Smoothing for continuous dynamical state space models with sampled system coefficients based on sparse kernel learning

Article *in* Nonlinear Dynamics · June 2020

DOI: 10.1007/s11071-020-05698-0

CITATION
1

READS
155

3 authors, including:

Nijia Qian
China University of Mining and Technology
7 PUBLICATIONS **6** CITATIONS

SEE PROFILE

Guobin Chang
China University of Mining and Technology
82 PUBLICATIONS **812** CITATIONS

SEE PROFILE

Nonlinear Dyn (2020) 100:3597–3610
https://doi.org/10.1007/s11071-020-05698-0

ORIGINAL PAPER

Smoothing for continuous dynamical state space models with sampled system coefficients based on sparse kernel learning

Nijia Qian · Guobin Chang · Jingxiang Gao

Received: 10 February 2020 / Accepted: 9 May 2020 / Published online: 18 May 2020

Abstract A new smoother for a continuous dynamical state space model with sampled system coefficients is proposed. This is completely different from conventional approaches, such as Rauch–Tung–Striebel smoother. In the proposed method, the state vector as a continuous function of time is represented by kernel models. The state process model, namely the differential equation, is treated as part of the measurement model at the sampling instants of the system coefficients. Sparse solution of the kernel weights is obtained through a special regularization strategy called the Lasso estimator. The optimization problem appearing in the Lasso estimation is solved by the fast iterative shrinkage threshold algorithm. The hyperparameters involved, namely the kernel widths and the regularization coefficients, are selected objectively through generalized cross-validation or corrected Akaike information criterion tailored to the Lasso estimator. A simple two-dimension example is employed in the simulation to demonstrate the application and also the performance of the proposed method. It is shown that the proposed method could provide state vector estimates with satisfactory accuracy not only at the sampling instants of the observations but also at any other instants. The sparsity of the solution could also be clearly seen in the experiment. The proposed method can be viewed as an alternative smoothing method, rather than a replacement for conventional smoothers, due to the difficult model tuning and increased computation load.

Keywords Dynamical state space model · Rauch–Tung–Striebel smoothing · Sparse kernel learning · Fast iterative shrinkage threshold algorithm · Generalized cross-validation · Akaike information criterion

Abbreviations

RTS	Rauch–Tung–Striebel
FISTA	Fast iterative shrinkage threshold algorithm
AIC	Akaike information criterion
GCV	Generalized cross-validation
RBF	Radial basis function
NP	Non-polynomial
MSE	Mean square error

N. Qian · G. Chang (✉) · J. Gao
MNR Key Laboratory of Land Environment and Disaster Monitoring, China University of Mining and Technology, Xuzhou, China
e-mail: guobinchang@hotmail.com

N. Qian · G. Chang · J. Gao
School of Environment Science and Spatial Informatics, China University of Mining and Technology, Xuzhou, China

1 Introduction

A dynamical system is often conveniently represented by a state space model based on Kalman's pioneering work. The state vector may be an abstract one, whose elements may or may not represent real-world variables. However, for a vector to be a state vector, its elements as a whole is not only sufficient but also necessary to completely describe the dynamical property of the system under study. So, estimation of the state vector is an important task in the field of state space system theory. Besides the state vector estimation in real time, namely the so-called filtering, the post-processing estimation, namely the so-called smoothing, is also of significance in many situations. Out of the three kinds of smoothing, namely the fixed-interval, the fixed-lag and the fixed-point, only the first kind, which may be the most important one in the authors' experience, is studied in this work. Smoothing finds wide applications in mission evaluation [1], system/signal reconstruction [2], final result production [3], etc. As long as the state vector is estimated not necessarily in real time, smoothing should be preferred to filtering, as the accuracy of the former is generally higher than the latter except for the only instant at the end of the whole working time span. The smoothing of a dynamical state space model is exactly the focus of this study. For the general theories on state estimation, please consult several good textbooks, e.g., [4–7].

Almost all dynamical state space models are constructed first in continuous time domain. This is due to the fact that the physical/chemical/biological/social rules governing the varying of the state vector are often represented in continuous time domain. And hence discretization is necessary to conduct state estimation on computers or any other digital processing units. This discretization could be invoked in two different phases. One can first discretize the continuous state space model and then perform state estimation for the resulted discrete state space model; alternatively, one could also design continuous state estimation algorithm for the original continuous model which is also in the form of differential equations including the continuous Riccati equations and then solve or discretize these equations with certain algorithms. For both kinds, discretization errors are inevitable due to the following two reasons. First, for either kind of discretization, we are confronted with solving differential equation of a certain type. These solutions are often numerical rather than analytical. This is also often called numerical integration. And hence numerical errors are often inevitable due to, e.g., series truncation. Second, though in continuous-time form, the state space model may involve coefficients/parameters which are only available at discrete instants. Or in other words, the model or system coefficients can only be obtained by sampling using a digital sensor, just as the observations of the dynamical state space model but maybe with different sampling rates. In the interval between two consecutive sampling instants, the time-variant coefficients would often be assumed constant or other simple model constructed using the discrete samples. This could also result in modeling errors. This case with sampled system coefficients could be found in, e.g., integrated navigation involving inertial navigation system [8, 9] or attitude determination involving rate gyros [10, 11]. This case is also the topic of this study. Also note for this case, the process noise may be exactly the result of the measurement noise of the sampled system coefficients. With the above analysis, it could be made clear that there may be some room for improving the smoothing accuracy by reducing discretization errors, compared with conventional smoothing methods.

Only the first of the two kinds of discretization is considered in this work, namely first discretizing the continuous model and then designing a smoothing approach for this discretized model. To be more specific, the famous RTS smoothing approach [12] is chosen as a representative among many other approaches belonging to the first kind discretization, e.g., the forward and backward two-filter approach [13]. The RTS smoother, as a direct extension of the Kalman filter to the smoothing case, is called optimal and canonical in the field of state estimation, as long as the following prerequisites are fulfilled: known [14, 15] and Gaussian [16–18] noise distributions, none model uncertainties [19], linear system model [20, 21]. This seems to tell that there is no space to further improve the accuracy, though further improving the numerical efficiency could be another case [22]. However, it is important and also not hard from the analysis in the above paragraph, to realize that the RTS smoother is optimal only for the discrete state space model. This optimality could not be extended to the smoothing of original continuous-time model in general. This is due to the discretization errors

introduced in the process of obtaining the discrete model from the original continuous model. Putting it in another way, the discrete model could only be viewed as an approximate rather than an equivalent model of the original continuous model. One should be clearly aware of the fact that the estimate of the original continuous model is exactly what we should be interested in, though maybe at discrete instants, rather than that of the discrete model [23]. So, it should not be viewed as a completely surprise to see the improved accuracy of the proposed approach compared to the RTS smoothing. Note that there are also prices to pay for the accuracy improvement of the proposed method, such as increased difficulty of tuning and heavier computational load. So, from an engineering viewpoint, we only mean to provide an alternative rather than replace the conventional approaches; and the easily tuned and computationally efficient RTS smoother should also be employed as long as its accuracy fulfills the mission requirements.

An alternative approach is proposed to state smoothing of a continuous-time dynamical state space model with sampled system coefficients. The following five key points are vital in developing this approach. First, the state process differential equations at system coefficients sampling instants are treated as pseudo-measurement equations, besides the real measurement equations for the sampled observations. This treatment is completely different from conventional state space theory. By this treatment, the conventional discretization or numerical integration of the differential equations of either of the two kinds are avoided. Second, the state vector, as function of time, is represented by kernel models [24, 25]. Kernel model provides an elegant parameterization of the state vector of which analytical solution is not easily available from the state space model. With the kernel model, the problem becomes estimating the kernel weights rather than estimating the state vector directly. There is also a byproduct with the kernel model, namely that we would obtain an analytical function for the state vector rather than series of state vector values at discrete instants. With the above two key points, the smoothing problem degenerates to the estimation of the kernel weights with the measurement equations including the state process differential equations and the real observation's measurement equations. Third, sparse regularization is introduced in the estimation. Regularization is necessary because there are too many parameters to be estimated which makes the problem badly conditioned [26]. Rather than the L2-norm or Tikhonov regularization [27], the sparse L1-norm regularization is used [28]. With this regularization, the estimated parameters are not only shrinked but also sparsified [29]. This could be numerically beneficial in evaluating state vector at certain instants with the obtained kernel model. Fourth, numerically efficient algorithms, e.g., the FISTA [30] among many others, are employed to solve the optimization problem in parameter estimation. Fifth, hyperparameters, namely the regularization coefficients and the kernel widths, are chosen objectively with the GCV [31, 32] or corrected AIC [33, 34], both tailored to the specific L1-norm regularization problem [35].

