quad \gamma_{2t} = \sum_{j=1}^{[n/2]} \alpha_{(2j-1)t}, \quad \gamma_{(3:n)t} = \Delta^2 \boldsymbol{\alpha}_t,$$

where $[n/2]$ is the largest integer smaller than or equal to $n/2$. We can also write this linear transform in a matrix format, such that $\boldsymbol{\alpha}_t = \mathbf{C} \boldsymbol{\gamma}_t$, where

$$(3.12) \quad \begin{aligned} \mathbf{C}_{1(2j)}^{-1} &= 1, \quad \mathbf{C}_{2(2j-1)}^{-1} = 1, \quad j = 1, \dots, [n/2]; \\ \mathbf{C}_{j(j-2)}^{-1} &= -1, \quad \mathbf{C}_{j(j-1)}^{-1} = 2, \quad \mathbf{C}_{jj}^{-1} = -1, \quad j = 2, \dots, n; \end{aligned}$$

and other elements of \mathbf{C}^{-1} are equal to 0.

By using the new state $\boldsymbol{\gamma}_t$, we express the state space model as:

$$(3.13) \quad \begin{aligned} Y_{it} &\sim \text{Poisson}\{(\exp[\mathbf{BC}\boldsymbol{\gamma}_t])_i\}, \\ \boldsymbol{\gamma}_t &= \boldsymbol{\gamma}_{t-1} + \mathbf{w}_t, \end{aligned}$$

where $\mathbf{w}_t \sim \text{normal}(\mathbf{0}, \mathbf{Q})$. Here we slightly abuse the notations—even though \mathbf{w}_t and \mathbf{Q} are used again, they are of different values from those in Equation (3.6).

This transformation in the state vector allows us to use the structure of \mathbf{Q} to add the smoothness constraint on the estimated density $\hat{f}_t(x)$. Aware that $[\gamma_{1t}, \gamma_{2t}]$ are the summations of the even and odd terms of $\boldsymbol{\alpha}_t$, respectively, and $[\gamma_{3t}, \dots, \gamma_{nt}]$ are the second differences of $\boldsymbol{\alpha}_t$, we assume that their innovations are independent to each other, making \mathbf{Q} a diagonal matrix, denoted as $\text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2)$, in which σ_1^2, σ_2^2 are the variances of w_{1t}, w_{2t} , and $\sigma_3^2, \dots, \sigma_n^2$ are the variances of w_{3t}, \dots, w_{nt} . For simplicity, we further assume that σ_1^2 and σ_2^2 have the same value, denoted as σ_α^2 , and all the remaining $\sigma_3^2, \dots, \sigma_n^2$ are equal, their value denoted as σ_ϵ^2 . According to the previous analysis, requiring $\sigma_\alpha^2 \gg \sigma_\epsilon^2$ for the new state vector is effectively forcing the second order difference of $\boldsymbol{\alpha}_t$ to be small and thus resulting the smoothness constraint imposed onto the estimated density curves. We can still use Algorithm 1 to update $\hat{\boldsymbol{\gamma}}_t$ after replacing \mathbf{B} and $\boldsymbol{\alpha}_t$ with \mathbf{BC} and $\boldsymbol{\gamma}_t$, respectively, so that the fast computation and online estimation/updating capability are retained.

4. Parameter Estimation and Selection. In order for our prospective analysis to work, we do need a short starting up period, which is to gather a limited amount of training video data to initialize the parameters in the model. We typically use the first few hundreds of frames for parameter estimation, equivalent to the first 15 to 20 seconds of the process.

In our state space model, there are two parameters σ_α^2 and σ_ϵ^2 that need to be estimated using the training data from the short starting up period, from $t = 1$ until time T . The two parameters determine the covariance matrix of \mathbf{w}_t : σ_α^2 represents the degree of variability of the underlying state $\boldsymbol{\gamma}_t$, whereas σ_ϵ^2 controls its second order, indicating the smoothness of the estimated density curve. It is not convenient to find the values of σ_α^2 and σ_ϵ^2 by maximized likelihood estimation (MLE) as calculation of the likelihood of such a mixed system needs complicated process like importance sampling (Durbin and Koopman, 1997) or simulation smoothing (de Jong and Shephard, 1995), let alone to optimize the likelihood to estimate its parameters. Here, we adopt a Bayesian approach to obtain the two parameters in the covariance matrix.

4.1. Bayesian Modeling. We regard σ_α^2 and σ_ϵ^2 as latent random variables and choose their prior distributions first. Then, we obtain their posterior distribution through a sampling method and use the corresponding posterior means as the estimate of the parameters.

Since σ_α^2 and σ_ϵ^2 define the covariance matrix of \mathbf{w}_t , which we assume follow a normal distribution, we choose the corresponding conjugate prior as an inverse-gamma distribution, making the posterior distribution in the same family. We can write the hierarchical structure of the Bayesian

model as:

$$(4.1) \quad \begin{aligned} Y_{it} &\sim \text{Poisson}\{(\exp[\mathbf{BC}\boldsymbol{\gamma}_t])_i\}, \\ \boldsymbol{\gamma}_t - \boldsymbol{\gamma}_{t-1} &= \mathbf{w}_t \sim \text{normal}(\mathbf{0}, \mathbf{Q}), \quad \mathbf{Q} = \text{diag}(\sigma_\alpha^2, \sigma_\alpha^2, \sigma_\epsilon^2, \dots, \sigma_\epsilon^2), \\ \sigma_\alpha^2 &\sim \text{inverse-gamma}(a_1, b_1), \quad \sigma_\epsilon^2 \sim \text{inverse-gamma}(a_2, b_2), \end{aligned}$$

where the initial state $\boldsymbol{\gamma}_0$ is set as $[-2, -2, \dots, -2]$ so that $f(x)$ will evolve from a zero function.

