Autonomous Tracking and State Estimation With Generalized Group Lasso

Rui Gao, Simo Särkkä, *Senior Member, IEEE*, Rubén Claveria-Vega, and Simon Godsill

***Abstract*—We address the problem of autonomous tracking and state estimation for marine vessels, autonomous vehicles, and other dynamic signals under a (structured) sparsity assumption. The aim is to improve the tracking and estimation accuracy with respect to the classical Bayesian filters and smoothers. We formulate the estimation problem as a dynamic generalized group Lasso problem and develop a class of smoothing-and-splitting methods to solve it. The Levenberg–Marquardt iterated extended Kalman smoother-based multiblock alternating direction method of multipliers (LM-IEKS-mADMMs) algorithms are based on the alternating direction method of multipliers (ADMMs) framework. This leads to minimization subproblems with an inherent structure to which three new augmented recursive smoothers are applied. Our methods can deal with large-scale problems without preprocessing for dimensionality reduction. Moreover, the methods allow one to solve nonsmooth nonconvex optimization problems. We then prove that under mild conditions, the proposed methods converge to a stationary point of the optimization problem. By simulated and real-data experiments, including multisensor range measurement problems, marine vessel tracking, autonomous vehicle tracking, and audio signal restoration, we show the practical effectiveness of the proposed methods.**

***Index Terms*—Alternating direction method of multipliers (ADMMs), autonomous tracking, group Lasso, Kalman smoother (KS), sparsity, state estimation.**

I. Introduction

Autonomous tracking and state estimation problems are active research topics with many real-world applications, including intelligent maritime navigation, autonomous vehicle tracking, and audio signal estimation [1]–[5]. The aim is to autonomously estimate and track the state (e.g., position, velocity, or direction) of the dynamic system using imperfect measurements [6]. A frequently used approach for autonomous tracking and estimation problems is based on Bayesian filtering and smoothing. When the target dynamics and observation models are linear and Gaussian, the Kalman smoother (KS) [1], [7] provides the optimal Bayesian solution, which coincides with the optimal minimum mean square error estimator in that case. In the case of nonlinear dynamic systems, the iterated extended KS (IEKS) [8]–[10] makes use of local affine approximations by means of a Taylor series for the nonlinear functions, and then iteratively carries out KS. Sigma-point-based smoothing methods [11], [12] employ sigma-points to approximate the probability density of the states, which can preserve higher order accuracy than IEKS. Random sampling-based filters, such as particle filters [1], [13]–[15], can be used to deal with nonlinear tracking situations involving potentially arbitrary nonlinearities, noise, and constraints. Although these trackers and estimators are capable of utilizing the measurement information to obtain the estimates, they ignore sparsity dictated by the physical attributes of dynamic systems.

The motivation for our work comes from the following real-world applications. One significant application is *marine vessel tracking* [5], [6]. Vessels are frequently pitching and rolling on the surface of the ocean, which can be modeled as sparsity in the process noise. Our methodology is also applicable to *autonomous vehicle tracking*, which enables a vehicle to autonomously avoid obstacles and maintain safe distances to other vehicles. In the presence of many sudden stops (i.e., velocities are zero), the tracking accuracy can be improved by employing sparsity [16]. Other examples of tracked targets include robots [9] and unmanned aerial vehicles [17]. Another practical application is *audio signal restoration*, where, typically, only a few time–frequency elements are expected to be present and, thus, sparsity is an advisable assumption [18]. For example, the Gabor synthesis representation with sparsity constraints has proven to be suitable for audio restoration [19]. Similar problems can also be found in electrocardiogram (ECG) signal analysis [20] and automatic music transcription [21]. Hence, computationally effective sparsity modeling methods are in demand.

Since sparsity may improve the tracking and estimation performance, there is a growing amount of literature that proposes sparse regularizers, such as the Lasso (i.e., the least absolute shrinkage and selection operator or L_1-regularization) [22], [23] or total variation (TV) [24], [25] for these applications. The existing methods for sparse tracking and estimation can be split into two broad categories: 1) robust smoothing approaches and 2) optimization-based approaches. The former approaches merge filtering and smoothing with L_1-regularization. For instance, the

Manuscript received 22 July 2020; revised 20 January 2021; accepted 15 May 2021. Date of publication 24 June 2021; date of current version 17 October 2022. This work was supported by the Academy of Finland. This article was recommended by Associate Editor A. Katriniok. *(Corresponding author: Rui Gao.)*

Rui Gao and Simo Särkkä are with the Department of Electrical Engineering and Automation, Aalto University, 02150 Espoo, Finland (e-mail: rui.gao@aalto.fi; simo.sarkka@aalto.fi).

Rubén Claveria-Vega and Simon Godsill are with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, U.K. (e-mail: rmc83@cam.ac.uk; sjg@eng.cam.ac.uk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3085426.

Digital Object Identifier 10.1109/TCYB.2021.3085426

modified recursive filter-based compressive sensing methods were developed in [26]–[29]. An L_1-Laplace robust KS was presented in [30]. Using both sparsity and dynamics information, a sparse Bayesian learning framework was proposed in [31]. The latter approaches formulate the entire tracking and state estimation problem as an L_1-penalized minimization problem and then apply iterative algorithms to solve the minimization problem [16], [32]–[36]. While these L_1-penalized estimators offer several benefits, they penalize individual elements of the state vector or process noise instead of groups of elements in them.

Recently, there have been important advances in the structured sparsity methodology for collision avoidance [37], [38]; path-following and tracking [39], [40]; and visual tracking [41], [42]. In [43], a discriminative supervised hashing method was proposed for object tracking tasks. An adaptive elastic echo state network was developed for multivariate time-series prediction in [44]. The work in [45] formulated a moving tracking problem with L_2-norm constraints and then introduced a temporal consistency dictionary learning algorithm. However, the methods lack strong convergence and performance guarantees, particularly when the objective becomes nonconvex. Moreover, relatively, few methods exist for incorporating structured sparsity into autonomous tracking and state estimation problems. Taking these developments into consideration, the main goal here is to develop new efficient methods for regularized autonomous tracking and estimation problems, which allow for group Lasso type of sparseness assumptions on groups of state or process noise elements.

When we formulate a regularized autonomous tracking and state estimation problem as a generalized L_2-minimization problem (also called the dynamic generalized group Lasso problem), the resulting problem is difficult to solve due to its nonsmoothness and/or nonconvexity. Splitting-based optimization methods [46]–[48], such as multiblock alternating direction method of multipliers (ADMMs) [47], are methods that can tackle this kind of problem. One advantage of these methods is that they decompose the original problem into a sequence of easier subproblems. Although these methods can directly work on the original optimization problem, such direct use ignores the inherent structure induced by the implied Markovian structure in the optimization problem. In this article, we propose a class of efficient smoothing-and-splitting methods that outperform the classical optimization methods in terms of computational time due to the leveraging of the Markovian structure.

In this article, we focus on autonomous tracking and state estimation problems with sparsity-inducing priors. Our first contribution is to provide a flexible formulation of the dynamic generalized group Lasso problems arising in autonomous tracking and state estimation. Special cases of the formulation are Lasso, isotropic TV, anisotropic TV, fused Lasso, group Lasso, and sparse group Lasso. Meanwhile, the formulation can cope with sparsity on the process noise or the state in dynamic systems. Since the resulting optimization problems are nonsmooth, possibly nonconvex, and large-dimensional, our second contribution is to provide a class of the smoothing-and-splitting methods to address them. We develop the new KS-mADMM, Gauss–Newton IEKS-mADMM (GN-IEKS-mADMM), and Levenberg–Marquardt IEKS-mADMM (LM-IEKS-mADMM) methods, which use augmented recursive smoothers to solve the primal subproblems in the mADMM iterations. As a third contribution, we prove that under mild conditions, the proposed methods converge to a stationary point. Our fourth contribution is to apply the proposed methods to real-world applications of marine vessel tracking, autonomous vehicle tracking, and audio signal restoration.

The remainder of this article is structured as follows. In Section II, we formulate the sparse autonomous tracking and state estimation problem as a generalized L_2-minimization problem. Particularly, we present a broad class of regulariser configurations parameterized by sets of matrices and vectors. We introduce the batch tracking and estimation methods in Section III and present three augmented recursive smoothing methods in Section IV. In Section V, we establish the convergence. In Section VI, we report numerical results on simulated and real-life datasets. Section VII draws the concluding remarks.

The notation is as follows. Matrices $\mathbf{X}$ and vectors $\mathbf{x}$ are indicated in boldface. $(\cdot)^\top$ represents the transposition and $(\cdot)^{-1}$ represents the matrix inversion. The $\mathbf{R}$-weighted norm of $\mathbf{x}$ is denoted by $\|\mathbf{x}\|_{\mathbf{R}} = \sqrt{\mathbf{x}^\top \mathbf{R}\mathbf{x}}$. $\|\mathbf{x}\|_1 = \sum |x_i|$ denotes the L_1-norm and $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$ denotes the L_2-norm. $\mathbf{X}_{g,t}$ is the (g,t):th element of matrix $\mathbf{X}$, and $\mathbf{x}^{(k)}$ denotes the value of $\mathbf{x}$ at the k:th iteration. $\mathrm{vec}(\cdot)$ represents a vectorization operator, $\mathrm{diag}(\cdot)$ represents a block-diagonal matrix operator with the elements in its argument on the diagonal, and $\mathbf{x}_{1:T} = \mathrm{vec}(\mathbf{x}_1, \ldots, \mathbf{x}_T)$. $\partial\phi(\mathbf{x})$ denotes a subgradient of ϕ. $\mathbf{J}_\phi$ is the Jacobian of $\phi(\mathbf{x})$. $\delta_+(\mathbf{A})$ denotes the smallest eigenvalue of $\mathbf{A}$. $p(\mathbf{x})$ denotes the probability density function (pdf) of $\mathbf{x}$ and $\mathcal{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P})$ denotes a Gaussian pdf with mean $\mathbf{m}$ and covariance $\mathbf{P}$ evaluated at $\mathbf{x}$.