The rest of the paper is organized as follows. Section 2 is devoted to the problem formulation and terminology introduction. In this section, the dynamical state space model in continuous time, the discretization and the RTS smoother are introduced. The main methodology is developed in Sect. 3. In the three subsections of this section, the measurement model, the parameter estimation and the hyperparameter selection are presented, respectively. In Sect. 4, simulation of a two-dimensional example is carried out to show the application and the performance of the proposed method, followed by a discussion. It is concluded in Sect. 5.

2 Dynamical state space model and its discretization and optimal smoothing

A continuous-time dynamical state space model can be expressed by the following process and observation equations [4, 5]:

$$\dot{x}_t = \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{w}_t \tag{1}$$

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k \tag{2}$$

where the subscript t denotes the time and k represents the epoch or instant index; a variable with a dot over it denotes its derivative with respect to time; $\boldsymbol{x}_t$ refers to the target state vector to be estimated, while $\boldsymbol{y}_k$ refers to the observation vector; $\boldsymbol{A}_t$ denotes the system matrix, and $\boldsymbol{H}_k$ represents the design matrix of the measurement model; $\boldsymbol{w}_t$ is the process noise with power spectral density matrix $\boldsymbol{Q}$, while $\boldsymbol{v}_k$ is the measurement noise with covariance matrix $\boldsymbol{R}$. In this paper, we

assume that the dimensions of state vector, observation vector and the number of total epochs are n, m and l, respectively.

The following two are noted concerning the above model. First, in some publications, the above model is called hybrid continuous and discrete rather than simply continuous as in this work, due to the continuous nature of (1) and the discrete nature of (2), see, e.g., [22, 36–38]. Completely continuous model, namely with both (1) and (2) in continuous time domain, is without the scope of this study. In fact, this completely continuous model can hardly be encountered in modern-time real applications. Second, the state process model, namely the differential equation in (1), though in continuous time, may involve coefficients only available at discrete instants. These coefficients may appear as arguments in the system matrix $\boldsymbol{A}_t$, or in input vector which is omitted for simplicity and without loss of generality. The reason for these coefficients being available at discrete instant is often that they are also measured by sensors just as the observation $\boldsymbol{y}_k$ in (2), however, maybe with different sampling rates. An example of this case is the integrated navigation or attitude filtering involving inertial sensors. In this example, the state process model (1) involves the angular rates which are measured by rate gyros at discrete instants. See, e.g., [39] and [11] for details on integrated navigation and attitude filtering, respectively.

In traditional state estimation methods, the discrete state process model is obtained by discretization as follows [4]:

$$\boldsymbol{x}_k = \boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1} \tag{3}$$

with,

$$\boldsymbol{F}_{k-1} \approx \boldsymbol{I}_n + \Delta \boldsymbol{A}_{t_{k-1}} \tag{4}$$

$$\begin{aligned}\boldsymbol{Q}_{k-1} &= \mathrm{cov}[\boldsymbol{w}_{k-1}] \\ &\approx \Delta \boldsymbol{Q} + \frac{\Delta^2}{2}\boldsymbol{A}_{t_{k-1}}\boldsymbol{Q} + \frac{\Delta^2}{2}\boldsymbol{Q}\boldsymbol{A}_{t_{k-1}}^{\mathrm{T}} + \frac{\Delta^3}{3}\boldsymbol{A}_{t_{k-1}}\boldsymbol{Q}\boldsymbol{A}_{t_{k-1}}^{\mathrm{T}}\end{aligned} \tag{5}$$

In the above, Δ denotes the sampling interval of $\boldsymbol{y}_k$. The following two are noted concerning the above discretization. First, note the approximation mark in both (4) and (5). They are approximate equations in general rather than rigorous ones due to the series truncation. It is exactly the approximation here that introduces the discretization errors. It could be foreseen that the discretization errors would decrease as the sampling interval decreases. Second, other discretization or numerical integration methods other than the above one, such as Runge–Kutta method, or those with more series terms, could be employed to reduce the discretization errors to some extent. However, exact discretization of continuous linear systems usually brings significant troubles in series expression and computation complexity, etc. So, the proposed smoothing algorithm, compared with RTS smoother, is possible to improve the accuracy of state smoothing by dealing with discretization more accurately. In other words, we did not aim to develop an alternative discretization method against those with more series terms, though it really can be viewed as a special discretization method. In this paper, the RTS smoothing algorithm is employed for contrast, and the comparison with other conventional discretization or numerical integration approaches is out of the scope of this study.

To be complete, the RTS smoothing algorithm for the discrete state space model (3) and (2) is presented in Table 1.

In Table 1, $\hat{\boldsymbol{x}}$ refers to the estimate of state vector $\boldsymbol{x}$, with its covariance matrix $\boldsymbol{P}$, of which the inverse is $\boldsymbol{I}$. The superscripts '+' and '−' denote the estimates are posteriori and priori, respectively, while the subscript f denotes the forward filtering. $\boldsymbol{K}$ represents the Kalman filter gain.

The above RTS is chosen as a benchmark to check the performance of the proposed method. It is again stressed that the RTS smoother is statistically optimal but only for the discrete model (3) and (2). However, this model itself is not rigorous but approximate to the original model (1) and (2). Also note that with RTS, estimates of the state vector are only available at sampling instants of the observation $\boldsymbol{y}_k$. Some additional interpolation approaches would be necessary to get estimates of the state vector at other instants.

3 A new smoothing approach based on sparse kernel learning

This section is split into three parts to introduce the measurement model, the parameter estimation and the hyperparameters selection, respectively.

Table 1 Algorithmic flow of the RTS smoother

Step	Operation
Initialize the forward filter	$\hat{\boldsymbol{x}}_{f0} = E(\boldsymbol{x}_0)$
For $k = 0$	$\boldsymbol{P}_{f0} = E\left[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_{f0})(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_{f0})^{\mathrm{T}}\right]$
Execute the standard forward Kalman filter	$\boldsymbol{P}_{fk}^{-} = \boldsymbol{F}_{k-1}\boldsymbol{P}_{f,k-1}^{+}\boldsymbol{F}_{k-1}^{\mathrm{T}} + \boldsymbol{Q}_{k-1}$
For $k = 1, \ldots, l$ (where l is the final time)	$\boldsymbol{K}_{kf} = \boldsymbol{P}_{fk}^{-}\boldsymbol{H}_k^{\mathrm{T}}\left(\boldsymbol{H}_k\boldsymbol{P}_{fk}^{-}\boldsymbol{H}_k^{\mathrm{T}} + \boldsymbol{R}_k\right)^{-1}$
	$\hat{\boldsymbol{x}}_{fk}^{-} = \boldsymbol{F}_{k-1}\hat{\boldsymbol{x}}_{f,k-1}^{+}$
	$\hat{\boldsymbol{x}}_{fk}^{+} = \hat{\boldsymbol{x}}_{fk}^{-} + \boldsymbol{K}_{fk}\left(\boldsymbol{y}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_{fk}^{-}\right)$
	$\boldsymbol{P}_{fk}^{+} = (\boldsymbol{I} - \boldsymbol{K}_{fk}\boldsymbol{H}_k)\boldsymbol{P}_{fk}^{-}$
Initialize the RTS smoother	$\hat{\boldsymbol{x}}_l = \hat{\boldsymbol{x}}_{fl}^{+}$
For $k = l$	$\boldsymbol{P}_l = \boldsymbol{P}_{fl}^{+}$
Execute the RTS smoother equations	$\boldsymbol{I}_{f,k+1}^{-} = \left(\boldsymbol{P}_{f,k+1}^{-}\right)^{-1}$
For $k = l - 1, \ldots, 1, 0,$	$\boldsymbol{K}_k = \boldsymbol{P}_{fk}^{+}\boldsymbol{F}_k^{\mathrm{T}}\boldsymbol{I}_{f,k+1}^{-}$
	$\boldsymbol{P}_k = \boldsymbol{P}_{fk}^{+} - \boldsymbol{K}_k\left(\boldsymbol{P}_{f,k+1}^{-} - \boldsymbol{P}_{k+1}\right)\boldsymbol{K}_k^{\mathrm{T}}$
	$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{fk} + \boldsymbol{K}_k\left(\hat{\boldsymbol{x}}_{k+1} - \hat{\boldsymbol{x}}_{f,k+1}^{-}\right)$