To determine the hyper-parameters in the inverse-gamma distributions of σ_α^2 and σ_ϵ^2 , we choose the non-informative prior (Spiegelhalter et al., 1996) as $a_1 = 1.0$ and $b_1 = 1.0$ for σ_α^2 . To make sure $\sigma_\alpha^2 \gg \sigma_\epsilon^2$, we choose the same shape parameter, i.e., $a_2 = 1.0$, but a much smaller scale parameter b_2 for σ_ϵ^2 (the mean of the inverse-gamma distribution is proportional to the scale parameter). We recommend choosing b_2 such that $b_1/b_2 = 100$. The robustness check of this choice is discussed in Section 4.3.

Compared to the original state space model, the hierarchical model adds another layer associated with the prior distributions of σ_α^2 and σ_ϵ^2 . Once observing $\mathbf{Y}_1, \dots, \mathbf{Y}_T$ in the starting up period, we employ a Markov chain Monte Carlo (MCMC) sampling method to update the posterior distributions of σ_α^2 and σ_ϵ^2 , and then use the posterior means as the estimate of the two parameters.

4.2. MCMC Sampling. Denote the values in the k th iteration of MCMC by $\boldsymbol{\gamma}_1^{(k)}, \dots, \boldsymbol{\gamma}_T^{(k)}$, $(\sigma_\alpha^2)^{(k)}$ and $(\sigma_\epsilon^2)^{(k)}$. After the initialization, we sample $(\sigma_\alpha^2)^{(k)}$ and $(\sigma_\epsilon^2)^{(k)}$ through the Gibbs sampling, given $\boldsymbol{\gamma}_1^{(k-1)}, \dots, \boldsymbol{\gamma}_T^{(k-1)}$. Since we adopt the conjugate priors, the posterior distributions are still inverse-gamma as:

$$(4.2) \quad (\sigma_\alpha^2)^{(k)} \sim \text{inverse-gamma}(a_1^{\text{post}}, b_1^{\text{post}}), \quad (\sigma_\epsilon^2)^{(k)} \sim \text{inverse-gamma}(a_2^{\text{post}}, b_2^{\text{post}}),$$

where $a_1^{\text{post}}, b_1^{\text{post}}, a_2^{\text{post}}$ and b_2^{post} are determined by a_1, b_1, a_2, b_2 , and the sampled $\boldsymbol{\gamma}_t^{(k-1)}$. The derivation of the posterior distribution of $(\sigma_\alpha^2)^{(k)}$ and $(\sigma_\epsilon^2)^{(k)}$ is included in Appendix C.

Then, we sample $\boldsymbol{\gamma}_1^{(k)}, \dots, \boldsymbol{\gamma}_T^{(k)}$, given $\mathbf{Q}^{(k)} = \text{diag}[(\sigma_\alpha^2)^{(k)}, (\sigma_\alpha^2)^{(k)}, (\sigma_\epsilon^2)^{(k)}, \dots, (\sigma_\epsilon^2)^{(k)}]$ and the observations, $\mathbf{Y}_1, \dots, \mathbf{Y}_T$. Unfortunately, the posterior distributions of $\boldsymbol{\gamma}_t^{(k)}$ are not of a standard type. We therefore implement a Metropolis-Hastings algorithm to sample $\boldsymbol{\gamma}_t^{(k)}$ from $t = 1$ to T . For each individual t , we first draw $\boldsymbol{\gamma}_t^{(k)}$ from the following proposal distribution:

$$(4.3) \quad \boldsymbol{\gamma}_t^{(k)} \sim \text{normal}(\boldsymbol{\gamma}_t^{(k-1)}, \mathbf{R}),$$

where $\mathbf{R} = \text{diag}(\sigma_1^2, \sigma_1^2, \sigma_2^2, \dots, \sigma_2^2)$ shares a similar structure as \mathbf{Q} . The acceptance ratio of a newly sampled $\boldsymbol{\gamma}_t^{(k)}$, r , is defined in a standard way, as the ratio of the conditional pdf given the current $\boldsymbol{\gamma}_t^{(k)}$ to that given the previous $\boldsymbol{\gamma}_t^{(k-1)}$. After getting r , we compare it with a uniform random variable, u , in $[0, 1]$, to determine whether to accept the new $\boldsymbol{\gamma}_t^{(k)}$ or not.

After repeating the above sampling iterations K times, the posterior means can be obtained by:

$$(4.4) \quad \hat{\sigma}_\alpha^2 = \frac{1}{K - K_B} \sum_{k=K_B+1}^K (\sigma_\alpha^2)^{(k)}, \quad \hat{\sigma}_\epsilon^2 = \frac{1}{K - K_B} \sum_{k=K_B+1}^K (\sigma_\epsilon^2)^{(k)},$$

where K_B is the amount of the burn-in steps. We list the detailed steps in Algorithm 2.

Algorithm 2 parameter estimation through Bayesian sampling

1. Initialize $\gamma_1^{(0)}, \dots, \gamma_T^{(0)}, (\sigma_\alpha^2)^{(0)}$ and $(\sigma_\epsilon^2)^{(0)}$.
2. Set $k = 1$, then sample $(\sigma_\alpha^2)^{(k)}$ and $(\sigma_\epsilon^2)^{(k)}$ as: $(\sigma_\alpha^2)^{(k)} \sim \text{inverse-gamma}(a_1^{\text{post}}, b_1^{\text{post}})$ and $(\sigma_\epsilon^2)^{(k)} \sim \text{inverse-gamma}(a_2^{\text{post}}, b_2^{\text{post}})$, where

$$\begin{aligned} a_1^{\text{post}} &= a_1 + (T - 1), b_1^{\text{post}} = b_1 + \frac{1}{2} \sum_{j=1}^2 \sum_{t=2}^T [\gamma_{jt}^{(k-1)} - \gamma_{j(t-1)}^{(k-1)}]^2, \\ a_2^{\text{post}} &= a_2 + \frac{n-2}{2}(T - 1), b_2^{\text{post}} = b_2 + \frac{1}{2} \sum_{j=3}^n \sum_{t=2}^T [\gamma_{jt}^{(k-1)} - \gamma_{j(t-1)}^{(k-1)}]^2. \end{aligned}$$