II. Problem Statement

Let $\mathbf{y}_t \in \mathbb{R}^{N_y}$ be a measurement of a dynamic system and $\mathbf{x}_t \in \mathbb{R}^{N_x}$ be an unknown state (sometimes called the source or signal). The state and measurement are related according to a dynamic state-space model of the form

$$\mathbf{x}_t = \mathbf{a}_t(\mathbf{x}_{t-1}) + \mathbf{q}_t, \quad \mathbf{y}_t = \mathbf{h}_t(\mathbf{x}_t) + \mathbf{r}_t \tag{1}$$

where $\mathbf{h}_t : \mathbb{R}^{N_x} \to \mathbb{R}^{N_y}$ and $\mathbf{a}_t : \mathbb{R}^{N_x} \to \mathbb{R}^{N_x}$ are the measurement and state transition functions, respectively, and $t = 1, \ldots, T$ is the time step number. The process and measurement noises $\mathbf{q}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$ and $\mathbf{r}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$ are assumed to be zero-mean Gaussian with covariances $\mathbf{Q}_t$ and $\mathbf{R}_t$, respectively. The initial condition at $t = 1$ is given by $\mathbf{x}_1 \sim \mathcal{N}(\mathbf{m}_1, \mathbf{P}_1)$. A particular special case of (1) is an affine Gaussian model by

$$\mathbf{a}_t(\mathbf{x}_{t-1}) = \mathbf{A}_t\,\mathbf{x}_{t-1} + \mathbf{b}_t, \quad \mathbf{h}_t(\mathbf{x}_t) = \mathbf{H}_t\,\mathbf{x}_t + \mathbf{e}_t \tag{2}$$

where $\mathbf{A}_t \in \mathbb{R}^{N_x \times N_x}$ and $\mathbf{H}_t \in \mathbb{R}^{N_y \times N_x}$ are the transition and measurement matrices, and $\mathbf{b}_t$ and $\mathbf{e}_t$ are bias terms.

The goal here is to obtain the "best estimate" of $\mathbf{x}_{1:T}$ from imperfect measurements $\mathbf{y}_{1:T}$. For computing $\mathbf{x}_{1:T}$ with

TABLE I
EXAMPLES OF SPARSITY-PROMOTING REGULARISERS THAT ARE INCLUDED IN THE PRESENT FRAMEWORK

Regularisation	$\mathbf{G}_{g,t}$ **descriptions**
L_2-regularisation	$\mathbf{G}_{g,t}$ is an identity matrix
Lasso	$N_g = N_x$, $P_g = 1$ for all g, $\mathbf{G}_{g,t}$ has 1 at g:th column and zeros otherwise.
Isotopic TV	$N_g = 1$, $P_1 = N_x - 1$ $\mathbf{G}_{1,t}$ encodes a finite difference operator.
Anisotopic TV	$\mathbf{G}_{g,t}$ encodes the g:th row of a finite difference operator.
Fused Lasso	$g = 1, \ldots, N_x$, $P_g = 1$ for all g, $\mathbf{G}_{g,t}$ has 1 at g:th column and zeros otherwise; $g = N_x + 1, \ldots, N_g$, $\mathbf{G}_{g,t}$ encodes the g:th row of a finite difference operator.
Group Lasso	$\mathbf{G}_{g,t}$ has 1, corresponding to the selected elements of $\mathbf{x}_t$ in the group and zeros otherwise.
Sparse group Lasso	$g = 1, \ldots, N_x$, $P_g = 1$, $\mathbf{G}_{g,t}$ has 1 at g:th column and zeros otherwise; $g = N_x + 1, \ldots, N_g$, $\mathbf{G}_{g,t}$ has the same setting with group Lasso.

TABLE II
FLEXIBLE SPARSITY ASSUMPTIONS BY SELECTING $\mathbf{B}_t$ AND $\mathbf{d}_t$

Dynamic systems	Sparsity on:	$\mathbf{B}_t$ and $\mathbf{d}_t$
Affine Gaussian	state	Settings on $\mathbf{B}_t = \mathbf{0}$, $\mathbf{d}_t = \mathbf{0}$
	process noise	$\mathbf{B}_t = \mathbf{A}_t$, $\mathbf{d}_t = \mathbf{b}_t$
Nonlinear	state	$\mathbf{B}_t = \mathbf{0}$, $\mathbf{d}_t = \mathbf{0}$
	process noise	$\mathbf{B}_t\mathbf{x}_{t-1} + \mathbf{d}_t$ as the affine approximation of $\mathbf{a}_t(\mathbf{x}_{t-1})$ (see Section IV-B)

sparsity-inducing priors, we define a set of matrices $\{\mathbf{G}_{g,t} \in \mathbb{R}^{P_g \times N_x} \mid g = 1, \ldots, N_g\}$, matrices $\mathbf{B}_t$, and vectors $\mathbf{d}_t$, for $t = 1, \ldots, T$, and impose sparsity on the groups of elements of the state or the process noise. Mathematically, the problem of computing the state estimate $\mathbf{x}^\star_{1:T}$ is formulated as

$$\begin{aligned}\mathbf{x}^\star_{1:T} = \arg\min_{\mathbf{x}_{1:T}} &\frac{1}{2}\sum_{t=1}^{T}\|\mathbf{y}_t - \mathbf{h}_t(\mathbf{x}_t)\|^2_{\mathbf{R}_t^{-1}} \\ &+ \frac{1}{2}\sum_{t=2}^{T}\|\mathbf{x}_t - \mathbf{a}_t(\mathbf{x}_{t-1})\|^2_{\mathbf{Q}_t^{-1}} + \frac{1}{2}\|\mathbf{x}_1 - \mathbf{m}_1\|^2_{\mathbf{P}_1^{-1}} \\ &+ \sum_{t=1}^{T}\sum_{g=1}^{N_g}\mu\left\|\mathbf{G}_{g,t}(\mathbf{x}_t - \mathbf{B}_t\,\mathbf{x}_{t-1} - \mathbf{d}_t)\right\|_2 \end{aligned} \tag{3}$$

where $\mu > 0$ is a penalty parameter.

A merit of our formulation is its flexibility, because the selections of $\mathbf{G}_{g,t}$, $\mathbf{B}_t$, and $\mathbf{d}_t$ can be adjusted to represent different regularisers. With matrix $\mathbf{G}_{g,t}$, the formulation (3) accommodates a large class of sparsity-promoting regularisers (e.g., Lasso, isotopic TV, anisotopic TV, fused Lasso, group Lasso, and sparse group Lasso). A list of such regularisers is reported in Table I. Meanwhile, the formulation (3) also allows for putting sparsity assumptions on the state or the process noise by different selections of $\mathbf{B}_t$ and $\mathbf{d}_t$ (see Table II).

A simple, yet illustrative, example can be found in autonomous vehicle tracking. When there are stop-and-go points (e.g., vehicle stops) in the data, the zero-velocity and zero-angle values at those time points can be grouped together via the L_2-norm and $\mathbf{G}_{g,t}$. That means three elements can be forced to be equal to 0 at the same time. Another application is in audio restoration, where the matrices $\mathbf{G}_{g,t}$ are defined so that only two elements of the state $\mathbf{x}_t$—corresponding to the real and imaginary parts of a synthesis coefficient—are extracted at a time step. Thus, these pairs, which are associated with the same time–frequency basis functions, tend to be nonzero or 0 together.

Problem (3) is more difficult to solve than the common L_2-minimization problem (which corresponds to $\mathbf{G}_{g,t} = \mathbf{I}$, where $\mathbf{I}$ is an identity matrix) or the squared L_2-minimization problem (the problem with $\|\mathbf{G}_{g,t}(\cdot)\|_2^2$), since the penalty term $\|\mathbf{G}_{g,t}(\cdot)\|_2$ is nonsmooth. Furthermore, $\mathbf{G}_{g,t}$ is possibly rank-deficient matrix. In this article, we first derive batch tracking and estimation methods, which are based on the batch computation of the state sequence. To speed up the batch methods, we then propose augmented recursive smoother methods for the primal variable update.

III. BATCH TRACKING AND ESTIMATION METHODS

In this section, we introduce the multiblock ADMM (mADMM) framework. Based on this framework, we derive batch algorithms for solving the regularized tracking and state estimation problem.

A. General Multiblock ADMM Framework

The methods that we develop are based on the mADMM [47]. The mADMM provides an algorithmic framework that is applicable to problems of the form (3), and it can be instantiated by defining the auxiliary variables and their update steps. We introduce auxiliary variables $\mathbf{v}_t$ and $\mathbf{w}_{g,t}$, $g = 1, \ldots, N_g$, $t = 1, \ldots, T$, and then build the following constraints:

$$\begin{aligned}\mathbf{x}_t - \mathbf{B}_t\,\mathbf{x}_{t-1} - \mathbf{d}_t &= \mathbf{v}_t \\ \mathbf{w}_{1,t} &= \mathbf{G}_{1,t}\,\mathbf{v}_t \\ &\vdots \\ \mathbf{w}_{N_g,t} &= \mathbf{G}_{N_g,t}\,\mathbf{v}_t. \end{aligned} \tag{4}$$

Note that in (4), we could alternatively introduce auxiliary variables $\mathbf{w}_{g,t} = \mathbf{G}_{g,t}(\mathbf{x}_t - \mathbf{B}_t\,\mathbf{x}_{t-1} - \mathbf{d}_t)$, but this replacement would require $\mathbf{G}_{g,t}$ to be invertible when using the augmented recursive smoothers later on. To avoid such restrictions, we employ variables $\mathbf{v}_t$ and $\mathbf{w}_{g,t}$ to build the more general constraints in this article.