3.1 Measurement model

The measurement model of the observation of the state space model, namely the one in (2), is only part of the measurement model in the sparse kernel learning smoothing method. The other part is developed in the following. We first make it clear that in the case considered in our work, and the sampling rate of the system coefficients is higher than or equal to that the of the observation $\boldsymbol{y}_k$. Synchronization assumption is made here without loss of generality, namely that the sampling instants of $\boldsymbol{y}_k$ are assumed to be also the sampling instants of the system coefficients.

We start with the introduction of the kernel model for the state vector. Each element of the state vector, e.g., the i-th component $x_t^{(i)}$ of the state vector $\boldsymbol{x}_t$, is viewed as a function of the time. It is represented as a kernel model, namely as weighted sum of the following kernel functions:

$$x_t^{(i)} = \sum_{j=1}^{q} \beta_{ij} K_i(t, t_j) \tag{6}$$

In the above, $K(\cdot,\cdot)$ denotes a kernel function, which is chosen empirically. The argument t_j of the kernel function denotes the kernel location, which is also empirically chosen. In this study, these kernel locations are chosen as all the sampling instants, totaling q. According to the above assumption on sampling rates, we know that $q = pl$, with p being the ratio of the sampling rate of the system coefficients to that of the observation $\boldsymbol{y}_k$. The other argument t of the kernel function denotes the time instants at which the state vector is to be estimated. This argument is continuous, implying that the model (6) can output state vector estimates at any given time, not necessarily the sampling instants. Kernel weights β_{ij} are the parameters that are to be estimated. With this kernel model, we do not estimate the state vector directly; rather, we first estimate the parameters β_{ij} and then use (6) to give state vector at any given time. From (6), it seems all components of $\boldsymbol{x}$ are independent and this is not the truth in most real applications; however, we introduce the correlation between them through system/process model (1). Without losing generality, here Gaussian RBF kernel is adopted as follows:

$$K_i(t, t_j) = \exp\left[-\frac{(t - t_j)^2}{2\sigma^2}\right] \tag{7}$$

where σ is the width of RBF kernel, which controls the radial action range of the kernel. The following is noted concerning the kernel model, which is one of the key points of the proposed method. The kernel model, also called the kernel trick in other applications, provides a powerful tool for parameterizing the state

vector of complex time-varying nature. The kernel method has sound mathematical roots [24] and finds wide applications in diverse fields of statistics, signal processing and machine learning [40].

We can reconstruct (1) into a pseudo-measurement equation as follows:

$$z_j = 0 = \boldsymbol{A}_j \boldsymbol{x}_j - \dot{\boldsymbol{x}}_j + \boldsymbol{w}_j \tag{8}$$

In the above, the subscript j denotes the time instant at which the system coefficients are available. It is well-known that the process noise is system itself driven, while the measurement noise results from measurement conditions. The reason why we can directly treat the process noise as pseudo-measurement noise is that sampling the model/system coefficients or measuring input vector (omitted in this paper) using a digital sensor usually inevitably introduces measurement noise. These measurement noise could be considered exactly as (part of) the process noise in system model. Examples of such applications include inertial navigation system [41, 42] or attitude determination involving rate gyros [10, 11]. Now, it is readily known from (6) that $\dot{x}_t^{(i)} = \sum_{j=1}^{q} \beta_{ij} \frac{\partial K_i(t,t_j)}{\partial t} = \sum_{j=1}^{q} -\beta_{ij} \frac{t-t_j}{\sigma^2} \exp\left[-\frac{(t-t_j)^2}{2\sigma^2}\right]$.

By considering both (8) and (2), the following overall measurement model is obtained:

$$\boldsymbol{\xi} = \boldsymbol{G}\boldsymbol{\beta} + \boldsymbol{\eta} \tag{9}$$

where $\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{z}_1^{\mathrm{T}} & \boldsymbol{z}_2^{\mathrm{T}} & \cdots & \boldsymbol{z}_q^{\mathrm{T}} & \boldsymbol{y}_1^{\mathrm{T}} & \boldsymbol{y}_2^{\mathrm{T}} & \cdots & \boldsymbol{y}_l^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ is the measurement vector of $(qn + lm) \times 1$ dimensions; $\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{w}_1^{\mathrm{T}} & \boldsymbol{w}_2^{\mathrm{T}} & \cdots & \boldsymbol{w}_q^{\mathrm{T}} & \boldsymbol{v}_1^{\mathrm{T}} & \boldsymbol{v}_2^{\mathrm{T}} & \cdots & \boldsymbol{v}_l^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ denotes measurement error vector of $(qn + lm) \times 1$ dimensions, corresponding to its covariance matrix $\boldsymbol{Q}_{\eta\eta}$. $\boldsymbol{G}$ is nothing but the design matrix of the above measurement model. To be specific, substituting $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ in (2) and (8) with kernel representation in (6), we can readily construct the design matrix $\boldsymbol{G}$. The parameter vector $\boldsymbol{\beta}$ of $(nq \times 1)$ dimensions is constituted of all kernel weights β_{ij} and is to be estimated.

This completes the presentation of the measurement model. To emphasize, we should first get a good estimate for $\boldsymbol{\beta}$ and then substitute this estimate into the model (6) from which state vector at any given time could be readily output. The measurements used in this model include not only the real measurements, namely $\boldsymbol{y}_k$, but also the pseudo ones, namely $z_j = 0$.

3.2 Parameter estimation

For the measurement model (9), conventional methods, such as least-squares or maximum-likelihood, could be used to estimate the parameter vector $\boldsymbol{\beta}$ [43]. However, these methods can hardly be satisfactory. This is due to the fact that the model (9) is badly conditioned in general. So, regularization is often necessary [27]. Regularization, also called ridge estimation or shrinkage estimation, though producing biased estimates of the parameter, often produces better model, namely with better prediction or generalization performances [44]. Rather than the L2 norm regularization, also called the Tikhonov regularization, the L1 norm regularization, also called the Lasso estimator, is adopted [28]. With Lasso, we estimate parameter $\boldsymbol{\beta}$ by minimizing the following regularization cost function:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ (\boldsymbol{\xi} - \boldsymbol{G}\boldsymbol{\beta})^{\mathrm{T}} \boldsymbol{Q}_{\eta\eta}^{-1} (\boldsymbol{\xi} - \boldsymbol{G}\boldsymbol{\beta}) + \mu \|\boldsymbol{\beta}\|_1 \right\} \tag{10}$$

where the quadratic term $(\boldsymbol{\xi} - \boldsymbol{G}\boldsymbol{\beta})^{\mathrm{T}} \boldsymbol{Q}_{\eta\eta}^{-1} (\boldsymbol{\xi} - \boldsymbol{G}\boldsymbol{\beta})$ is nothing but the cost function of the least-square estimates of Eq. (9). The L1 norm regularization term, acting as a penalty constraint, which makes the sum of absolute values of $\boldsymbol{\beta}$ elements minimum, leads to a sparse solution. This kind of sparse solution implies that some of its elements would exactly be zero. The regularization parameter μ, always greater than 0, plays a key role in manipulating the tradeoff between the goodness of fit to the measurements and solution sparsity. The solution $\boldsymbol{\beta}$ will be sparser with an increasing μ since an increasing penalty term is exerted to its cost function. How to determine the hyperparameter μ together with the other hyperparameter, namely the kernel width σ appearing in (7) objectively and optimally will be introduced in Sect. 3.3.