3. Let $\mathbf{Q}^{(k)} = \text{diag}[(\sigma_\alpha^2)^{(k)}, (\sigma_\alpha^2)^{(k)}, (\sigma_\epsilon^2)^{(k)}, \dots, (\sigma_\epsilon^2)^{(k)}]$.
4. Set $t = 1$, sample $\gamma_t^{(k)}$ from a proposal distribution: $\gamma_t^{(k)} \sim \text{normal}(\gamma_t^{(k-1)}, \mathbf{R})$.
5. Calculate the acceptance rate r as:

$$r = \frac{\prod_{i=1}^m p_{\text{Poi}}(Y_{it} | [\mathbf{BC}\gamma_t^{(k)}]_i) p_{\text{nor}}(\gamma_t^{(k)} | \gamma_{t-1}^{(k)}, \mathbf{Q}^{(k)}) p_{\text{nor}}(\gamma_t^{(k)} | \gamma_{t+1}^{(k-1)}, \mathbf{Q}^{(k)})}{\prod_{i=1}^m p_{\text{Poi}}(Y_{it} | [\mathbf{BC}\gamma_t^{(k-1)}]_i) p_{\text{nor}}(\gamma_t^{(k-1)} | \gamma_{t-1}^{(k-1)}, \mathbf{Q}^{(k)}) p_{\text{nor}}(\gamma_t^{(k-1)} | \gamma_{t+1}^{(k-1)}, \mathbf{Q}^{(k)})},$$

where $p_{\text{Poi}}(\cdot)$ is the pdf of a Poisson distribution and $p_{\text{nor}}(\cdot | \cdot, \cdot)$ is the pdf of a multivariate normal distribution.

6. Generate a uniform random number, u , in $[0, 1]$. If $r > u$, accept $\gamma_t^{(k)}$; otherwise set $\gamma_t^{(k)} = \gamma_t^{(k-1)}$.
7. Set $t = t + 1$, and repeat Step 4 to 6 until $t = T$.
8. Set $k = k + 1$, and repeat Step 2 to 7 until $k = K$.
9. Estimate σ_α^2 and σ_ϵ^2 as the posterior means:

$$\hat{\sigma}_\alpha^2 = \frac{1}{K - K_B} \sum_{k=K_B+1}^K (\sigma_\alpha^2)^{(k)}, \quad \hat{\sigma}_\epsilon^2 = \frac{1}{K - K_B} \sum_{k=K_B+1}^K (\sigma_\epsilon^2)^{(k)}.$$

TABLE 1

The parameters, $\hat{\sigma}_\alpha^2$ and $\hat{\sigma}_\epsilon^2$, and their 90% credible intervals, estimated using Video 1 data and under different b_2 values. In the following, $a_1 = a_2 = b_1 = 1.0$.

b_2	b_1/b_2	$\hat{\sigma}_\alpha^2$	$\hat{\sigma}_\epsilon^2$	$\hat{\sigma}_\alpha^2/\hat{\sigma}_\epsilon^2$
0.1	10	5.94 (4.44, 7.70) $\times 10^{-2}$	4.03 (3.52, 4.64) $\times 10^{-3}$	14.73
0.05	20	6.47 (4.89, 8.41) $\times 10^{-2}$	3.92 (3.44, 4.54) $\times 10^{-3}$	16.49
0.01	100	6.39 (4.46, 8.77) $\times 10^{-2}$	3.82 (3.27, 4.35) $\times 10^{-3}$	16.72
0.005	200	6.39 (4.80, 8.16) $\times 10^{-2}$	3.66 (3.25, 4.18) $\times 10^{-3}$	17.46

4.3. *Select the hyper-parameters.* In this subsection, we discuss the choices of the hyper-parameters in the Bayesian model (4.1) and the MCMC algorithm: a_1 , b_1 , a_2 and b_2 in the prior distribution, the initial values of the MCMC sampling, $\gamma_t^{(0)}$, $(\sigma_\alpha^2)^{(0)}$ and $(\sigma_\epsilon^2)^{(0)}$, and σ_1^2 and σ_2^2 in the covariance matrix \mathbf{R} of the proposal distribution. The parameters in the MCMC sampling matter less, as a long burn-in stage (namely a large enough K_B) makes the MCMC robust to initialization. As long as the MCMC has a good mixing, different proposal distributions give similar estimation outcomes. We set those parameters in the following way: $(\sigma_\alpha^2)^{(0)} = 4 \times 10^{-2}$, $(\sigma_\epsilon^2)^{(0)} = 2 \times 10^{-3}$, run the extended Kalman filter in Algorithm 1 to obtain $\gamma_t^{(0)}$, and let $\sigma_1^2 = 2 \times 10^{-2}$ and $\sigma_2^2 = 1 \times 10^{-3}$.

While a_1 , b_1 and a_2 are specified in Section 4.1, we run the MCMC to find a suitable value for b_2 . We found that as long as b_1/b_2 is large enough, say, more than an order of magnitude, the estimation outcome appears robust. Table 1 presents the posterior means of the two parameters

estimated from Video 1, with a total of $K = 1 \times 10^5$ iterations and $K_B = 4 \times 10^4$ burn-in steps. Despite significantly different b_2 's are used, the estimated results for other parameters stay similar. In practice, we recommend using $b_2 = 0.01$ as the default setting. We also check the convergence of the MCMC by plotting the chains of $(\sigma_\alpha^2)^{(k)}$ and $(\sigma_\epsilon^2)^{(k)}$ with multiple initial values and find that all the chains mix well after the burn-in stage.

5. Application to TEM Videos. We test our state space model and its online updating on the three clips of *in situ* TEM video described in Section 2. The number of the B-spline basis functions is fixed at 20 in all three cases. Because of incorporation of the smoothness constraint in our state space model, our final estimation of the NPSD is not sensitive to the choices of this parameter. To save space, we discuss the full results on Video 1 clip. For the other two clips, we present limited analysis results to confirm the generality of the modeling and analysis.