For simplicity of notation, we denote $\mathbf{w}_t = \begin{bmatrix}\mathbf{w}_{1,t}^\top, & \ldots, & \mathbf{w}_{N_g,t}^\top\end{bmatrix}^\top$ and $\mathbf{G}_t = \begin{bmatrix}\mathbf{G}_{1,t}^\top, & \ldots, & \mathbf{G}_{N_g,t}^\top\end{bmatrix}^\top$, and then solve (3), using an equivalent constrained optimization problem

$$\begin{aligned}\min_{\substack{\mathbf{x}_{1:T}, \mathbf{w}_{1:T},\\ \mathbf{v}_{1:T}}} &\frac{1}{2}\sum_{t=1}^{T}\|\mathbf{y}_t - \mathbf{h}_t(\mathbf{x}_t)\|^2_{\mathbf{R}_t^{-1}} + \sum_{t=1}^{T}\sum_{g=1}^{N_g}\mu\left\|\mathbf{w}_{g,t}\right\|_2 \\ &+ \frac{1}{2}\sum_{t=2}^{T}\|\mathbf{x}_t - \mathbf{a}_t(\mathbf{x}_{t-1})\|^2_{\mathbf{Q}_t^{-1}} + \frac{1}{2}\|\mathbf{x}_1 - \mathbf{m}_1\|^2_{\mathbf{P}_1^{-1}} \\ \text{s.t.} &\begin{bmatrix}\mathbf{x}_t - \mathbf{B}_t\,\mathbf{x}_{t-1} - \mathbf{d}_t \\ \mathbf{w}_t\end{bmatrix} = \begin{bmatrix}\mathbf{I} \\ \mathbf{G}_t\end{bmatrix}\mathbf{v}_t, \quad t = 1, \ldots, T. \end{aligned} \tag{5}$$

The variables $\mathbf{x}_{1:T}$, $\mathbf{w}_{1:T}$, and $\mathbf{v}_{1:T}$ can be handled by defining the augmented Lagrangian function

$$\begin{aligned}\mathcal{L}_\gamma\left(\mathbf{x}_{1:T}, \mathbf{w}_{1:T}, \mathbf{v}_{1:T}; \boldsymbol{\eta}_{1:T}\right) &\triangleq \frac{1}{2}\sum_{t=1}^{T}\|\mathbf{y}_t - \mathbf{h}_t(\mathbf{x}_t)\|^2_{\mathbf{R}_t^{-1}} \\ &+ \frac{1}{2}\sum_{t=2}^{T}\|\mathbf{x}_t - \mathbf{a}_t(\mathbf{x}_{t-1})\|^2_{\mathbf{Q}_t^{-1}} + \frac{1}{2}\|\mathbf{x}_1 - \mathbf{m}_1\|^2_{\mathbf{P}_1^{-1}} \\ &+ \sum_{t=1}^{T}\sum_{g=1}^{N_g}\mu\left\|\mathbf{w}_{g,t}\right\|_2 + \sum_{t=1}^{T}\boldsymbol{\eta}_t^\top\left(\begin{bmatrix}\mathbf{u}_t\\ \mathbf{w}_t\end{bmatrix} - \begin{bmatrix}\mathbf{I}\\ \mathbf{G}_t\end{bmatrix}\mathbf{v}_t\right) \\ &+ \sum_{t=1}^{T}\frac{\gamma}{2}\left\|\begin{bmatrix}\mathbf{u}_t\\ \mathbf{w}_t\end{bmatrix} - \begin{bmatrix}\mathbf{I}\\ \mathbf{G}_t\end{bmatrix}\mathbf{v}_t\right\|_2^2 \end{aligned} \tag{6}$$

where $\mathbf{u}_t = \mathbf{x}_t - \mathbf{B}_t\mathbf{x}_{t-1} - \mathbf{d}_t$, $\boldsymbol{\eta}_t \in \mathbb{R}^{(N_x + P_g\times N_g)}$ is a Lagrangian multiplier and $\gamma > 0$ is a penalty parameter.

The mADMM framework minimizes the function $\mathcal{L}_\gamma$ by alternating the $\mathbf{x}_{1:T}$-minimization step, the $\mathbf{w}_{1:T}$-minimization step, the $\mathbf{v}_{1:T}$-minimization step, and the dual variable $\boldsymbol{\eta}_{1:T}$ update step. Given $(\mathbf{x}_{1:T}^{(k)}, \mathbf{w}_{1:T}^{(k)}, \mathbf{v}_{1:T}^{(k)}, \boldsymbol{\eta}_{1:T}^{(k)})$, the iteration of mADMM has the following steps:

$$\begin{aligned}\mathbf{x}_{1:T}^{(k+1)} &= \arg\min_{\mathbf{x}_{1:T}}\sum_{t=1}^{T}\frac{1}{2}\|\mathbf{y}_t - \mathbf{h}_t(\mathbf{x}_t)\|^2_{\mathbf{R}_t^{-1}} + \frac{1}{2}\|\mathbf{x}_1 - \mathbf{m}_1\|^2_{\mathbf{P}_1^{-1}} \\ &+ \frac{1}{2}\sum_{t=2}^{T}\|\mathbf{x}_t - \mathbf{a}_t(\mathbf{x}_{t-1})\|^2_{\mathbf{Q}_t^{-1}} \\ &+ \frac{\gamma}{2}\sum_{t=1}^{T}\left\|\mathbf{u}_t - \mathbf{v}_t^{(k)} + \frac{\overline{\boldsymbol{\eta}}_t^{(k)}}{\gamma}\right\|_2^2 \end{aligned} \tag{7a}$$

$$\mathbf{w}_t^{(k+1)} = \arg\min_{\mathbf{w}_t}\sum_{g=1}^{N_g}\mu\left\|\mathbf{w}_{g,t}\right\|_2 + \frac{\gamma}{2}\left\|\mathbf{w}_t - \mathbf{G}_t\mathbf{v}_t^{(k)} + \frac{\underline{\boldsymbol{\eta}}_t^{(k)}}{\gamma}\right\|_2^2 \tag{7b}$$

$$\mathbf{v}_t^{(k+1)} = \arg\min_{\mathbf{v}_t}\frac{\gamma}{2}\left\|\begin{bmatrix}\mathbf{u}_t^{(k+1)}\\ \mathbf{w}_t^{(k+1)}\end{bmatrix} - \begin{bmatrix}\mathbf{I}\\ \mathbf{G}_t\end{bmatrix}\mathbf{v}_t + \frac{\boldsymbol{\eta}_t^{(k)}}{\gamma}\right\|_2^2 \tag{7c}$$

$$\boldsymbol{\eta}_t^{(k+1)} = \boldsymbol{\eta}_t^{(k)} + \gamma\left(\begin{bmatrix}\mathbf{u}_t^{(k+1)}\\ \mathbf{w}_t^{(k+1)}\end{bmatrix} - \begin{bmatrix}\mathbf{I}\\ \mathbf{G}_t\end{bmatrix}\mathbf{v}_t^{(k+1)}\right) \tag{7d}$$

where $\boldsymbol{\eta}_t = \mathrm{vec}(\overline{\boldsymbol{\eta}}_t, \underline{\boldsymbol{\eta}}_{1,t}, \ldots, \underline{\boldsymbol{\eta}}_{N_g,t})$. We solve $\mathbf{w}_t$, $\mathbf{v}_t$, and $\boldsymbol{\eta}_t$ subproblems for each t, respectively. The $\mathbf{w}_t$-subproblem and $\mathbf{v}_t$-subproblem have the solutions

$$\mathbf{w}_t^{(k+1)} = \mathcal{S}_{\mu/\gamma}\left(\mathbf{G}_{g,t}\mathbf{v}_t^{(k)} - \underline{\boldsymbol{\eta}}_{g,t}^{(k)}/\gamma\right) \tag{8a}$$

$$\mathbf{v}_t^{(k+1)} = \frac{1}{\gamma}\left(\mathbf{I} + \mathbf{G}_t^\top\mathbf{G}_t\right)^{-1}\left(\begin{bmatrix}\mathbf{I}\\ \mathbf{G}_t\end{bmatrix}^\top\left(\gamma\begin{bmatrix}\mathbf{u}_t^{(k+1)}\\ \mathbf{w}_t^{(k+1)}\end{bmatrix} + \boldsymbol{\eta}_t^{(k)}\right)\right) \tag{8b}$$

where $\mathcal{S}_{\mu/\gamma}(\cdot)$ is the shrinkage operator [49].

Given the mADMM framework, the solutions in (8a), (8b), and (7d) are the basic steps of our methods. In a single iteration, the $\mathbf{w}_t$-update can be computed in $\mathcal{O}(N_g)$ operations, and each $\mathbf{v}_t$-update takes $\mathcal{O}(N_x^3)$. However, when the $\mathbf{x}_{1:T}$-subproblem is solved by the batch estimation methods, it typically takes $\mathcal{O}(N_x^3T^3)$ operations. Thus, the main computational demand is in updating $\mathbf{x}_{1:T}$. Our main goal here is to derive efficient methods for the $\mathbf{x}_{1:T}$-minimization step. Before that, we first develop batch methods to solve the $\mathbf{x}_{1:T}$-subproblem.

B. Batch Solution for Affine Systems

The first batch method we explore is for the affine Gaussian systems. We first stack all the state variables into single variables and then rewrite the $\mathbf{x}_{1:T}$-subproblem (7a) in the form

$$\begin{aligned}\mathbf{x}^\star = \arg\min_{\mathbf{x}}\frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x} - \mathbf{e}\|^2_{\mathbf{R}^{-1}} + \frac{1}{2}\|\mathbf{m} - \mathbf{A}\mathbf{x} - \mathbf{b}\|^2_{\mathbf{Q}^{-1}} \\ + \frac{\gamma}{2}\left\|\boldsymbol{\Phi}\mathbf{x} - \mathbf{d} - \mathbf{v}^{(k)} + \overline{\boldsymbol{\eta}}^{(k)}/\gamma\right\|_2^2\end{aligned} \tag{9}$$

where we have set

$$\mathbf{x} = \mathbf{x}_{1:T}, \quad \boldsymbol{\Phi} = \begin{pmatrix}\mathbf{I} & \mathbf{0} & & \\ -\mathbf{B}_2 & \mathbf{I} & \ddots & \\ & \ddots & \ddots & \mathbf{0} \\ & & -\mathbf{B}_T & \mathbf{I}\end{pmatrix}. \tag{10}$$

The other variables $\mathbf{y}$, $\mathbf{e}$, $\mathbf{m}$, $\mathbf{d}$, $\mathbf{e}$, $\mathbf{v}$, $\overline{\boldsymbol{\eta}}$, $\mathbf{H}$, $\mathbf{R}$, $\mathbf{Q}$, and $\mathbf{A}$ are defined analogously to [16, eq. (17)]. By setting the derivative to 0, the solution is