The reason why we let $\boldsymbol{\beta}$ sparse can be explained mainly from the perspective of model complexity. From the viewpoint of modeling, there is no "true value" of $\boldsymbol{\beta}$. When we estimate it, instead of approaching its "true value" as much as possible, we always wish to make the prediction error of the model as small as possible. From this point of view, different $\boldsymbol{\beta}$ can correspond to the same or similar model prediction accuracy. Besides, regularization constraint

such as sparse L1 norm regularization or dense L2 norm regularization is always necessary in the proposed method, because the problem of parameter estimation is ill-conditioned (this ill-conditioned property is due to the almost linear correlation between the different rows of the measurement matrix $\boldsymbol{G}$). So in terms of accuracy, $\boldsymbol{\beta}$ is not required to be sparse; in other words, L1 norm regularization is not required. Choosing L1 norm regularization to make $\boldsymbol{\beta}$ sparse is based on the complexity of the model. It is entirely possible that the same or similar prediction accuracy (smoothness) can be achieved by using L2 norm regularization, but the model will be too complex, especially for $\boldsymbol{\beta}$ with massive dimensions.

As a final note, the regularizations of different parts of the parameter vector $\boldsymbol{\beta}$ can correspond to different regularization coefficients μ. This would in general results in further improved accuracy at the price of tuning more hyperparameters. This would not present any difficulties in the following theory development, because by simple reparameterization, we can easily put this case in the same form as (10). The following is an example. Assuming the kernel weights are arranged into two parts, namely $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$. Then, the measurement model in (9) changes to $\boldsymbol{\xi} = \boldsymbol{A}\boldsymbol{\beta}_1 + \boldsymbol{B}\boldsymbol{\beta}_2 + \boldsymbol{e}$, and the equation in (10) changes to the following:

$$\begin{pmatrix} \hat{\boldsymbol{\beta}}_1 & \hat{\boldsymbol{\beta}}_2 \end{pmatrix} = \underset{\boldsymbol{\beta}_1 \quad \boldsymbol{\beta}_2}{\arg\min} \left\{ (\boldsymbol{\xi} - \boldsymbol{A}\boldsymbol{\beta}_1 - \boldsymbol{B}\boldsymbol{\beta}_2)^T \boldsymbol{Q}_{\boldsymbol{ee}}^{-1} (\boldsymbol{\xi} - \boldsymbol{A}\boldsymbol{\beta}_1 - \boldsymbol{B}\boldsymbol{\beta}_2) + \mu_1 \|\boldsymbol{\beta}_1\|_1 + \mu_2 \|\boldsymbol{\beta}_2\|_1 \right\} \tag{11}$$

Using the transformations: $\boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{\beta}_1 & \frac{\mu_2}{\mu_1}\boldsymbol{\beta}_2 \end{bmatrix}^{\mathrm{T}}$ and $\boldsymbol{C} = \begin{bmatrix} \boldsymbol{A} & \frac{\mu_1}{\mu_2}\boldsymbol{B} \end{bmatrix}$, we can turn Eq. (11) into the same form as Eq. (10), shown as follows:

$$\hat{\boldsymbol{\gamma}} = \arg\min_{\boldsymbol{\gamma}} \left\{ (\boldsymbol{\xi} - \boldsymbol{C}\boldsymbol{\gamma})^T \boldsymbol{Q}_{\boldsymbol{ee}}^{-1} (\boldsymbol{\xi} - \boldsymbol{C}\boldsymbol{\gamma}) + \mu_1 \|\boldsymbol{\gamma}\|_1 \right\} \tag{12}$$

Without loss of generality, the following methodology development is only based on (10).

Concerning the Lasso estimator or the L1 norm regularization, the following three are noted. First, one key reason to employ the L1 norm regularization is to make the solution sparse. It should be noted that there are other ways to produce sparse solution, e.g., using ε-insensitive loss function rather than the sum-of-squares one as in the support vector machine [45], or the structured Bayesian estimation as in the relevance vector machine [46]. Second, one of the reasons why we choose L1 norm regularization rather than other sparsity-promoting approaches is that the associate problem is a convex optimization [47, 48]. This makes many highly efficient numerical algorithms for convex optimization readily applicable, e.g., the FISTA [30], which is employed in this study. In fact, Lp norm regularization with $p \geq 1$ resulting in convex optimization. Further, Lp norm regularization with $p \leq 1$ promoting sparsity of the solution. So, it becomes clear why the L1 norm is so special. Third, the L1 regularization has find wide applications, maybe with different names, e.g., the total variation in image processing [49], the atom decomposition [50] or soft-threshold denoising [51] in signal processing, or the compressed sensing [52, 53].

The convex optimization problem in Eq. (10) can boil down to a second-order cone programming problem, and so far many algorithms have been proposed to solve it [48, 54]. In this work, the most popular FISTA is employed [30]. FISTA is a first-order or gradient algorithm with the famous Nesterov acceleration strategy [47], so it can reach a fast convergence with computational simplicity. Specifically, the iteration steps of FISTA algorithm for Eq. (10) are as follows:

$$\begin{aligned} \boldsymbol{\beta}_k &= T_{\lambda\mu}\left(\boldsymbol{\vartheta}_k - 2\lambda \boldsymbol{G}^{\mathrm{T}} \boldsymbol{Q}_{\boldsymbol{\eta\eta}}^{-1} (\boldsymbol{G}\boldsymbol{\vartheta}_k - \boldsymbol{\xi})\right) \\ s_{k+1} &= \frac{1 + \sqrt{1 + 4s_k^2}}{2} \\ \boldsymbol{\vartheta}_{k+1} &= \boldsymbol{\beta}_k + \frac{s_k - 1}{s_{k+1}} (\boldsymbol{\beta}_k - \boldsymbol{\beta}_{k-1}) \end{aligned} \tag{13}$$

where $\lambda = \frac{1}{2\lambda_{\max}(\boldsymbol{G}^{\mathrm{T}} \boldsymbol{Q}_{\boldsymbol{\eta\eta}}^{-1} \boldsymbol{G})}$ is a step size, $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue. The method for

calculating the j-th element of $\boldsymbol{x}$ in soft thresholding function reads as $[T_\alpha(\boldsymbol{x})]_j = (|x_j| - \alpha)_+ \mathrm{sgn}(x_j)$. The hinge function and sign function involved in soft thresholding function read as $(|x_j| - \alpha)_+ = \begin{cases} |x_j| - \alpha, & \text{for } |x_j| > \alpha \\ 0, & \text{for } |x_j| \le \alpha \end{cases}$ and $\mathrm{sgn}(x_j) = \begin{cases} 1, & \text{for } x_j > 0 \\ 0, & \text{for } x_j = 0 \\ -1, & \text{for } x_j < 0 \end{cases}$, respectively. The above iteration algorithm is initialized with $\boldsymbol{\beta}_0 = \boldsymbol{\vartheta}_1 = \mathbf{0}$ and $s_1 = 1$. The following stopping criterion is adopted in iterations: $\frac{\|\boldsymbol{\beta}_k - \boldsymbol{\beta}_{k-1}\|_1}{nq} < \tau$, where τ is a predetermined threshold, and we take $\tau = 10^{-8}$ in this paper.

After the estimated $\hat{\boldsymbol{\beta}}$ is obtained and substituted into Eq. (6), we can get the smoothed estimates at sampling instants as well as other arbitrary instants.