5.1. *Analysis of the three videos.* Our first step is to find σ_α^2 and σ_ϵ^2 for each clip of videos. In Video 1, there are 1,149 frames in total, with 15 frame per second (fps) frame rate. We choose the first 300 frames as the training set, corresponding to the first 20 seconds of the process. Using the Bayesian estimation method in Section 4 with the default parameter setting, our estimate of the two system parameters is $\hat{\sigma}_\alpha^2 = 6.39 \times 10^{-2}$ and $\hat{\sigma}_\epsilon^2 = 3.82 \times 10^{-3}$.

Next we apply our updating method to the whole video. In our test, the TEM videos have already been fully recorded. We mimic a prospective analysis, starting at the end of the initialization period. For the remaining 849 frames in Video 1, the total processing time of using our algorithm is 1.23 second, or 1.5×10^{-4} seconds per frame, much faster than the frame rate of the video (which is 15 frames per second or 0.067 seconds per frame). Combined with the image processing time (0.04 seconds per frame), the overall model processing is still fast enough for online monitoring. Figure 5 illustrates the updating process running from 25.67 second through 28.33 second. The upper row shows the input histograms, whereas the lower row shows the updated NPSDs. To demonstrate the difference of the estimated distributions, the time difference between two consecutive images in that plot is chosen to be 10 frames.

We also show in Figure 6 the estimated NPSDs in different growth stages at 15s, 30s, 45s and 60s, respectively. Figure 6(a) presents the NPSD at the beginning of the growth stage when the nanocrystals are initializing in the chemical solution. The variance of the particle sizes is large and the support of the distribution is broad. Figure 6(b) presents a NPSD at the orientated attachment (Aldous, 1999) growth stage, at which time the smaller particles collide with each other and are merged into larger ones. The variance of the particle sizes is smaller than that of the first stage. There is a noticeable bimodal pattern in the NPSD, in which the two peaks correspond to the sizes of the smaller particles and the merged (larger) particles, respectively. The final two plots in Figure 6(c) and (d) are in the final growth stage, known as the Ostwald ripening (Lifshitz and Slyozov, 1961) stage. In that stage, the larger particles grow at the expense of dissolving smaller particles. The size distribution tends to get concentrated and become unimodal. The variance continues to decrease. Material scientists expect to get nanocrystals having more uniform sizes at the end of the growth process. Our state space model's online tracking results are consistent with the manual analysis results presented in the original report (Zheng

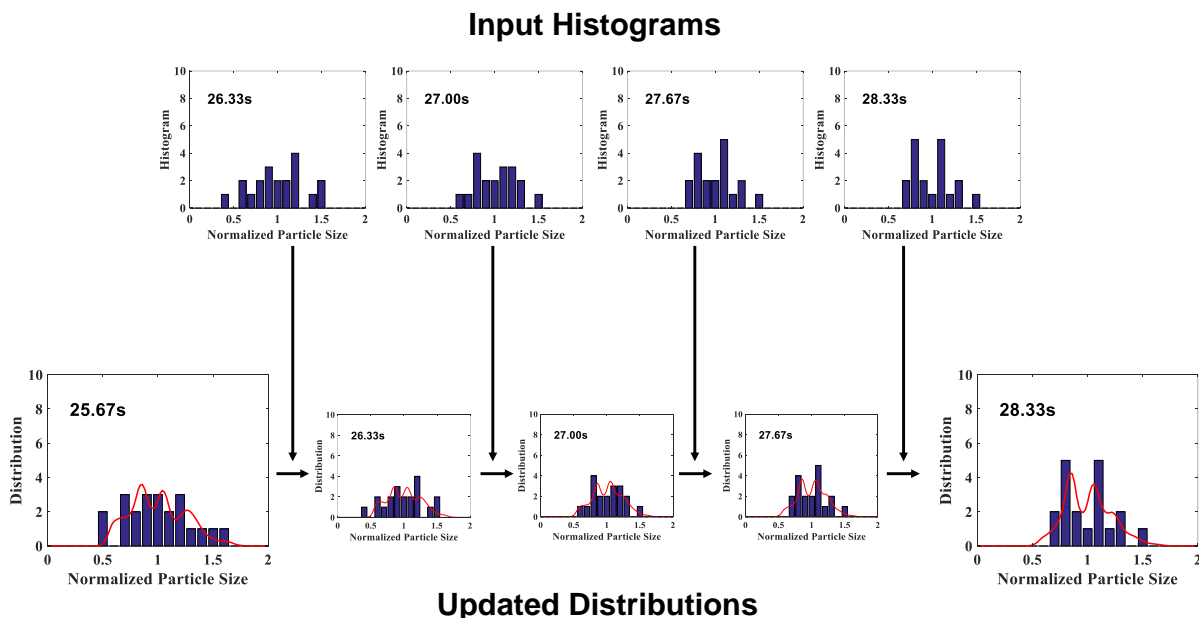


FIG 5. *Illustration of the updating outcomes of the state space model.*

et al., 2009).

The last part of analysis performed on Video 1 is to show the innovation sequence of this nanocrystal growth process. Loosely speaking, the innovation sequence is the difference between what is newly observed at time t and what is anticipated, based on the state space model and historical observations. In the literature, the innovation sequence is commonly used to indicate a process change: if the underlying process is stable, then the innovation is supposedly to be random noise, whereas if the underlying process is going through a change, then the innovation sequence shows departure from random noise. The innovation at time t , denoted by ν_t and its covariance matrix \mathbf{F}_t , is computed in Step 6 of Algorithm 1. To monitor the multivariate vector ν_t , we calculate the Mahalanobis squared distance (Mahalanobis, 1936) between ν_t and $\mathbf{0}$ at each