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \left(\mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H} + \mathbf{A}^\top\mathbf{Q}^{-1}\mathbf{A} + \gamma\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\right)^{-1} \\ &\times\left(\mathbf{H}^\top\mathbf{R}^{-1}(\mathbf{y} - \mathbf{e}) + \mathbf{A}^\top\mathbf{Q}^{-1}(\mathbf{m} - \mathbf{b})\right. \\ &\left. + \gamma\boldsymbol{\Phi}^\top\left(\mathbf{d} + \mathbf{v}^{(k)} - \overline{\boldsymbol{\eta}}^{(k)}/\gamma\right)\right).\end{aligned} \tag{11}$$

In other words, computing the $\mathbf{x}$-minimization amounts to solving a linear system with the coefficient matrix $\mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H} + \mathbf{A}^\top\mathbf{Q}^{-1}\mathbf{A} + \gamma\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$. When the matrix inverse exists, the $\mathbf{x}$-subproblem has a unique solution. In addition, with a sparsity assumption on the states $\mathbf{x}_t$, $\boldsymbol{\Phi}$ is an identity matrix, and $\mathbf{d}$ is a zero vector. When the noise $\mathbf{q}_t$ is sparse, we can set

$$\boldsymbol{\Phi} = \mathbf{A}, \quad \mathbf{d} = \mathbf{m} - \mathbf{b} \tag{12}$$

which corresponds to the setting of $\mathbf{B}_t$ and $\mathbf{d}_t$ according to Table II.

The disadvantage of the batch solution is that it requires an extensive amount of computations when T is large. For this reason, in Section IV-A, we propose to use an augmented recursive smoother, which is mathematically equivalent to the batch method, to improve the computational performance.

C. Gauss–Newton for Nonlinear Systems

When the system is nonlinear, we use a similar batch notation as in the affine case and, in addition, define the nonlinear functions

$$\begin{aligned}\mathbf{a}(\mathbf{x}) &= \mathrm{vec}(\mathbf{x}_1, \mathbf{x}_2 - \mathbf{a}_2(\mathbf{x}_1), \ldots, \mathbf{x}_T - \mathbf{a}_T(\mathbf{x}_{T-1})) \\ \mathbf{h}(\mathbf{x}) &= \mathrm{vec}(\mathbf{h}_1(\mathbf{x}_1), \ldots, \mathbf{h}_T(\mathbf{x}_T)).\end{aligned} \tag{13}$$

The primal $\mathbf{x}_{1:T}$-subproblem then has the form

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}}\theta(\mathbf{x}) \tag{14}$$

where

$$\theta(\mathbf{x}) = \frac{1}{2}\|\mathbf{y}-\mathbf{h}(\mathbf{x})\|^2_{\mathbf{R}^{-1}} + \frac{1}{2}\|\mathbf{m}-\mathbf{a}(\mathbf{x})\|^2_{\mathbf{Q}^{-1}} + \frac{\gamma}{2}\left\|\mathbf{\Phi}\mathbf{x}-\mathbf{d}-\mathbf{v}^{(k)}+\overline{\boldsymbol{\eta}}^{(k)}/\gamma\right\|_2^2. \tag{15}$$

The function $\theta(\mathbf{x})$ can now be minimized by the Gauss–Newton (GN) method [46]. In GN, we first linearize the nonlinear functions $\mathbf{a}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$ and then replace them in $\theta(\mathbf{x})$ by the linear (or actually affine) approximations. The GN iteration then becomes

$$\begin{aligned}\mathbf{x}^{(k,i+1)} = &\left(\mathbf{J}_\theta^\top\mathbf{J}_\theta\left(\mathbf{x}^{(k,i)}\right)\right)^{-1}\\ &\times\Big[\mathbf{J}_h^\top\left(\mathbf{x}^{(k,i)}\right)\mathbf{R}^{-1}\left(\mathbf{y}-\mathbf{h}\left(\mathbf{x}^{(k,i)}\right)+\mathbf{J}_h\left(\mathbf{x}^{(k,i)}\right)\mathbf{x}^{(k,i)}\right)\\ &+\mathbf{J}_a^\top\left(\mathbf{x}^{(k,i)}\right)\mathbf{Q}^{-1}\\ &\times\left(\mathbf{m}-\mathbf{a}\left(\mathbf{x}^{(k,i)}\right)+\mathbf{J}_a\left(\mathbf{x}^{(k,i)}\right)\mathbf{x}^{(k,i)}\right)\\ &+\gamma\mathbf{\Phi}^\top\left(\mathbf{d}+\mathbf{v}^{(k)}-\overline{\boldsymbol{\eta}}^{(k)}/\gamma\right)\Big]\end{aligned} \tag{16}$$

where

$$\mathbf{J}_\theta^\top\mathbf{J}_\theta(\mathbf{x}) = \mathbf{J}_h^\top(\mathbf{x})\mathbf{R}^{-1}\mathbf{J}_h(\mathbf{x}) + \mathbf{J}_a^\top(\mathbf{x})\mathbf{Q}^{-1}\mathbf{J}_a(\mathbf{x}) + \gamma\mathbf{\Phi}^\top\mathbf{\Phi}.$$

The above computations are carried out iteratively until a maximum number of iterations $I_{\max}$ is reached. We take the solution $\mathbf{x}^{(k,I_{\max})}$ as the next iterate $\mathbf{x}^{(k+1)}$. While GN avoids the trouble of computing the Hessians of the model functions, it has problems when the Jacobians are rank-deficient. The Levenberg–Marquardt (LM) method is introduced next to address this problem.

D. Levenberg–Marquardt Method

The LM method [50], also called the regularized or damped GN method, improves the performance of GN by using an additional regularization term. With damping factors $\lambda^{(i)} > 0$ and a sequence of positive-definite regularization matrices $\mathbf{S}^{(i)}$, function $\theta(\mathbf{x})$ can be approximated by

$$\begin{aligned}\theta(\mathbf{x}) \approx &\frac{1}{2}\left\|\mathbf{y}-\mathbf{h}\left(\mathbf{x}^{(i)}\right)+\mathbf{J}_h\left(\mathbf{x}^{(i)}\right)\left(\mathbf{x}-\mathbf{x}^{(i)}\right)\right\|^2_{\mathbf{R}^{-1}}\\ &+\frac{1}{2}\left\|\mathbf{m}-\mathbf{a}\left(\mathbf{x}^{(i)}\right)+\mathbf{J}_a\left(\mathbf{x}^{(i)}\right)\left(\mathbf{x}-\mathbf{x}^{(i)}\right)\right\|^2_{\mathbf{Q}^{-1}}\\ &+\frac{\gamma}{2}\left\|\mathbf{\Phi}\mathbf{x}-\mathbf{d}-\mathbf{v}^{(k)}+\overline{\boldsymbol{\eta}}^{(k)}/\gamma\right\|_2^2\\ &+\frac{\lambda^{(i)}}{2}\left\|\mathbf{x}-\mathbf{x}^{(i)}\right\|^2_{[\mathbf{S}^{(i)}]^{-1}}.\end{aligned} \tag{17}$$

Using the minimum of this approximate cost function at each step i as the next iterate, we obtain the following iteration:

$$\begin{aligned}\mathbf{x}^{(k,i+1)} = &\left(\mathbf{J}_\theta^\top\mathbf{J}_\theta\left(\mathbf{x}^{(k,i)}\right)+\lambda^{(i)}\left[\mathbf{S}^{(i)}\right]^{-1}\right)^{-1}\\ &\Big[\mathbf{J}_h^\top\left(\mathbf{x}^{(k,i)}\right)\mathbf{R}^{-1}\left(\mathbf{y}-\mathbf{h}\left(\mathbf{x}^{(k,i)}\right)+\mathbf{J}_h\left(\mathbf{x}^{(k,i)}\right)\mathbf{x}^{(k,i)}\right)\\ &+\mathbf{J}_a^\top\left(\mathbf{x}^{(k,i)}\right)\mathbf{Q}^{-1}\\ &\times\left(\mathbf{m}-\mathbf{a}\left(\mathbf{x}^{(k,i)}\right)+\mathbf{J}_a\left(\mathbf{x}^{(k,i)}\right)\mathbf{x}^{(k,i)}\right)\\ &+\gamma\mathbf{\Phi}^\top\left(\mathbf{d}+\mathbf{v}^{(k)}-\overline{\boldsymbol{\eta}}^{(k)}/\gamma\right)\Big]\end{aligned} \tag{18}$$

which is the LM method, when augmented with an adaptation scheme for the regularization parameters $\lambda^{(i)} > 0$. The regularization parameter here helps to overcome some problematic cases, for example, the case when $\mathbf{J}_\theta^\top\mathbf{J}_\theta(\mathbf{x})$ is rank-deficient, by ensuring the existence of the unique minimum of the approximate cost function.

At each mADMM iteration, the computation in the $\mathbf{x}_{1:T}$-subproblem, such as (11), (16), and (18), has a high cost when T is large (e.g., $T = 10^8$). As discussed above, the main computational demand is indeed in the update of $\mathbf{x}_{1:T}$. Therefore, we utilize the equivalence between batch solutions and recursive smoothers, and then develop efficient augmented recursive smoother methods for solving the $\mathbf{x}_{1:T}$-subproblem.

IV. Augmented Recursive Smoothers

In the section, we will present the augmented KS, GN-IEKS, and LM-IEKS methods for solving the $\mathbf{x}_{1:T}$-subproblem.