3.3 Hyperparameter selection

In the above estimation method, parameter vector $\boldsymbol{\beta}$ can be estimated with given hyperparameters, namely μ and σ. Different hyperparameters would bring different parameter estimates, and hence different models are obtained. So, in order to obtain as accurate as possible state estimates with the sparse kernel learning, it is necessary to optimize hyperparameters. However, the exhaustive trial method is needed to determine the absolute optimal hyperparameters, which is a NP-hard problem. We can settle for the relatively suboptimal hyperparameters from a predefined candidate set. The goodness or badness of the model obtained with a certain pair of hyperparameters can be represented by the GCV or AICc, detailed in the following.

The number of effective parameters or generalized degree-of-freedom in parameter estimation is $n_1 = \sum_{i=1}^{l} \frac{\partial \hat{\xi}_i}{\partial \xi_i} = \sum_{i=1}^{l} \mathrm{cov}\left[\hat{\xi}_i, \xi_i\right]$ according to the Stein unbiased risk estimate concept [44]. According to this theory, [35] pointed out that the number n_0 of nonzero elements in parameter estimation of Lasso problems is an asymptotically uniformly unbiased estimates of the number of effective parameters n_1. Applying this conclusion to GCV [31, 32] or AICc [33, 34] of Eq. (8), we can derive the followings:

$$\mathrm{GCV}(\mu, \sigma) = \frac{n\mathrm{RSS}(\mu, \sigma)}{(n - n_1)^2} \approx \frac{n\mathrm{RSS}(\mu, \sigma)}{(n - n_0)^2} \tag{14}$$

or

$$\begin{aligned} \mathrm{AIC_c}(\mu, \sigma) &= n \ln \frac{RSS(\mu, \sigma)}{n} + 2n_1 + \frac{2n_1(n_1 + 1)}{n - n_1 - 1} \\ &\approx n \ln \frac{RSS(\mu, \sigma)}{n} + 2n_0 + \frac{2n_0(n_0 + 1)}{n - n_0 - 1} \end{aligned} \tag{15}$$

where $\mathrm{RSS}(\mu, \sigma) = \left[\boldsymbol{\xi} - \boldsymbol{G}\hat{\boldsymbol{\beta}}(\mu, \sigma)\right]^{\mathrm{T}} \boldsymbol{Q}_{\eta\eta}^{-1} \left[\boldsymbol{\xi} - \boldsymbol{G}\hat{\boldsymbol{\beta}}(\mu, \sigma)\right]$ denotes the sum of squares of the residuals of Eq. (9). n_0 refers to the number of nonzero elements in $\hat{\boldsymbol{\beta}}$ while n refers to the number of all elements in $\hat{\boldsymbol{\beta}}$. Note in the above, the arguments μ and σ are explicitly shown to clearly tell that those variables are functions of these arguments. The hyperparameters μ and σ, which make the minimum values of Eqs. (14) or (15), are the most ideal ones.

4 Experiment

4.1 Simulation test and analysis

To evaluate the effectiveness of the proposed sparse kernel learning smoothing method for state space model, a simple tracking problem is used to demonstrate its application and performance. First consider a dynamical system governed by the following differential equation:

$$\dot{\boldsymbol{x}}_t = \begin{bmatrix} \dot{p}_t \\ \dot{u}_t \end{bmatrix} = \begin{bmatrix} 0.02 & 0.005 \\ 0.005 & 0.01 \end{bmatrix} \begin{bmatrix} p_t \\ u_t \end{bmatrix} + \boldsymbol{s}_t = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{s}_t \tag{16}$$

The following two are noted concerning the above equation. First, it is the true values of the state vector that are governed by the above model. Second, the input vector $\boldsymbol{s}_t$ is explicitly shown, though its true values are assumed to be zero, without loss of generality. So, series of true values of the state vector are generated strictly according to the above model. These true values are used in the following as reference to check the estimation accuracies of different smoothing approaches. The generation of the true values is described as follows. We discrete the

above differential equations with a relatively smaller interval, e.g., $\Delta = t_k - t_{k-1}$=0.1. The obtained discretization equation $\boldsymbol{x}_k = \bar{\boldsymbol{F}}\boldsymbol{x}_k$ is exactly the one that is used for generating the reference truth values. Here, we take $\boldsymbol{x}_0$ = [0.5; 1] as the initial value of state vector. Then, we substitute these true values into the following observation model to generate observations with a relatively larger sampling interval $\Delta = t_k - t_{k-1}$=1.

$$\boldsymbol{y}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k \tag{17}$$

where $\boldsymbol{y}_k$ contains both the measurements of p_k and u_k to avoid the unobservable problem and $\boldsymbol{H}_k$ = [1, 0; 0, 1] is the corresponding measurement matrix. We take the covariance matrix of measurement noise $\boldsymbol{v}_k$ as $\boldsymbol{R}_k$ = [0.015^2, 0; 0, 0.01^2]. These measurements are to be used in the following smoothing tests.

Then, the details of the smoothing are presented as follows. Note that the model (16) represents the ideal information, which can hardly be completely accessible in practice. For instance, the input vector $\boldsymbol{s}_t$ is not exactly known but can only be measured by a sensor. So there would be two consequences: the first is that only at discrete instants that the values of $\boldsymbol{s}_t$ are available; the second is that noises are introduced into (16) when one uses the measurement of $\boldsymbol{s}_t$ in place of $\boldsymbol{s}_t$ itself. The noise here is exactly (part of) the process noise in the state process differential equation used in the smoothing, shown in the following.

$$\begin{aligned}\dot{\boldsymbol{x}}_t &= \begin{bmatrix}\dot{p}_t \\ \dot{u}_t\end{bmatrix} = \begin{bmatrix}0.02 & 0.005 \\ 0.005 & 0.01\end{bmatrix}\begin{bmatrix}p_t \\ u_t\end{bmatrix} + \tilde{\boldsymbol{s}}_t + \boldsymbol{w}_t \\ &= \boldsymbol{A}\boldsymbol{x}_t + \tilde{\boldsymbol{s}}_t + \boldsymbol{w}_t\end{aligned} \tag{18}$$

where $\tilde{\boldsymbol{s}}_t$ denotes the measurement of $\boldsymbol{s}_t$. Here, $\boldsymbol{w}_t$ denotes the process noise, with its power spectral density matrix $\boldsymbol{Q}$ = [0.01^2, 0; 0, 0.01^2].

For the proposed sparse kernel learning-based smoothing method, Eq. (18) at sampling instants of $\tilde{\boldsymbol{s}}_t$ is used as part of the measurement equations. For the conventional RTS smoother, a differential equation with interval exactly the same as the sampling interval of $\boldsymbol{y}_k$ should be firstly obtained by numerically integrating (18). This is denoted as $\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_k + \boldsymbol{w}_k$.

The following four methods are considered in this experiment. For all the four methods, the sampling interval of $\boldsymbol{y}_k$ is 1 s. To be fair, we only compared Method #1 with Method #2 because they have the same sampling interval. Besides, the last three of the four methods are employed to check the smoothing performance of our method with increasing sampling rates.

- Method #1: the conventional RTS smoother;
- Method #2: the proposed method with the sampling rate of the $\tilde{\boldsymbol{s}}_t$ being the same as $\boldsymbol{y}_k$;
- Method #3: the proposed method with the sampling rate of the $\tilde{\boldsymbol{s}}_t$ being 5 times as $\boldsymbol{y}_k$;
- Method #4: the proposed method with the sampling rate of the $\tilde{\boldsymbol{s}}_t$ being 10 times as $\boldsymbol{y}_k$.