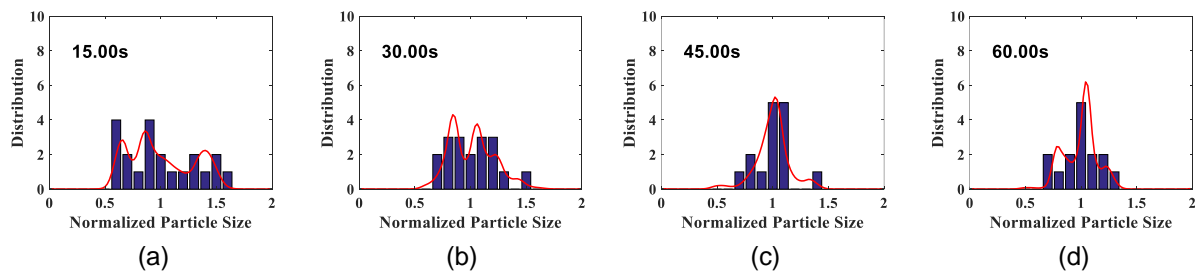


FIG 6. *The estimated NPSD of Video 1 at different growth stages.*

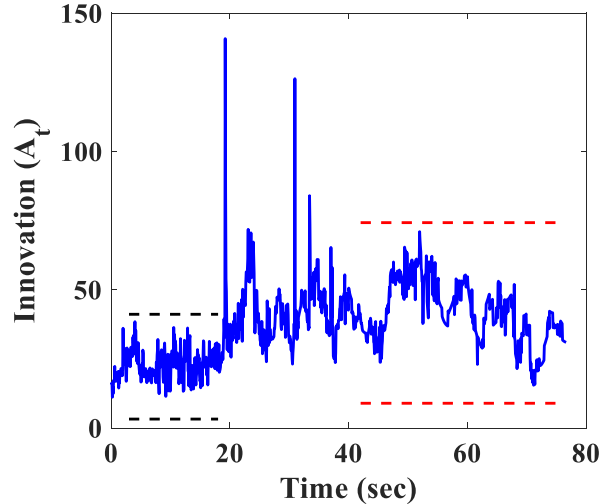


FIG 7. *Statistic, A_t , obtained from the innovation sequence of the Kalman filter with the 3-sigma control limits for the two growth stages.*

t , such that

$$(5.1) \quad A_t = \boldsymbol{\nu}_t^T \mathbf{F}_t^- \boldsymbol{\nu}_t.$$

The sequence $\{A_1, A_2, \dots\}$ for Video 1 is plotted in Figure 7. We observe that there is a noticeable process change between the 20 second and 40 second time marks with an increased variance. Before and after that period, the innovation sequence appears to have smaller magnitudes. This observation is consistent with the physical understanding discovered by Zheng et al. (2009), i.e., the beginning stage of the growth is driven by the mechanism of orientated attachment, the latter stage is driven by the mechanism of Ostwald ripening, and there is a transition period in between. The timing of the transition period, discovered in the retrospective analysis (Qian et al., 2017), is between 25.8 second and 39.9 second. We also plot the 3-sigma control limits for the two stages in Figure 7, where the peaks in the transition period is far greater than the upper control limits. The result in Figure 7 shows that by tracking the innovation sequence of the state space model, it offers the opportunity to detect possible mechanism changes in the process.

Next, we test our algorithm on Video 2, which was published in the same paper as Video 1 (Zheng et al., 2009) and captures a similar nanocrystal self-assembly growth process. There are total 637 frames in Video 2 with 15 fps frame rate. We still choose the first 300 frames to estimate the parameters. The Bayesian method produces the estimate of σ_α^2 as 7.24×10^{-2} and that of σ_ϵ^2 as 4.19×10^{-3} . Using these parameters, we estimate the NPSDs and show some results in Figure 8. The total updating time is 0.098 seconds, or 1.54×10^{-4} seconds per frame; this computational performance is consistent with that for processing Video 1 (and the image processing also takes 0.04 seconds per frame). Video 2 is a shorter clip and contains fewer particles. By observing the density plots in Figure 8, we are satisfied with the density curves estimated by our state space model.

Lastly, we test our algorithm on Video 3. It was published in Woehl et al. (2013) and captures a

different growth process than that in Videos 1 and 2. This process is of silver nanocrystal growth. There are only 112 frames in this video clip with 1 fps frame rate, so we pick the first 50 frames as the training set to estimate the parameters. For the process in Video 3, the parameters are accordingly estimated as $\sigma_\alpha^2 = 1.75 \times 10^{-1}$ and $\sigma_\epsilon^2 = 7.56 \times 10^{-3}$. Applying our updating method to Video 3, the total run time is 0.02 seconds, or 1.79×10^{-4} seconds per frame. The image processing time for Video 3 is 0.2 seconds per frame, so that the combined computation is again faster than the frame rate. Figure 9 presents the estimated NPSD of Video 3. In this process, the NPSD is always uni-modal and its variance gets larger in the process.

5.2. Comparison with alternative methods. In this subsection, we demonstrate the merits of the proposed method, especially the benefit of having both the curve smoothness and temporal continuity constraints. We demonstrate all comparison results using Video 1 but the same insight holds true for other videos. We do not compare our method with a retrospective method because a retrospective (offline) method sees all data and has the luxury of time, whereas a prospective (online) method only sees a subset of the data, unless it reaches the very end of the video, and must be time conscious.

The first comparison is to conduct an out-of-sample quantitative test, comparing the proposed state space method with three types of alternative: the first type is a pure histogram-based treatment (no smoothness constraint at all), the second type is to impose the curve smoothness within a frame but estimate the NPSD one frame a time without considering and imposing temporal continuity, and the third type is a state space model without the curve smoothness (i.e., with temporal continuity across frames but no curve smoothness within a frame). In the second type of alternative, we include three popular methods: the smoothed histogram (Simonoff, 1983), the kernel estimation (Sheather and Jones, 1991), and the penalized B-splines (Eilers and Marx, 1996). For the state space model without the curve smoothness, we use α_t instead of γ_t as the state, and the covariance matrix \mathbf{Q} is set as $\text{diag}(\sigma_\alpha^2, \sigma_\alpha^2, \dots, \sigma_\alpha^2)$. The single parameter σ_α^2 can be estimated by a simplified Bayesian model, assuming $\sigma_\alpha^2 \sim \text{inverse-gamma}(1, 1)$. The first 300 frames are still used for the training purpose. The Bayesian estimate of σ_α is 5.9×10^{-2} , which is rather close to that estimated in the previous subsection.