A. Augmented Kalman Smoother for Affine Systems

Solving the $\mathbf{x}_{1:T}$-subproblem involves the minimization of a quadratic optimization problem, which can be efficiently solved by KS (see [51] for details). We rewrite the batch minimization problem (9) as

$$\begin{aligned}\mathbf{x}^\star_{1:T} = \arg\min_{\mathbf{x}_{1:T}} &\frac{1}{2}\sum_{t=1}^T\|\mathbf{y}_t-\mathbf{H}_t\mathbf{x}_t-\mathbf{e}_t\|^2_{\mathbf{R}_t^{-1}}\\ &+\frac{1}{2}\sum_{t=2}^T\|\mathbf{x}_t-\mathbf{A}_t\mathbf{x}_{t-1}-\mathbf{b}_t\|^2_{\mathbf{Q}_t^{-1}}+\frac{1}{2}\|\mathbf{x}_1-\mathbf{m}_1\|^2_{\mathbf{P}_1^{-1}}\\ &+\frac{\gamma}{2}\sum_{t=2}^T\left\|\mathbf{x}_t-\mathbf{B}_t\mathbf{x}_{t-1}-\mathbf{d}_t-\mathbf{v}_t+\frac{\overline{\boldsymbol{\eta}}_t}{\gamma}\right\|_2^2\\ &+\frac{\gamma}{2}\left\|\mathbf{x}_1-\mathbf{m}_1-\mathbf{v}_1+\frac{\overline{\boldsymbol{\eta}}_1}{\gamma}\right\|_2^2.\end{aligned} \tag{19}$$

It is worth noting that when $\mathbf{B}_t = \mathbf{0}$ and $\mathbf{d}_t = \mathbf{0}$, the cost function corresponds to the function minimized by KS, which leads to a similar method as was presented in [16]. For notational convenience, we leave out the iteration number k of mADMM in the following.

Here, we consider the general case, where $\mathbf{B}_t$ and $\mathbf{d}_t$ are nonzero. Such a case is more complicated as we cannot have two dynamic models in a state-space model. For building a dynamic state-space model, we need to fuse the terms in the pairs $(1/2)\|\mathbf{x}_t-\mathbf{A}_t\mathbf{x}_{t-1}-\mathbf{b}_t\|^2_{\mathbf{Q}_t^{-1}}$ and $(1/2)\|\mathbf{x}_t-\mathbf{B}_t\mathbf{x}_{t-1}-\mathbf{d}_t-\mathbf{v}_t+\overline{\boldsymbol{\eta}}_t/\gamma\|_2^2$, along with $(1/2)\|\mathbf{x}_1-\mathbf{m}_1\|^2_{\mathbf{P}_1^{-1}}$ and $(1/2)\|\mathbf{x}_1-\mathbf{m}_1-\mathbf{v}_1+\overline{\boldsymbol{\eta}}_1/\gamma\|_2^2$ into single terms. We combine matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ to an artificial transition matrix $\tilde{\mathbf{A}}_t$, fuse $\mathbf{b}_t$ and $(\mathbf{d}_t+\mathbf{v}_t-\overline{\boldsymbol{\eta}}_t/\gamma)$ to an artificial bias $\tilde{\mathbf{b}}_t$, and introduce an artificial covariance $\tilde{\mathbf{Q}}_t$, which yields

$$\begin{aligned}\tilde{\mathbf{A}}_t &= \left(\mathbf{Q}_t^{-1}+\gamma\mathbf{I}\right)^{-1}\left(\mathbf{Q}_t^{-1}\mathbf{A}_t+\gamma\mathbf{B}_t\right)\\ \tilde{\mathbf{b}}_t &= \left(\mathbf{Q}_t^{-1}+\gamma\mathbf{I}\right)^{-1}\left(\mathbf{Q}_t^{-1}\mathbf{b}_t+\gamma\mathbf{d}_t+\gamma\mathbf{v}_t-\overline{\boldsymbol{\eta}}_t\right)\\ \tilde{\mathbf{Q}}_t^{-1} &= \mathbf{Q}_t^{-1}+\gamma\mathbf{I}.\end{aligned} \tag{20}$$

Now, the new artificial dynamic model (20) allows us to use KS to solve the minimization problem. Problem (19) becomes

$$\mathbf{x}^{\star}_{1:T} = \arg\min_{\mathbf{x}_{1:T}} \frac{1}{2}\sum_{t=1}^{T}\|\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t - \mathbf{e}_t\|^2_{\mathbf{R}_t^{-1}} + \frac{1}{2}\|\mathbf{x}_t - \tilde{\mathbf{A}}_t\mathbf{x}_{t-1} - \tilde{\mathbf{b}}_t\|^2_{\tilde{\mathbf{Q}}_t^{-1}} + \frac{1}{2}\|\mathbf{x}_1 - \tilde{\mathbf{m}}_1\|^2_{\tilde{\mathbf{P}}_1^{-1}} \tag{21}$$

which corresponds to a state-space model, where additionally the initial state has mean $\tilde{\mathbf{m}}_1 = (\mathbf{P}_1^{-1} + \gamma\mathbf{I})^{-1}\,(\mathbf{P}_1^{-1}\mathbf{m}_1 + \gamma\mathbf{m}_1 + \gamma\mathbf{v}_t - \overline{\boldsymbol{\eta}}_t)$ and covariance $\tilde{\mathbf{P}}_1^{-1} = \mathbf{P}_1^{-1} + \gamma\mathbf{I}$. The solution in (21) can be then computed by running KS on the augmented state-space model

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t \mid \tilde{\mathbf{A}}_t\mathbf{x}_{t-1} + \tilde{\mathbf{b}}_t, \tilde{\mathbf{Q}}_t\right) \tag{22a}$$

$$p(\mathbf{y}_t \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{y}_t \mid \mathbf{H}_t\mathbf{x}_t + \mathbf{e}_t, \mathbf{R}_t). \tag{22b}$$

The augmented KS requires only $\mathcal{O}(N_x^3 T)$ operations, which are much less than the corresponding batch solution in (11). The augmented KS method is summarized in Algorithm 1.

B. Gauss–Newton IEKS for Nonlinear Systems

The solution of (15) has similar computational scaling challenges as the affine case discussed in the previous section. However, we can use the equivalence of IEKS and GN [8] to construct an efficient iterative solution for the optimization problem in the primal space. In the GN-IEKS method, we first approximate the nonlinear model by linearisation and then use KS on the linearized model. The $\mathbf{x}_{1:T}$-subproblem now takes the form of (7a). In IEKS, at the i:th iteration, we form affine approximations of $\mathbf{a}_t(\mathbf{x}_{t-1})$ and $\mathbf{h}_t(\mathbf{x}_t)$ as follows:

$$\begin{aligned}\mathbf{a}_t(\mathbf{x}_{t-1}) &\approx \mathbf{a}_t\left(\mathbf{x}_{t-1}^{(i)}\right) + \mathbf{J}_{a_t}\left(\mathbf{x}_{t-1}^{(i)}\right)\left(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^{(i)}\right)\\ \mathbf{h}_t(\mathbf{x}_t) &\approx \mathbf{h}_t\left(\mathbf{x}_t^{(i)}\right) + \mathbf{J}_{h_t}\left(\mathbf{x}_t^{(i)}\right)\left(\mathbf{x}_t - \mathbf{x}_t^{(i)}\right).\end{aligned} \tag{23}$$

We replace the nonlinear functions in the cost function with the above approximations, and compute the next iterate as the solution to the minimization problem

$$\begin{aligned}\mathbf{x}_{1:T}^{(i+1)} = \arg\min_{\mathbf{x}_{1:T}} &\frac{1}{2}\left\|\mathbf{y}_t - \mathbf{h}_t\left(\mathbf{x}_t^{(i)}\right) + \mathbf{J}_{h_t}\left(\mathbf{x}_t^{(i)}\right)\left(\mathbf{x}_t - \mathbf{x}_t^{(i)}\right)\right\|^2_{\mathbf{R}_t^{-1}}\\ &+ \frac{1}{2}\sum_{t=2}^{T}\left\|\mathbf{x}_t - \mathbf{a}_t\left(\mathbf{x}_{t-1}^{(i)}\right)\right.\\ &\qquad\left. + \mathbf{J}_{a_t}\left(\mathbf{x}_{t-1}^{(i)}\right)\left(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^{(i)}\right)\right\|^2_{\mathbf{Q}_t^{-1}}\\ &+ \frac{\gamma}{2}\sum_{t=2}^{T}\left\|\mathbf{x}_t - \mathbf{B}_t\mathbf{x}_{t-1} - \mathbf{d}_t - \mathbf{v}_t + \frac{\overline{\boldsymbol{\eta}}_t}{\gamma}\right\|_2^2\\ &+ \frac{\gamma}{2}\left\|\mathbf{x}_1 - \mathbf{m}_1 - \mathbf{v}_1 + \frac{\overline{\boldsymbol{\eta}}_1}{\gamma}\right\|_2^2 + \frac{1}{2}\|\mathbf{x}_1 - \mathbf{m}_1\|^2_{\mathbf{P}_1^{-1}}\end{aligned} \tag{24}$$

which is equivalent to (19) with

$$\begin{aligned}\mathbf{A}_t &= \mathbf{J}_{a_t}\left(\mathbf{x}_{t-1}^{(i)}\right), & \mathbf{b}_t &= \mathbf{a}_t\left(\mathbf{x}_{t-1}^{(i)}\right) - \mathbf{J}_{a_t}\left(\mathbf{x}_{t-1}^{(i)}\right)\mathbf{x}_{t-1}^{(i)}\\ \mathbf{H}_t &= \mathbf{J}_{h_t}\left(\mathbf{x}_t^{(i)}\right), & \mathbf{e}_t &= \mathbf{h}_t\left(\mathbf{x}_t^{(i)}\right) - \mathbf{J}_{h_t}\left(\mathbf{x}_t^{(i)}\right)\mathbf{x}_t^{(i)}.\end{aligned} \tag{25}$$

Algorithm 1: Augmented KS

Input: $\mathbf{y}_t$, $\mathbf{B}_t$, $\mathbf{d}_t$, $\mathbf{A}_t$, $\mathbf{H}_t$, $\mathbf{R}_t$, $\mathbf{Q}_t$, $\mathbf{v}^{(k)}$, $\overline{\boldsymbol{\eta}}^{(k)}$, $\mathbf{m}_1$, $\mathbf{P}_1$, and γ.