We conduct 200 Monte Carlo (MC) runs and evaluate smoothing performance in terms of standard deviation (STD), root mean square (RMS) and averaged RMS (ARMS) of state estimates [55, 56]. Taking p component in state vector $\boldsymbol{x}$ as an example, the statistic results can be calculated as follows.

$$\begin{cases}\mathrm{STD}_{p,k} = \sqrt{\dfrac{1}{l}\sum_{s=1}^{l}\left(p_k^s - \bar{p}_k^s\right)^2} \\ \mathrm{RMS}_{p,k} = \sqrt{\dfrac{1}{l}\sum_{s=1}^{l}\left(p_k^s - \hat{p}_k^s\right)^2} \\ \mathrm{ARMS}_p = \sqrt{\dfrac{1}{lt}\sum_{k=1}^{t}\sum_{s=1}^{l}\left(p_k^s - \hat{p}_k^s\right)^2}\end{cases} \tag{19}$$

where p_k^s, $\hat{p}_k^s$ and $\bar{p}_k^s$ represent the estimated, the true and the mean value (in each MC run), respectively. The symbol t refers to the number of MC experiments, and here it is taken as 200 as previously mentioned.

First, the estimation errors statistics of the Method #1 and Method #2 for the two elements of the state vector are compared, as shown in Figs. 1, 2 and Table 2. From them, we can find that the proposed method, namely Method #2, outperforms the conventional RTS smoother, namely the Method #1, in terms of estimation accuracy. This at least demonstrates the effectiveness of the proposed method.

Besides, the sparsity of the proposed Method #2 is checked. The sparsity may be an important factor in the subsequent usage of the obtained model. With fewer nonzero parameter elements, the computational load would be lighter in using the model to calculate the state vector at a given instant. The statistics of sparsity rate is presented in Table 3. We can conclude the sparsity is significant for all cases. To be more specific, the average percents of zero elements are larger than 70% for all components.

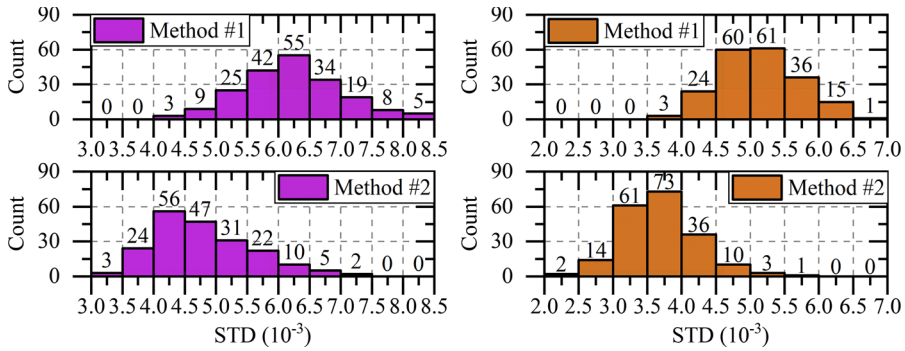

Fig. 1 STD distribution histogram of the p component error (Left) and u component errors (Right) with Method #1 and Method #2. The results of 200 Monte Carlo tests are displayed

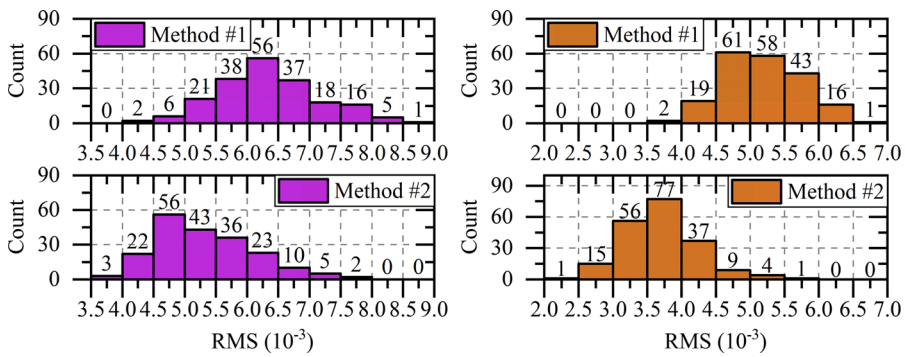

Fig. 2 RMS distribution histogram of the p component error (Left) and u component errors (Right) with Method #1 and Method #2. The results of 200 Monte Carlo tests are displayed

Table 2 ARMS statistics of Method # 1 and Method # 2 based on 200 MC experiments

Approaches	p component	u component
Method #1	6.36×10^{-3}	5.19×10^{-3}
Method #2	4.84×10^{-3}	3.69×10^{-3}

Table 3 The sparsity rate of $\boldsymbol{\beta}$ in Method #2 based on 200 MC experiments

Component	Max. (%)	Min. (%)	Mean (%)
p component	78.0	39.0	71.1
u component	78.0	37.0	72.2

Then, we evaluate the three versions of the proposed method and display their estimation error in one MC experiment in Fig. 3. We found that as the sampling rate of the system coefficients, namely the sampling rate input $\tilde{s}_t$ increases, the estimation accuracies also increase. This should not be a surprise, because it can be readily explained from the overall measurement model in (9). With the increase in the sampling rates, there are more and more measurements. So, the measurement model becomes stronger and stronger; and finally superior performance of the model could be expected.

Finally, the results of the proposed method as an analytical model are also checked in the above MC experiment. To be more specific, with the proposed method, in any of its three different versions, estimates of the state vector at any given instants, not necessarily the sampling ones, could be output. First, the estimation errors at 0.1 s interval are shown in Fig. 4 for the two components, respectively. It is clearly seen that the overall pattern of the performance as functions of time does not changed significantly from Fig. 3. This is obviously a positive finding, as it shows the stability of the proposed method across the time axis.

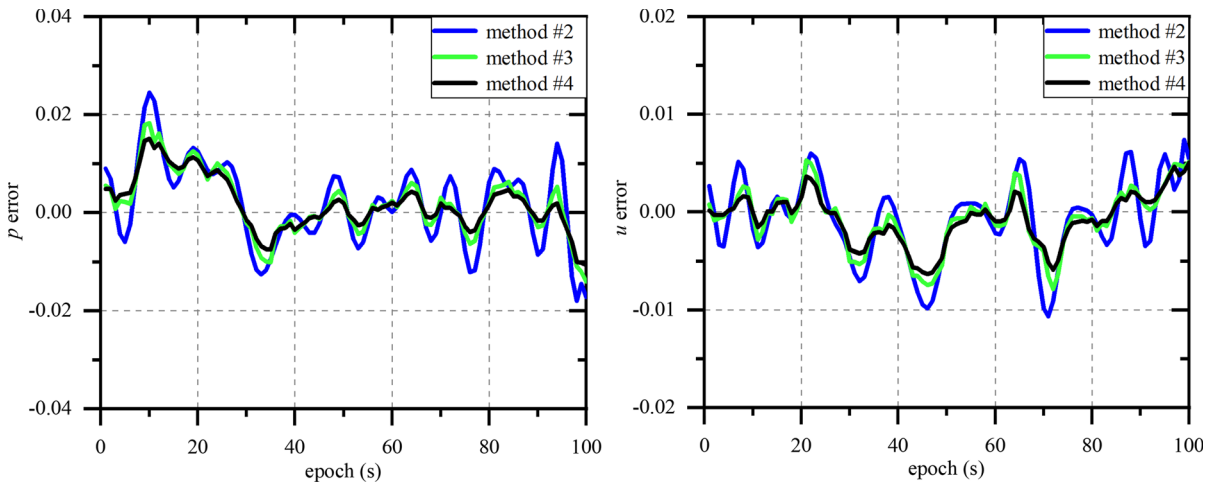

Fig. 3 Smoothing error of the Method #2, Method #3 and Method #4 for p component (Left) and u component (Right) at 1 s sampling instants in one MC run

4.2 Discussion

In this study, we focus on the smoothing of a continuous dynamical state space model with sampling system coefficients. The topic of state smoothing has long been studied in many diverse fields. Fruitful results can be found in the literatures; and the famous Rauch–Tung–Striebel (RTS) smoother is an excellent representative among others. These smoothers are called optimal. However, in this study it is made explicit that the optimality of these canonical approaches can hold only for the discrete state space model. In obtaining the discrete model from the original continuous model by numerical integration, numerical errors are often inevitable. Very little was found in the literatures on this question. An initial objective of the study was to develop an alternative state smoothing approach with potentially better performance by reducing discretization error. The approach is based on a straightforward kernel function parameterization of the state vector, thus deriving a regression problem with sparse regularization constrained. By solving the kernel weights, we can obtain state estimates at any instants (Fig. 4).