The out-of-sample test calculates the log-likelihood of the estimated probability density functions based on a number of observed nanocrystals. We randomly pick 90% the observed nanocrys-

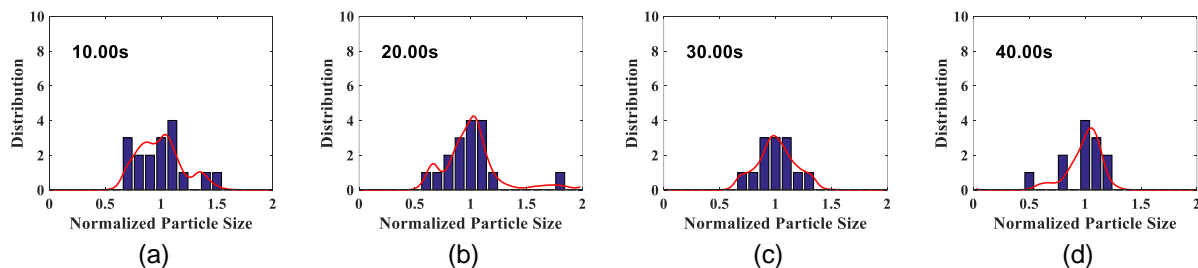


FIG 8. The estimated NPSDs of Video 2.

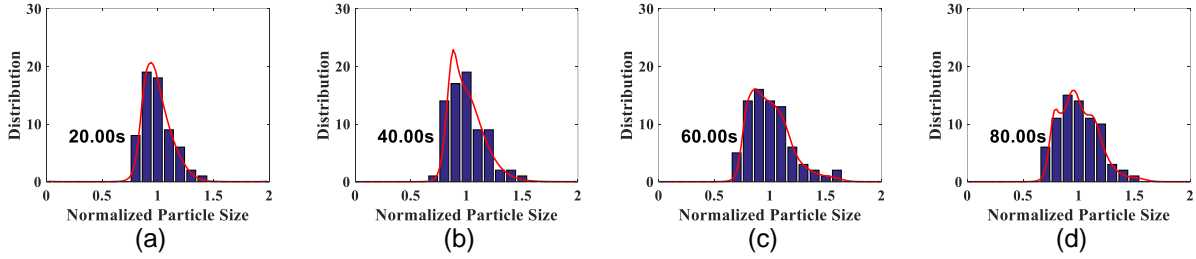


FIG 9. The estimated NPSDs of Video 3.

tals in each and every image frame and use them to establish our model and estimate the NPSD. Then, we use the remaining 10% observed nanocrystals in each and every frame to calculate the log-likelihood. For a given testing nanocrystal observation having a normalized particle size x^ℓ at frame t , its log-likelihood is:

$$(5.2) \quad \log f_t(x^\ell) = \sum_{j=1}^n B_j(x^\ell)[\mathbf{C}\gamma_t]_j - \log C_t(\mathbf{C}\gamma_t).$$

We proceed to calculate the summation of the log-likelihoods for all of the 10% out-of-sample testing nanocrystals at all time frames and then use this summation as the accuracy metric for the distribution estimation. We repeat the out-of-sample test 500 times for each of the six methods. The mean of the log-likelihoods results are summarized in Table 2.

In the out-of-sample test, the shortcoming of using the histogram directly is highlighted—almost all the log-likelihoods obtained are negative infinity. When certain samples fall into an empty interval of the histogram (meaning that this interval does not have any training observations), the direct histogram method sets the likelihood of this testing sample as 0, causing the log-likelihood to be negative infinity.

The distribution estimation methods with the curve smoothness can overcome this negative infinity problem. However, these methods estimate the distribution from each frame independently, lacking the ability to borrow information across time frames. When the number of observations at individual frames is not large enough, they fail to produce a quality estimate, as evident by the poor results in the out-of-sample test.

TABLE 2

Comparison results of the out-of-sample test among six approaches: using the observed histograms directly, three estimation methods considering the curve smoothness only, the state space method without the curve smoothness, and the proposed method; all tested on Video 1.

Methods		Mean of log-likelihoods
Observed histograms (no constraint)		$-\infty$
Curve smoothness only	Smoothed histograms	-41.6
	Kernel estimation	-24.4
	Penalized B-splines	-46.7
State space model (with temporal continuity)	Without curve smoothness	129.8
	With curve smoothness	196.1

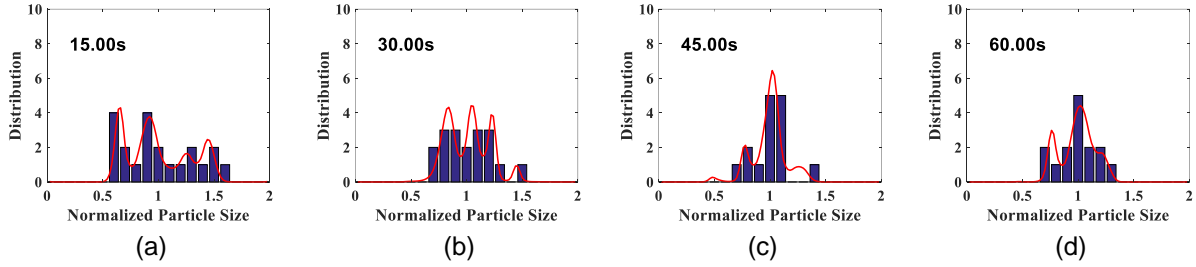


FIG 10. The estimated NPSDs of Video 1 by the state space model without the curve smoothness.

By using the state space transition equations, the two state space methods incorporate the temporal continuity, allowing the estimators to borrow information from other image frames and leading to much better performances than the other alternatives.

Between the state space models with and without the curve smoothness, the one with the curve smoothness produces a much higher log-likelihood measure. We conduct a statistical testing and see whether the log-likelihood difference between the two approaches is significant. A one-way ANOVA, in which the null hypothesis is that the two log-likelihoods have the same mean, yields a p-value of 6×10^{-162} , which confirm that the difference is indeed significant.