Output: $\mathbf{x}^{*}_{1:T}$.

compute $\tilde{\mathbf{A}}_t$, $\tilde{\mathbf{Q}}_t$, and $\tilde{\mathbf{b}}_t$ by (20);
for $t = 1, \ldots, T$ **do**
  $\mathbf{m}_t^- = \tilde{\mathbf{A}}_t\mathbf{m}_{t-1} + \tilde{\mathbf{b}}_t$;
  $\mathbf{P}_t^- = \tilde{\mathbf{A}}_t\,\mathbf{P}_{t-1}\,\tilde{\mathbf{A}}_t^{\top} + \tilde{\mathbf{Q}}_t$;
  $\mathbf{S}_t = \mathbf{H}_t\,\mathbf{P}_t^-\,\mathbf{H}_t^{\top} + \mathbf{R}_t$;
  $\mathbf{K}_t = \mathbf{P}_t^-\,\mathbf{H}_t^{\top}\,[\mathbf{S}_t]^{-1}$;
  $\mathbf{m}_t = \mathbf{m}_t^- + \mathbf{K}_t\big(\mathbf{y}_t - (\mathbf{H}_t\,\mathbf{m}_t^- + \mathbf{e}_t)\big)$;
  $\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t\,\mathbf{S}_t\,[\mathbf{K}_t]^{\top}$;
end
$\mathbf{m}_T^s = \mathbf{m}_T$ and $\mathbf{P}_T^s = \mathbf{P}_T$;
for $t = T-1, \ldots, 1$ **do**
  $\mathbf{G}_t = \mathbf{P}_t\,\tilde{\mathbf{A}}_{t+1}^{\top}\,[\mathbf{P}_{t+1}^-]^{-1}$;
  $\mathbf{m}_t^s = \mathbf{m}_t + \mathbf{G}_t\left(\mathbf{m}_{t+1}^s - \mathbf{m}_{t+1}^-\right)$;
  $\mathbf{P}_t^s = \mathbf{P}_t + \mathbf{G}_t\left(\mathbf{P}_{t+1}^s - \mathbf{P}_{t+1}^-\right)\mathbf{G}_t^{\top}$;
end
return $\mathbf{x}^{*}_{1:T} = \mathbf{m}^s_{1:T}$;

Algorithm 2: GN-IEKS

Input: $\mathbf{y}_t$, $\mathbf{B}_t$, $\mathbf{d}_t$, $\mathbf{a}_t$, $\mathbf{h}_t$, $\mathbf{R}_t$, $\mathbf{Q}_t$, $\mathbf{v}^{(k)}$, $\overline{\boldsymbol{\eta}}^{(k)}$, $\mathbf{m}_1$, $\mathbf{P}_1$, and γ.

Output: $\mathbf{x}^{*}_{1:T}$.

set $i \leftarrow 0$ and start from a suitable initial guess $\mathbf{x}^{(0)}_{1:T}$;
while *not converged or* $i < I_{\max}$ **do**
  linearise $\mathbf{a}_t$ and $\mathbf{h}_t$ according to (23);
  compute $\tilde{\mathbf{A}}_t$, $\tilde{\mathbf{Q}}_t$, $\tilde{\mathbf{b}}_t$ by (20);
  compute $\mathbf{x}^{(i+1)}_{1:T}$ by (24) using the augmented KS;
  $i \leftarrow i + 1$;
end
return $\mathbf{x}^{*}_{1:T} = \mathbf{x}^{(i)}_{1:T}$;

The precise expressions of $\mathbf{B}_t$ and $\mathbf{d}_t$ depend on our choice of sparsity. When $\mathbf{q}_t$ is sparse, the expressions are given by

$$\mathbf{B}_t = \mathbf{J}_{a_t}\left(\mathbf{x}_{t-1}^{(i)}\right), \quad \mathbf{d}_t = \mathbf{a}_t\left(\mathbf{x}_{t-1}^{(i)}\right) - \mathbf{J}_{a_t}\left(\mathbf{x}_{t-1}^{(i)}\right)\mathbf{x}_{t-1}^{(i)} \tag{26}$$

which needs the same computations as in (20). Thus, we can solve the minimization problem in (7a) by iteratively linearizing the nonlinearities and then by applying KS. This turns out to be mathematically equivalent to applying GN to the batch problem as we did in Section III-C. The steps of the GN-IEKS method are summarized in Algorithm 2.

C. Levenberg–Marquardt IEKS

There also exists a connection between the LM and a modified version of IEKS. The LM-IEKS method [10] is based on replacing the minimization of the approximate cost function in (24) by a regularized minimization of the form

$$\mathbf{x}^{\star}_{1:T} = \arg\min_{\mathbf{x}_{1:T}} \frac{1}{2}\left\|\mathbf{y}_t - \mathbf{h}_t\left(\mathbf{x}_t^{(i)}\right) + \mathbf{J}_{h_t}\left(\mathbf{x}_t^{(i)}\right)\left(\mathbf{x}_t - \mathbf{x}_t^{(i)}\right)\right\|^2_{\mathbf{R}_t^{-1}}$$

Algorithm 3: LM-IEKS

Input: $\mathbf{y}_t$, $\mathbf{B}_t$, $\mathbf{d}_t$, $\mathbf{a}_t$, $\mathbf{h}_t$, $\mathbf{R}_t$, $\mathbf{Q}_t$, $\mathbf{v}^{(k)}$, $\overline{\boldsymbol{\eta}}^{(k)}$, $\mathbf{m}_1$ and $\mathbf{P}_1$; $\mathbf{S}_t$, γ, λ, and α.

Output: $\mathbf{x}^*_{1:T}$.

```
set i ← 0 and start from a suitable initial guess x_{1:T}^{(0)};
while not converged or i < I_max do
    linearise a_t and h_t according to (23);
    compute Ã_t, Q̃_t, b̃_t by (20);
    update x_{1:T}^{(i+1)} by (28) based on the augmented KS;
    if θ(x_{1:T}^{(i+1)}) < θ(x_{1:T}^{(i)}) then
        λ^{(i)} ← λ^{(i)}/α; i ← i + 1;
    else
        λ^{(i)} ← λ^{(i)} α;
    end
end
return x*_{1:T} = x_{1:T}^{(i)};
```

$$
\begin{aligned}
&+ \frac{1}{2}\sum_{t=2}^{T}\left\|\mathbf{x}_t - \mathbf{a}_t\left(\mathbf{x}_{t-1}^{(i)}\right) + \mathbf{J}_{a_t}\left(\mathbf{x}_{t-1}^{(i)}\right)\left(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^{(i)}\right)\right\|^2_{\mathbf{Q}_t^{-1}} \\
&+ \frac{\gamma}{2}\sum_{t=2}^{T}\left\|\mathbf{x}_t - \mathbf{B}_t\mathbf{x}_{t-1} - \mathbf{d}_t - \mathbf{v}_t + \frac{\overline{\boldsymbol{\eta}}_t}{\gamma}\right\|_2^2 \\
&+ \frac{1}{2}\|\mathbf{x}_1 - \mathbf{m}_1\|^2_{\mathbf{P}_1^{-1}} \\
&+ \frac{\lambda^{(i)}}{2}\sum_{t=1}^{T}\left\|\mathbf{x}_t - \mathbf{x}_t^{(i)}\right\|^2_{[\mathbf{S}_t^{(i)}]^{-1}} \\
&+ \frac{\gamma}{2}\left\|\mathbf{x}_1 - \mathbf{m}_1 - \mathbf{v}_1 + \frac{\overline{\boldsymbol{\eta}}_1}{\gamma}\right\|_2^2
\end{aligned} \tag{27}
$$

where we have assume that $\mathbf{S}^{(i)} = \operatorname{diag}(\mathbf{S}_1^{(i)}, \ldots, \mathbf{S}_T^{(i)})$. Similar to GN-IEKS, when $\mathbf{B}_t$ and $\mathbf{d}_t$ are nonzero, we need to build a new state-space model in order to have only one dynamic model. Following [10], the regularization can be implemented by defining an additional pseudomeasurement $\mathbf{z}_t = \mathbf{x}_t^{(i)}$ with a noise covariance $\Sigma_t^{(i)} = \mathbf{S}_t^{(i)}/\lambda^{(i)}$. Using (20) and (25), we have the augmented state-space model

$$
\begin{aligned}
p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) &= \mathcal{N}\left(\mathbf{x}_t \mid \tilde{\mathbf{A}}_t\mathbf{x}_{t-1} + \tilde{\mathbf{b}}_t, \tilde{\mathbf{Q}}_t\right) \\
p(\mathbf{y}_t \mid \mathbf{x}_t) &= \mathcal{N}(\mathbf{y}_t \mid \mathbf{H}_t\mathbf{x}_t + \mathbf{e}_t, \mathbf{R}_t) \\
p(\mathbf{z}_t \mid \mathbf{x}_t) &= \mathcal{N}\left(\mathbf{z}_t \mid \mathbf{x}_t, \Sigma_t^{(i)}\right)
\end{aligned} \tag{28}
$$

which provides the minimum of the cost function as the KS solution. By combining this with $\lambda^{(i)}$ adaptation and iterating, we can implement the LM algorithm for the $\mathbf{x}_{1:T}$-subproblem using the recursive smoother (see [10]). See Algorithm 3 for more details.

D. Discussion

All the methods discussed above, namely, augmented KS, GN-IEKS, and LM-IEKS, provide efficient ways to solve the $\mathbf{x}_{1:T}$-subproblem. When we leverage the Markov structure of the $\mathbf{x}_{1:T}$-subproblem arising in the mADMM iteration, we can significantly reduce the computation burden. In particular, when functions $\mathbf{a}_t(\mathbf{x}_{t-1})$ and $\mathbf{h}_t(\mathbf{x}_t)$ are affine, the augmented KS method can be used in the $\mathbf{x}_{1:T}$-subproblem [see (19)]. Both GN-IEKS and LM-IEKS are based on the use of linearisation of the functions $\mathbf{a}_t(\mathbf{x}_{t-1})$ and $\mathbf{h}_t(\mathbf{x}_t)$, and they work well for most nonlinear minimization problems. However, when the Jacobians [e.g., $\mathbf{J}_{a_t}(\mathbf{x}_{t-1}^{(i)})$ or $\mathbf{J}_{h_t}(\mathbf{x}_t^{(i)})$ in (23)] are rank-deficient, the GN-IEKS method cannot be used. As a robust extension of GN-IEKS, LM-IEKS significantly improves the performance of GN-IEKS. It should be noted that when the regularization term is not used in LM-IEKS [when $\lambda^{(i)} = 0$], then LM-IEKS reduces to GN-IEKS [10].