The results of the simulation test showed that the proposed smoothing approach could achieve improved accuracy compared with the conventional RTS smoother, and the accuracy of state estimates at arbitrary observation instants did not decrease significantly. Also, the average sparsity rates of parameter vectors are always higher than 70%. There are three possible explanations for these results. First, thanks to the elegant representation performance of Gaussian kernel, we can combine the pseudo-measurement model derived from the process model and the real measurement model and then establish a parametric regression model including all available information for state smoothing. This great representation property for the whole state space model also plays an important role in why we can obtain state estimates at any instants without significant accuracy degradation. Second, we partially attribute the outperformance of the proposed method to the reduced discretization errors. The densified sampling of system coefficients in the proposed method provides more measurements than RTS smoother, which reduces the discretization error to a certain extent. Third, the RTS smoother is always the optimal one in unbiased estimation; however, provided that the parameters are tuned reasonably, the regularization can reduce the variance greatly with the introduction of small bias, so that the improvement of overall accuracy (in the sense of MSE) can be possible.

Completely different from previous studies concerning smoothing for state space model, in which one usually directly estimates state vector, we utilize kernels to transforms state smoothing problem into a special kind of regression problem. Though it is very novel in the authors' experience, there are still some limitations of this work. On the one hand, there are actually some rigorous expressions of linear system discretization with regard to (1), but these expressions

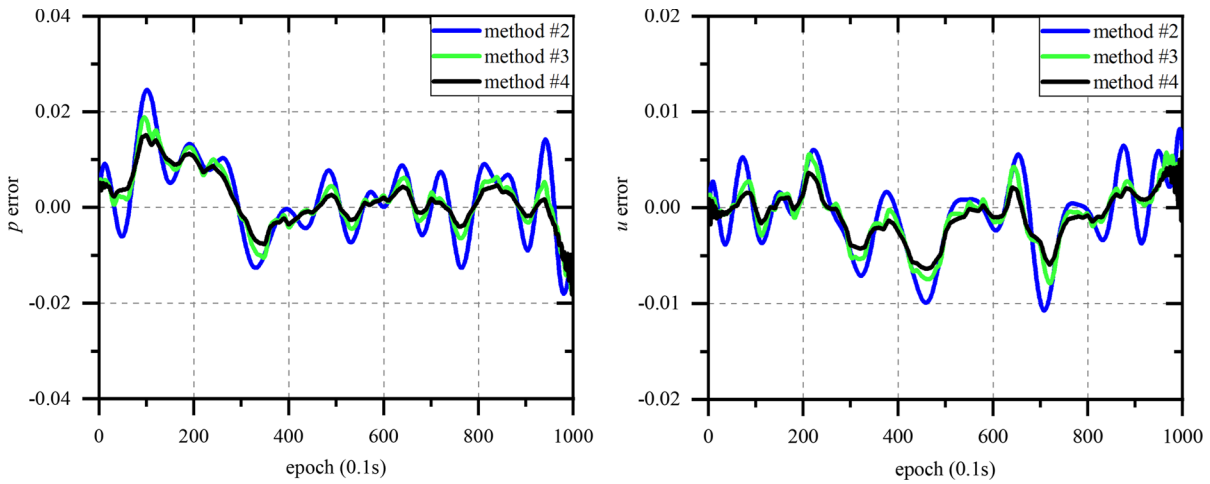

Fig. 4 Smoothing error of the Method #2, Method #3 and Method #4 for p component (Left) and u component (Right) with densified 10 Hz sampling rates in the MC run

usually bring significant troubles such as fussy series terms expression, complex computation load. The reason why we adopt the expansion as shown in (3) for comparison is that this is the most widely used expansion form. On the other hand, our conclusions are based on a simple two-dimensional tracking problem. In the future, we will implement more complex tests and verify the performance of the proposed method in practical engineering applications. In addition, it should be noted that the outperformance of Method #3 and Method #4 with regard to the Method #2 mainly results from the increasing sampling rates. This is in fact a merit of our method as well, since we can increase the sampling rate by simply adjusting the parameter p. Of course, this is at the expense of more computation. However, we can take a compromise method in which we do not have to put kernels at each sampling instants necessarily. As a result, the number of measurements increases, but the number of unknowns remains unchanged, so the involved computation is limited. Such a compromise method will be further considered in the future.

This study thus offers a new strategy for smoothing of continuous state space model. The preliminary results showed that the estimation accuracy is superior to that of the conventional RTS smoother. We would like to stress again that there are prices to pay for the improved performance of the proposed method, namely increased difficulties in model tuning and increased computational load. In future, computation efficiency as well as above limitations are the key points to be studied.

5 Concluding remarks

This paper proposed a new smoothing method for a continuous dynamical state space model with sampling system coefficients. The proposed method is quite different from the canonical ones, such as the RTS smoother. It does not directly estimate the state vector; rather, it first employs kernel models to represent the state vector and then estimate the kernel weights. In the proposed method, the state process model, namely the differential equations at the sampling instants of the involved system coefficients, is treated as measurement equations. By this treatment, discretization or numerical integration is avoided. The kernel weights, which constitute the parameter vector, are estimated by a special kind of regularization method, namely the L1 norm regularization also called the Lasso estimator. By this method, the solution can be made sparse, namely with part of the elements being exactly zero. The corresponding optimization problem in the Lasso estimation is solved by an efficient numerical algorithm called FISTA. The hyperparameters involved in the kernel model and the Lasso estimator, namely the kernel widths and the regularization coefficients, are selected objectively using the generalized cross-validation or corrected Akaike information criterion which is tailored to the

specific L1 norm regularization in the Lasso estimator. As a byproduct, the proposed method in fact provides an analytical model rather than state vector estimate series at discrete instants. A simple two-dimension example is employed in the simulation. The main purpose is to demonstrate the application of the proposed method. Improved accuracies of the proposed method are seen compared to the benchmark RTS smoother. It is again stressed that increased difficulties in model tuning and increased computational load would limit the application of this method, and this will be the focus of our future study. From an engineering viewpoint, we only mean to provide an alternative rather than replace the conventional approaches; and the easily tuned and computationally efficient RTS smoother should also be employed as long as its accuracy fulfills the mission requirements.

Acknowledgements The authors are grateful for the three anonymous reviewers for their constructive comments, which helped in improving the quality of this manuscript. The authors disclosed receipt of the following financial support for the research of this article: National Natural Science Foundation of China (Grant Number: 41774005, 41674008 and 41974026) and China Postdoctoral Science Foundation (Grant Number: 2019M652010 and 2019T120477). Their contributions are gratefully acknowledged.

Funding This work was funded by National Natural Science Foundation of China (Grant Number: 41774005, 41674008 and 41974026) and China Postdoctoral Science Foundation (Grant Number: 2019M652010 and 2019T120477).

Availability of data and material The simulation data used to support the findings of this study are included within the paper.

Code availability The test is based on MATLAB source code.

Compliance with ethical standards

Conflicts of interest The authors declare no conflict of interest.