Given the benefit of using the state space framework demonstrated above, we hence set the focus of the next two comparisons to be between the two state space models, with and without the curve smoothness.

The second comparison is to inspect the resulting NPSD obtained by the two state space models. In Figure 10, we show the NPSDs at 15s, 30s, 45s and 60s, respectively, estimated by the state space model without the curve smoothness. Comparing the results in Figure 6 obtained at the same time marks by the state space model with the curve smoothness, the estimated distributions in Figure 10 are worse, as the state space model without the curve smoothness apparently overfits the histogram, and consequently, it is sensitive to small changes in the number of particles in a bin. To see this point, consider the following observations. In Figure 10(b), while the orientated attachment growth mechanism suggests a bimodal distribution, the estimated distribution gives us three peaks. Between Figure 10(c) and (d), the variance is supposed to decrease, as this is in the Ostwald ripening growth stage, but the estimated distribution shows an increasing variance. When displaying the online distribution estimation frame by frame, it is obvious to us that the state space model without the curve smoothness produces a time-varying NPSD that is far more volatile and often reacts too dramatically to noises and disturbances.

The third comparison is to show the robustness of the proposed method to possible changes in the number of intervals in the input histograms. In the previous studies, we set the length of interval as 0.1 which gives 20 intervals in a histogram. In this comparison experiment, we test the cases by setting the length of interval to 0.2, 0.15, 0.08 and 0.05, respectively, and then estimate the corresponding NPSD, using the state space model with and without the curve smoothness. We compare the resulting NPSDs with that obtained under the default setting, i.e., the length of interval 0.1 or 20 intervals in the histogram. The difference between the two NPSDs is measured

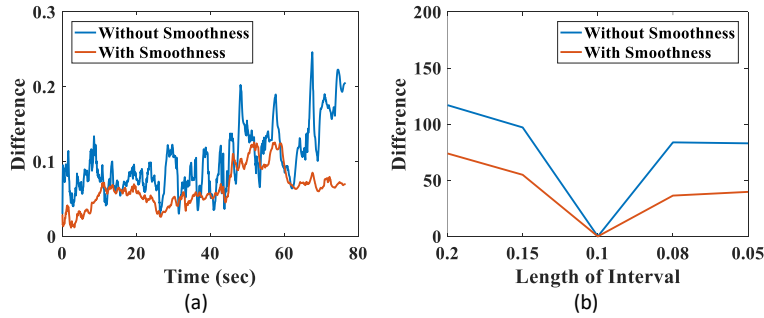


FIG 11. L_2 -norm difference between two NPSDs: (a) L_2 -norm differences at each time frame between the NPSDs estimated using the histograms with 10 intervals and 20 intervals; (b) the summation over all time frames of the L_2 -norm differences between the NPSDs estimated using histograms of various lengths of intervals and the default setting.

by a L_2 norm of the two density function curves.

In Figure 11(a), we plot the L_2 -norm differences at each time frame between the NPSDs estimated, respectively, using the binned data with 10 intervals (the length of an interval 0.2) and 20 intervals (the length of an interval 0.1). It is apparent that inclusion of the curve smoothness leads to an estimation less sensitive to the number of intervals, especially in the later stage of the process. In Figure 11(b), we present the summation over all frames of the L_2 -norm differences between the NPSDs estimated, respectively, using binned data of a various number of intervals and the default setting (i.e., 20 intervals or interval length 0.1). In the broad range of choices, the curve smoothness penalty generally decreases the L_2 -norm differences due to the change of intervals by half. Those results show that the proposed method can alleviate the overfitting when using a small interval for binned data.

6. Summary. In this paper, we propose an online method for monitoring the evolution of certain population characteristics observed in dynamic imaging (i.e., videos). Our model injects a flexible and robust modeling ability into a fast and closed-form updating algorithm. We demonstrate its application in monitoring the particle size distribution as a nanocrystal growth process is being observed by an *in situ* TEM.

The contributions of this work can be summarized as follows:

- The recursive, nonparametric method that models a time-varying probability density function and its specific tailoring to an evolving nanocrystal growth process;
- A closed-form updating algorithm in the form of an extended Kalman filter for tracking the nanocrystal growth in real time;
- The incorporation of both the curve smoothness and temporal continuity in the state space model for estimating the time-varying NPSD.

Even though our method is demonstrated in the context of estimating the normalized particle size distribution, we believe that the resulting method has some degree of generality and could be applicable to other online distribution estimation problems. For other applications, one needs

to replace the normalized particle size with a population characteristic of specific interest to that application. One importance assumption that may face challenges is the random walk assumption on the disturbance vector. Nonetheless, the random walk assumption appears a broadly accepted choice that can be a good starting point in a modeling effort, unless there exist contradicting evidences associated with a specific application to override its use.

Acknowledgements. The authors would like to acknowledge the generous support from their sponsors. This work is partially supported by AFOSR DDDAS program grants FA9550-17-1-0044 and FA9550-18-1-0144.

SUPPLEMENTARY MATERIAL

Supplement A: ([http://; .pdf](http://)). A pdf document including Appendix A, B and C. This document provides the derivations of the Gaussian approximation of the Poisson distribution, the detailed steps of Kalman filter, and the derivation of the posterior distributions of the system parameters for the proposed model.

Supplement B: ([http://; .zip](http://)). A zip file including the description of the testing videos, and the MATLAB codes to reproduce the results in the paper. A “Data and Codes.docx” file provides the detailed guidance to use the data and codes. The three videos have been published and are free to download, and all the codes have been tested under MATLAB 2016b.

References.