V. Convergence Analysis

In this section, we prove that under mild assumptions and a proper choice of the penalty parameter, our KS-mADMM, GN-IEKS-mADMM, and LM-IEKS-mADMM methods converge to a stationary point of the original problem. Although the convergence of the mADMM has already been proven, the existing results strongly depend on convexity assumptions or Lipschitz continuity conditions (see [52]–[54]). In the analysis, we require neither the convexity of the objective function nor Lipschitz continuity conditions. Instead, we use a milder condition on the amenability. This allows us to establish the convergence of the three methods.

For the case when the functions $\mathbf{a}_t(\mathbf{x}_{t-1})$ and $\mathbf{h}_t(\mathbf{x}_t)$ are affine [see (2)], we have the following lemma.

Lemma 1: Let $\{\mathbf{x}_{1:T}^{(k)}, \mathbf{w}_{1:T}^{(k)}, \mathbf{v}_{1:T}^{(k)}, \boldsymbol{\eta}_{1:T}^{(k)}\}$ be the iterates generated by (7). Then, we have

$$
\begin{aligned}
&\left\|\begin{bmatrix}\mathbf{v}^{(k+1)} \\ \boldsymbol{\eta}^{(k+1)}\end{bmatrix} - \begin{bmatrix}\mathbf{v}^\star \\ \boldsymbol{\eta}^\star\end{bmatrix}\right\|^2_{\boldsymbol{\Omega}} \\
&\quad\le \left\|\begin{bmatrix}\mathbf{v}^{(k)} \\ \boldsymbol{\eta}^{(k)}\end{bmatrix} - \begin{bmatrix}\mathbf{v}^\star \\ \boldsymbol{\eta}^\star\end{bmatrix}\right\|^2_{\boldsymbol{\Omega}} - \left\|\begin{bmatrix}\mathbf{v}^{(k)} \\ \boldsymbol{\eta}^{(k)}\end{bmatrix} - \begin{bmatrix}\mathbf{v}^{(k+1)} \\ \boldsymbol{\eta}^{(k+1)}\end{bmatrix}\right\|^2_{\boldsymbol{\Omega}}
\end{aligned} \tag{29}
$$

where $\boldsymbol{\Omega} = \begin{bmatrix}\gamma\mathbf{I} + \mathbf{G}^\top\mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}/\gamma\end{bmatrix}$ and $\mathbf{G} = \begin{bmatrix}\mathbf{G}_1 \\ \vdots \\ \mathbf{G}_T\end{bmatrix}$.

Proof: See Appendix A. ∎

We will then establish the convergence rate of the proposed method in terms of the iteration number.

Theorem 1 (Convergence of KS-mADMM): Let $\mathbf{Q}_t$ and $\mathbf{P}_1$ be positive semidefinite matrices. Then, the sequence $\{\mathbf{x}_{1:T}^{(k)}, \mathbf{w}_{1:T}^{(k)}, \mathbf{v}_{1:T}^{(k)}, \boldsymbol{\eta}_{1:T}^{(k)}\}$ generated by KS-mADMM converges to a stationary point $(\mathbf{x}_{1:T}^\star, \mathbf{w}_{1:T}^\star, \mathbf{v}_{1:T}^\star, \boldsymbol{\eta}_{1:T}^\star)$ with the rate $o(1/k)$.

Proof: The proof is based on the convexity of the function. Because of the equivalence between mADMM and KS-mADMM, we start by establishing the convergence of mADMM. When $\mathbf{Q}_t$ and $\mathbf{P}_1$ are positive semidefinite, the function in (5) is convex. Because of $[\boldsymbol{\Phi} \quad \mathbf{0}][\mathbf{0} \quad \mathbf{I}]^\top = \mathbf{0}$, we can write $\mathbf{x}$ and $\mathbf{w}$ into a function $\Xi(\boldsymbol{\zeta})$ in the batch form [52].

For simplicity of notation, we define $\mathbf{s} = \begin{bmatrix}\mathbf{v} & \boldsymbol{\eta}\end{bmatrix}^\top$. Using Lemma 1, we obtain

$$
\begin{aligned}
&\Xi(\boldsymbol{\zeta}^\star) - \Xi\left(\boldsymbol{\zeta}^{(k)}\right) + \left(\boldsymbol{\xi}^\star - \boldsymbol{\xi}^{(k)}\right)^\top F(\boldsymbol{\xi}^\star) \\
&\quad + \left\|\mathbf{s}^\star - \mathbf{s}^{(k)}\right\|^2_{\boldsymbol{\Omega}} \ge \left\|\mathbf{s}^\star - \mathbf{s}^{(k+1)}\right\|^2_{\boldsymbol{\Omega}}
\end{aligned} \tag{30}
$$

where $\boldsymbol{\xi}$ and $F(\boldsymbol{\xi})$ are defined in Appendix A [see (37)]. We sum the inequality (30) from 0 to k and divide each term by $k+1$. Since $\|\mathbf{s}^\star - \mathbf{s}^{(k+1)}\|^2_{\boldsymbol{\Omega}} \geq 0$, we then have

$$\frac{1}{k+1}\sum_{i=0}^{k} \Xi\left(\boldsymbol{\zeta}^{(k)}\right) - \Xi(\boldsymbol{\zeta}^\star) + \left(\frac{1}{k+1}\sum_{i=0}^{k} \boldsymbol{\xi}^{(k)} - \boldsymbol{\xi}^\star\right)^\top F(\boldsymbol{\xi}^\star) \leq \frac{1}{k+1}\left\|\mathbf{s}^\star - \mathbf{s}^{(0)}\right\|^2_{\boldsymbol{\Omega}}. \tag{31}$$

Let $\bar{\boldsymbol{\zeta}}^{(k)} = (1/[k+1])\boldsymbol{\zeta}^{(k)}$ and $\bar{\boldsymbol{\xi}}^{(k)} = (1/[k+1])\boldsymbol{\xi}^{(k)}$. Because of the convexity of Ξ, we further write (31) as

$$\Xi\left(\bar{\boldsymbol{\zeta}}^{(k)}\right) - \Xi(\boldsymbol{\zeta}^\star) + \left(\bar{\boldsymbol{\xi}}^{(k)} - \boldsymbol{\xi}^\star\right)^\top F(\boldsymbol{\xi}^\star) \leq \frac{1}{k+1}\left\|\mathbf{s}^\star - \mathbf{s}^{(0)}\right\|^2_{\boldsymbol{\Omega}}. \tag{32}$$

The convergence rate $o(1/k)$ of mADMM is thus established. As batch mADMM is equivalent to KS-mADMM, then the sequence $\{\mathbf{x}^{(k)}, \mathbf{w}^{(k)}, \mathbf{v}^{(k)}, \boldsymbol{\eta}^{(k)}\}$ and the sequence $\{\mathbf{x}^{(k)}_{1:T}, \mathbf{w}^{(k)}_{1:T}, \mathbf{v}^{(k)}_{1:T}, \boldsymbol{\eta}^{(k)}_{1:T}\}$ are identical. This concludes the proof. ■

When the functions $\mathbf{a}_t(\mathbf{x}_{t-1})$ and $\mathbf{h}_t(\mathbf{x}_t)$ are nonlinear, we have the function

$$s(\mathbf{x}) \triangleq \frac{1}{2}\|\mathbf{y} - \mathbf{h}(\mathbf{x})\|^2_{\mathbf{R}^{-1}} + \frac{1}{2}\|\mathbf{m} - \mathbf{a}(\mathbf{x})\|^2_{\mathbf{Q}^{-1}}. \tag{33}$$

Definition 1: The function $s(\mathbf{x})$ is *strongly amenable* [55] at $\mathbf{x}$ when the condition

$$\left[\mathbf{R}^{-1/2}\mathbf{J}_h; \mathbf{Q}^{-1/2}\mathbf{J}_a\right]^\top \mathbf{z} = \mathbf{0} \tag{34}$$

is satisfied only when $\mathbf{z}$ is 0.

Let $s(\mathbf{x})$ be strongly amenable. Then, $s(\mathbf{x})$ will be *prox-regular* [56]. We are now ready for introducing the following lemma.

Lemma 2 (Bounded and Nonincreasing Sequence): Assume that $\delta_+(\boldsymbol{\Phi}^\top\boldsymbol{\Phi}) > 0$ and $s(\mathbf{x})$ is strongly amenable. Then, there exists $\gamma > 0$ such that sequence $\mathcal{L}_\gamma(\mathbf{x}^{(k)}, \mathbf{w}^{(k)}, \mathbf{v}^{(k)}; \boldsymbol{\eta}^{(k)})$ is bounded and nonincreasing.

Proof: See Appendix B. ■

Next, we present the main theoretical result.

Theorem 2 (Convergence of GN-IEKS-mADMM): Let the assumptions in Lemma 2 be satisfied. Then there exists $\gamma > 0$ such that the sequence $\{\mathbf{x}^{(k)}_{1:T}, \mathbf{w}^{(k)}_{1:T}, ..\mathbf{v}^{(k)}_{1:T}, \boldsymbol{\eta}^{(k)}_{1:T}\}$ generated by GN-IEKS-mADMM locally converges to a local minimum.