References

1. Nastula, J., Chin, T.M., Gross, R., Sliwinska, J., Winska, M.: Smoothing and predicting celestial pole offsets using a Kalman filter and smoother. J. Geod. (2020). https://doi.org/10.1007/s00190-020-01349-9
2. Garcia, J., Besada, J.A., Molina, J.M., de Miguel, G.: Model-based trajectory reconstruction with IMM smoothing and segmentation. Inf. Fusion **22**, 127–140 (2015). https://doi.org/10.1016/j.inffus.2014.06.004
3. Hook, J.: Smoothing non-smooth systems with low-pass filters. Physica D **269**, 76–85 (2014). https://doi.org/10.1016/j.physd.2013.11.016
4. Simon, D.: Optimal state estimation: Kalman, H∞, and nonlinear approaches. Wiley, Newark (2006)
5. Särkkä, S.: Bayesian Filtering and Smoothing. Cambridge University Press, London (2013)
6. Grewal, M.S., Andrews, A.: Kalman Filtering Theory and Practice Using MATLAB, 4th edn. Wiley, New York (2015)
7. Chui, C.K., Chen, G.: Kalman Filtering: With real-Time Applications, 4th edn. Springer, Berlin (2009)
8. Zhang, X., Zhu, F., Tao, X., Duan, R.: New optimal smoothing scheme for improving relative and absolute accuracy of tightly coupled GNSS/SINS integration. GPS Solut. **21**(3), 861–872 (2017)
9. Xu, Y., Ahn, C.K., Shmaliy, Y.S., Chen, X., Bu, L.: Indoor INS/UWB-based human localization with missing data utilizing predictive UFIR filtering. IEEE-CAA J. Automat. Sin. **6**(4), 952–960 (2019). https://doi.org/10.1109/jas.2019.1911570
10. Lefferts, E.J., Markley, F.L., Shuster, M.D.: Kalman filtering for spacecraft attitude estimation. J. Guid. Control Dyn. **5**(5), 417–429 (1982)
11. Markley, F.L., Crassidis, J.L.: Fundamentals of Spacecraft Attitude Determination and Control. Springer, New York (2014)
12. Rauch, H.E., Tung, F., Strieble, C.T.: Maximum likelihood estimates of linear dynamic systems. AIAA J. **3**(8), 1445–1450 (1965)
13. Fraser, D., Potter, J.: The optimum linear smoother as a combination of two optimum linear filters. IEEE Trans. Autom. Control **14**(4), 387–390 (1969)
14. Simon, D., Shmaliy, Y.S.: Unified forms for Kalman and finite impulse response filtering and smoothing. Automatica **49**(6), 1892–1899 (2013)
15. Shmaliy, Y.S., Zhao, S., Ahn, C.K.: Unbiased FIR filtering: an iterative alternative to Kalman filtering ignoring noise and initial conditions. IEEE Control Syst. **37**(5), 70–89 (2017)
16. Aravkin, A., Burke, B., Ljung, L., Lozano, A., Pillonetto, G.: Generalized Kalman smoothing: modeling and algorithms. Automatica **86**, 63–86 (2017)
17. Chen, B., Liu, X., Zhao, H., Principe, J.C.: Maximum correntropy Kalman filter. Automatica **76**, 70–77 (2017)
18. Guo, J., Ou, J., Wang, H.: Robust estimation for correlated observations: two local sensitivity-based downweighting strategies. J. Geodesy **84**(4), 243–250 (2010)
19. Imani, M., Dougherty, E.R., Braga-Neto, U.: Boolean Kalman filter and smoother under model uncertainty. Automatica **111**, 108609 (2020)
20. Arasaratnam, I., Haykin, S.: Cubature kalman smoothers. Automatica **47**(10), 2245–2250 (2011)
21. Wang, X., Liang, Y., Pan, Q., Zhao, C., Yang, F.: Nonlinear Gaussian smoothers with colored measurement noise. IEEE Trans. Autom. Control **60**(3), 870–876 (2015)
22. Kulikova, M.V., Kulikov, G.Y.: NIRK-based accurate continuous-discrete extended Kalman filters for estimating continuous-time stochastic target tracking models. J. Comput. Appl. Math. **316**, 260–270 (2016)

23. Bell, B.M., Burke, J.V., Pillonetto, G.: An inequality constrained nonlinear Kalman–Bucy smoother by interior point likelihood maximization. Automatica **45**(1), 25–33 (2009)
24. Saitoh, S., Sawano, Y.: Theory of Reproducing Kernels and Applications. Springer, Singapore (2016)
25. Principe, J.C.: Information Theoretic Learning. Springer, Berlin (2010)
26. Girosi, F., Jones, M., Poggio, T.: Regularization theory and neural networks architectures. Neural Comput. **7**, 219–269 (1995)
27. Tikhonov, A.N., Arsenin, V.Y.: Solutions of Ill-posed Problems. Wiley, New York (1977)
28. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B (Methodological) **58**(1), 267–288 (1996)
29. Hastie, T., Tibshirani, R., Wainwright, M.: Statistical Learning with Sparsity: The Lasso and Generalizations. CRC Press, London (2015)
30. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. Siam J. Imaging Sci. **2**(1), 183–202 (2009). https://doi.org/10.1137/080716542
31. Craven, P., Wahba, G.: Smoothing noisy data with spline functions. Numer. Math. **31**(4), 377–403 (1979)
32. Golub, G.H., Heath, M.T., Wahba, G.: Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics **21**(2), 215–223 (1979)
33. Akaike, H.: A new look at the statistical model identification. IEEE Trans. Autom. Control **19**(6), 716–723 (1974)
34. Burnham, K.P., Anderson, D.R.: Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach. Springer, New York (2002)
35. Zou, H., Hastie, T., Tibshirani, R.: On the "degrees of freedom" of the lasso. Ann. Stat. **35**(5), 2173–2192 (2007)
36. Kulikov, G.Y., Kulikova, M.V.: Accurate continuous–discrete unscented Kalman filtering for estimation of nonlinear continuous-time stochastic models in radar tracking. Signal Process. **139**, 25–35 (2017)
37. Särkkä, S., Sarmavuori, J.: Gaussian filtering and smoothing for continuous-discrete dynamic systems. Signal Process. **93**(2), 500–510 (2013)
38. Arasaratnam, I., Ienkaran, H.S., Hurd, T.R.: Cubature Kalman filtering for continuous–discrete systems: theory and simulations. IEEE Trans. Signal Process. **58**(10), 4977–4993 (2010)
39. Grewal, M.S., Andrews, A.P., Bartone, C.G.: Global Navigation Satellite Systems, Inertial Navigation, and Integration. Wiley, New York (2013)
40. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2002)
41. Xu, Y., Ahn, C.K., Shmaliy, Y.S., Chen, X., Li, Y.: Adaptive robust INS/UWB-integrated human tracking using UFIR filter bank. Measurement **123**, 1–7 (2018). https://doi.org/10.1016/j.measurement.2018.03.043
42. Cui, B., Chen, X., Xu, Y., Huang, H., Liu, X.: Performance analysis of improved iterated cubature Kalman filter and its application to GNSS/INS. ISA Trans. **66**, 460–468 (2017). https://doi.org/10.1016/j.isatra.2016.09.010
43. Kay, S.M.: Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory, vol. 1. Pearson Education, New York (2013)
44. Stein, C.M.: Estimation of the mean of a multivariate normal distribution. Ann. Stat. **9**(6), 1135–1151 (1981)
45. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Stat. Comput. **14**(3), 199–222 (2004)
46. Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. J. Mach. Learn. Res. **1**, 211–244 (2001)
47. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course, vol. 87. Springe, New York (2004)
48. Parikh, N., Boyd, S.: Proximal algorithms. Found. Trends Optim. **1**(3), 127–239 (2014)
49. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**(1–4), 259–268 (1992)
50. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Rev. **43**(1), 129–159 (2001)
51. Donoho, D.L.: De-noising by soft-thresholding. IEEE Trans. Inf. Theory **41**(3), 613–627 (1995)
52. Donoho, D.L.: Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
53. Candès, E.J., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory **52**(2), 489–509 (2006)
54. Boyd, S., Parikh, N., Chu, E., Peleato, B.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2010)
55. Huang, Y., Zhang, Y., Wu, Z., Li, N., Chambers, J.: A novel adaptive kalman filter with inaccurate process and measurement noise covariance matrices. IEEE Trans. Autom. Control **63**(2), 594–601 (2018). https://doi.org/10.1109/tac.2017.2730480
56. Ardeshiri, T., Ozkan, E., Orguner, U., Gustafsson, F.: Approximate Bayesian smoothing with unknown process and measurement noise covariances. IEEE Signal Process. Lett. **22**(12), 2450–2454 (2015). https://doi.org/10.1109/lsp.2015.2490543