- Aldous, D. (1999). Deterministic and stochastic models for coalescence (aggregation and coagulation): a review of the mean-field theory for probabilists. *Bernoulli* 5(1), 3–48.
- Anscombe, F. J. (1948). The transformation of Poisson, binomial and negative-binomial data. *Biometrika* 35(3/4), 246–254.
- Bishop, Y., S. Fienberg, and P. Holland (1975). *Discrete Multivariate Analysis: Theory and Practice*. Springer-Verlag New York.
- Boal, A. K., F. Ilhan, J. E. DeRouchey, T. Thurn-Albrecht, T. P. Russell, and V. M. Rotello (2000). Self-assembly of nanoparticles into structured spherical and network aggregates. *Nature* 404(6779), 746–748.
- Brown, L., T. Cai, R. Zhang, L. Zhao, and H. Zhou (2010). The root-unroot algorithm for density estimation as implemented via wavelet block thresholding. *Probability Theory and Related Fields* 146(3-4), 401–433.
- de Jong, P. and N. Shephard (1995). The simulation smoother for time series models. *Biometrika* 82(2), 339–350.
- Doucet, A., N. J. Gordon, and V. Krishnamurthy (2001). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing* 49(3), 613–624.
- Durbin, J. and S. J. Koopman (1997). Monte Carlo maximum likelihood estimation for non-Gaussian state space models. *Biometrika* 84(3), 669–684.
- Eilers, P. H. and B. D. Marx (1996). Flexible smoothing with B-splines and penalties. *Statistical Science* 11(2), 89–102.
- Grzelczak, M., J. Vermant, E. M. Furst, and L. M. Liz-Marzán (2010). Directed self-assembly of nanoparticles. *ACS Nano* 4(7), 3591–3605.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering* 82(1), 35–45.
- Li, M., H. Schnablegger, and S. Mann (1999). Coupled synthesis and self-assembly of nanoparticles to give structures with controlled organization. *Nature* 402(6760), 393–395.
- Lifshitz, I. and V. Slyozov (1961). The kinetics of precipitation from supersaturated solid solutions. *Journal of Physics and Chemistry of Solids* 19(1), 35–50.

- Ljung, L. (1979). Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems. *IEEE Transactions on Automatic Control* 24(1), 36–50.
- Lo, A. Y. (1984). On a class of Bayesian nonparametric estimates: I. density estimates. *The Annals of Statistics* 12(1), 351–357.
- Ma, J., K. M. Kockelman, and P. Damien (2008). A multivariate Poisson-lognormal regression model for prediction of crash counts by severity using Bayesian methods. *Accident Analysis and Prevention* 40(3), 964–975.
- Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Sciences of India* 2(1), 49–55.
- Mena, R. H. and M. Ruggiero (2016). Dynamic density estimation with diffusive Dirichlet mixtures. *Bernoulli* 22(2), 901–926.
- Munesawang, P. and C. Sirisathitkul (2015). Size measurement of nanoparticle assembly using multilevel segmented TEM images. *Journal of Nanomaterials* 16(1), 58–63.
- Park, C. (2014). Estimating multiple pathways of object growth using nonlongitudinal image data. *Technometrics* 56(2), 186–199.
- Park, C., J. Huang, D. Huitink, S. Kundu, B. Mallick, H. Liang, and Y. Ding (2012). A multi-stage, semi-automated procedure for analyzing the morphology of nanoparticles. *IIE Transactions Special Issue on Nanomanufacturing* 44(7), 507–522.
- Park, C., J. Huang, J. Ji, and Y. Ding (2013). Segmentation, inference and classification of partially overlapping nanoparticles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(3), 669–681.
- Park, C., T. J. Woehl, J. E. Evans, and N. D. Browning (2015). Minimum cost multi-way data association for optimizing multitarget tracking of interacting objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(3), 611–624.
- Qian, Y., J. Z. Huang, and Y. Ding (2017). Identifying multi-stage nanocrystal growth using *in situ* TEM video data. *IIEE Transactions* 49(5), 532–543.
- Qian, Y., J. Z. Huang, X. Li, and Y. Ding (2016). Robust nanoparticles detection from noisy background by fusing complementary image information. *IEEE Transactions on Image Processing* 25(12), 5713–5726.
- Rodriguez, A. and E. Ter Horst (2008). Bayesian dynamic density estimation. *Bayesian Analysis* 3(2), 339–365.
- Sheather, S. J. and M. C. Jones (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society, Series B (Methodological)* 53(3), 683–690.
- Simonoff, J. S. (1983). A penalty function approach to smoothing large sparse contingency tables. *The Annals of Statistics* 11(1), 208–218.
- Spiegelhalter, D., A. Thomas, N. Best, and W. Gilks (1996). BUGS 0.5: Bayesian inference using Gibbs sampling manual (version ii). *MRC Biostatistics Unit, Institute of Public Health, Cambridge, UK*, 1–59.
- Wahba, G. (1990). *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics.
- Woehl, T. J., C. Park, J. E. Evans, I. Arslan, W. D. Ristenpart, and N. D. Browning (2013). Direct observation of aggregative nanoparticle growth: kinetic modeling of the size distribution and growth rate. *Nano Letters* 14(1), 373–378.
- Zhang, C., N. Chen, and Z. Li (2017). State space modeling of autocorrelated multivariate Poisson counts. *IIEE Transactions* 49(5), 518–531.
- Zheng, H., R. K. Smith, Y. W. Jun, C. Kisielowski, U. Dahmen, and A. P. Alivisatos (2009). Observation of single colloidal platinum nanocrystal growth trajectories. *Science* 324(5932), 1309–1312.

YANJUN QIAN
DEPARTMENT OF STATISTICAL SCIENCES
& OPERATIONS RESEARCH
VIRGINIA COMMONWEALTH UNIVERSITY
RICHMOND, VA 23220
E-MAIL: yqian3@vcu.edu

JIANHUA Z. HUANG
DEPARTMENT OF STATISTICS
TEXAS A & M UNIVERSITY
COLLEGE STATION, TX 77843
E-MAIL: jianhua@stat.tamu.edu

CHIWOO PARK
DEPARTMENT OF INDUSTRIAL &
MANUFACTURING ENGINEERING
FLORIDA STATE UNIVERSITY
TALLAHASSEE, FL 32301
E-MAIL: chiwoo.park@eng.famu.fsu.edu

YU DING
DEPARTMENT OF INDUSTRIAL &
SYSTEMS ENGINEERING
TEXAS A & M UNIVERSITY
COLLEGE STATION, TX 77843
E-MAIL: yuding@tamu.edu