Proof: By Lemma 2, the sequence $\mathcal{L}_\gamma(\mathbf{x}^{(k)}, \mathbf{w}^{(k)}, \mathbf{v}^{(k)}; \boldsymbol{\eta}^{(k)})$ is bounded and nonincreasing. Based on our paper [16], the $\mathbf{x}$-subproblem has a local minimum $\mathbf{x}^\star$. $\mathbf{w}$ and $\mathbf{v}$ subproblems are convex [57]. We then conclude that the iterative sequence $\{\mathbf{x}^{(k)}, \mathbf{w}^{(k)}, \mathbf{v}^{(k)}, \boldsymbol{\eta}^{(k)}\}$ locally converges to a local minimum $(\mathbf{x}^\star, \mathbf{w}^\star, \mathbf{v}^\star, \boldsymbol{\eta}^\star)$. According to [8], GN is equivalent to IEKS. Thus, we deduce that the iterative sequence $\{\mathbf{x}^{(k)}_{1:T}, \mathbf{w}^{(k)}_{1:T}, \mathbf{v}^{(k)}_{1:T}, \boldsymbol{\eta}^{(k)}_{1:T}\}$ is convergent to a local minimum $(\mathbf{x}^\star_{1:T}, \mathbf{w}^\star_{1:T}, \mathbf{v}^\star_{1:T}, \boldsymbol{\eta}^\star_{1:T})$. ■

Lemma 3 (Convergence of LM): Assume that the norm of Hessian $\mathbf{H}_\theta(\mathbf{x})$ is bounded by a positive constant $\kappa < \max\{\gamma\delta_+(\boldsymbol{\Phi}^\top\boldsymbol{\Phi}), \lambda^{(i)}\delta_+([\mathbf{S}^{(i)}]^{-1})\}$. Then, LM is locally (linearly) convergent. The convergence is quadratic when $\kappa \to 0$.

Proof: See Appendix C. ■

Theorem 3 (Convergence of LM-IEKS-mADMM): Let the assumptions of Lemmas 2 and 3 be satisfied. Then, there exists $\lambda^{(i)}, \gamma > 0$ such that the sequence $\{\mathbf{x}^{(k)}_{1:T}, \mathbf{w}^{(k)}_{1:T}, \mathbf{v}^{(k)}_{1:T}, \boldsymbol{\eta}^{(k)}_{1:T}\}$ generated by LM-IEKS-mADMM converges to a local minimum $(\mathbf{x}^\star_{1:T}, \mathbf{w}^\star_{1:T}, \mathbf{v}^\star_{1:T}, \boldsymbol{\eta}^\star_{1:T})$.

Proof: Similar to Theorem 2, we use Lemma 2 to establish that the sequence $\mathcal{L}_\gamma(\mathbf{x}^{(k)}, \mathbf{w}^{(k)}, \mathbf{v}^{(k)}; \boldsymbol{\eta}^{(k)})$ is bounded and nonincreasing. Due to the convexity, the $\mathbf{w}$ and $\mathbf{v}$ subproblems have a local minimum. By Lemma 3, the sequence $\mathbf{x}^{(i)}$ generated by LM converges to $\mathbf{x}^\star$. Then, the sequence $\{\mathbf{x}^{(k)}, \mathbf{w}^{(k)}, \mathbf{v}^{(k)}, \boldsymbol{\eta}^{(k)}\}$ locally converges to a minimum $(\mathbf{x}^\star, \mathbf{w}^\star, \mathbf{v}^\star, \boldsymbol{\eta}^\star)$ since the sequence $\{\mathbf{x}^{(k)}, \mathbf{w}^{(k)}, \mathbf{v}^{(k)}, \boldsymbol{\eta}^{(k)}\}$ generated by LM is identical to $\{\mathbf{x}^{(k)}_{1:T}, \mathbf{w}^{(k)}_{1:T}, \mathbf{v}^{(k)}_{1:T}, \boldsymbol{\eta}^{(k)}_{1:T}\}$ generated by LM-IEKS [8], [10]. ■

VI. Numerical Experiments

In this section, we experimentally evaluate the proposed methods in a selection of different applications, including linear target tracking problems, multisensor range measurement problems, ship trajectory tracking, audio restoration, and autonomous vehicle tracking. As for the convergence criteria, we can easily verify that the assumptions for convergence are satisfied for the linear/affine examples in Sections VI-A–VI-E. In addition, the nonlinear coordinated turn model in Section VI-D also satisfies assumptions for convergence. However, for the distance measurement in Section VI-B, it is hard to establish strong amenability although empirically the convergence occurs.

A. Linear Target Tracking Problems

In the first experiment, we consider simulated tracking of a moving target (such as car) with the Wiener velocity model [6] as the dynamic model and with noisy location measurements. In the simulation, the process noise $\mathbf{q}_t$ was set to be 0 with probability 0.8 at every step t. State $\mathbf{x}_t$ has the location $(x_{t,1}, x_{t,2})$ and the velocities $(x_{t,3}, x_{t,4})$. The measurement model matrix and the measurement noise covariance are

$$\mathbf{H}_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{R}_t = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}.$$

The transition matrix and the process noise covariance are

$$\mathbf{A}_t = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\mathbf{Q}_t = q_c \begin{bmatrix} \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & 0 & \Delta t & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 & \Delta t \end{bmatrix}.$$

We have $\Delta t = 0.1$, $q_c = 0.5$, $\sigma = 0.3$, $T = 100$, $\mathbf{m}_1 = \begin{bmatrix} 0.1 & 0 & 0.1 & 0 \end{bmatrix}^\top$, and $\mathbf{P}_1$ is an identity matrix. We set the matrix $\mathbf{G}_{g,t}$ to an identity matrix and use the parameters $\gamma = 1$, $\mu = 1$, and $K_{\max} = 50$.

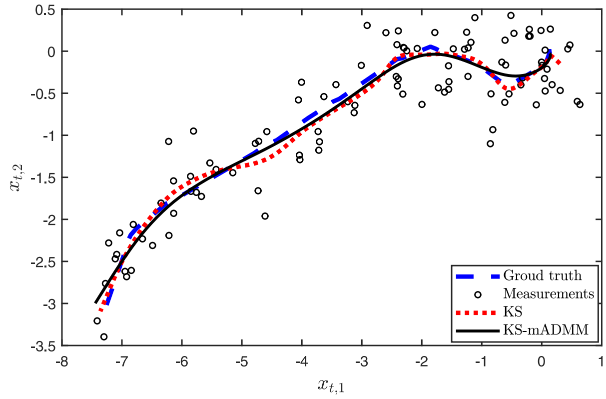

Fig. 1. Signals, measurements, and the estimates in the linear tracking problem. The values of $\mathbf{x}_{\mathrm{err}}$ are 0.103 and 0.072 in KS and KS-mADMM, respectively.

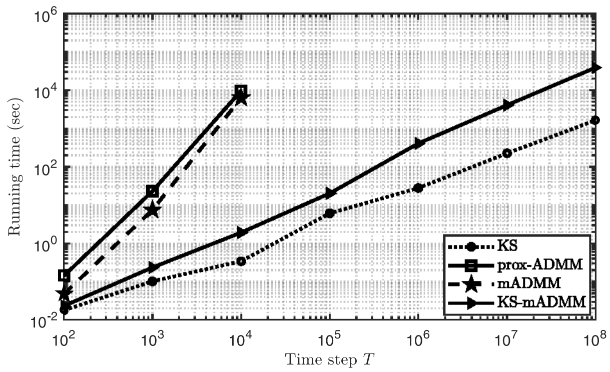

Fig. 2. Comparison of the running times in the linear car tracking example as function of the number of time steps.

We define the estimation error as

$$\mathbf{x}_{\mathrm{err}} = \frac{\sum_{t=1}^{T} \left\| \mathbf{x}_t^{(k)} - \mathbf{x}_t^{\mathrm{true}} \right\|_2}{\sum_{t=1}^{T} \left\| \mathbf{x}_t^{\mathrm{true}} \right\|_2}$$

where $\mathbf{x}_t^{\mathrm{true}}$ is the ground truth. The estimation results are plotted in Fig. 1, where the circles denote the noisy measurements and the blue dashed line denotes the true state. As we can seen, the KS-mADMM estimate (black line) is much closer to the ground truth than the KS estimate (red dashed line), which is also reflected by a lower error.

Recall that the difference in batch and recursive ADMM running time is dominated by the $\mathbf{x}_{1:T}$-subproblem. Fig. 2 demonstrates how the running time (sec) grows when T increases. Despite being mathematically equivalent, mADMM and KS-mADMM have very different running times. The running times of mADMM and proximal ADMM (prox-ADMM) have a similar growth rate; whereas, KS-mADMM has a growth rate that resembles a plain KS. Due to limited memory, we cannot report the results of the batch estimation methods (prox-ADMM and mADMM) when $T > 10^4$. At $T = 10^4$, the running times of KS, KS-mADMM, prox-ADMM, and mADMM were 0.34, 1.92, 6284, and 9646 s, respectively. The proposed method is computationally inexpensive, which makes it suitable for solving real-world applications, such as the marine vessel tracking in Section VI-C.

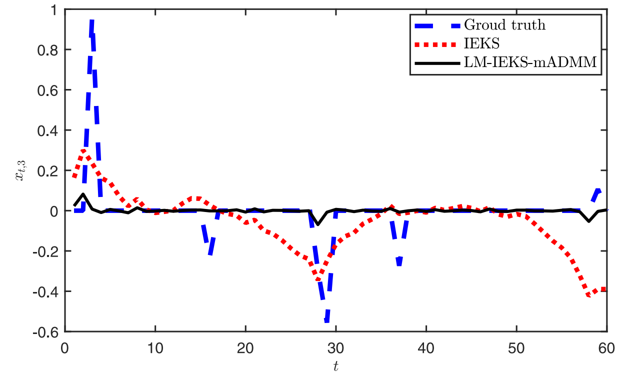

Fig. 3. Estimated trajectory in the nonlinear system. The relative errors are 0.53 and 0.46 generated by IEKS and LM-IEKS-ADMM.

B. Multisensor Range Measurement Problems

In this experiment, we consider a multisensor range measurement problem, where we have short periods of movement with regular stops. This problem frequently appears in many surveillance systems [3], [6]. The state $\mathbf{x}_t$ contains the position $(x_{t,1}, x_{t,2})$ and the velocities $(x_{t,3}, x_{t,4})$. The measurement dynamic model for sensor $n \in \{1, 2, 3\}$ is given by

$$\mathbf{h}_t^n(\mathbf{x}_t) = \sqrt{\left(x_{t,2} - s_y^n\right)^2 + \left(x_{t,1} - s_x^n\right)^2}$$

where (s_x^n, s_y^n) is the position of sensor n. The transition function $\mathbf{a}_t(\mathbf{x}_{t-1})$ is

$$\mathbf{a}_t(\mathbf{x}_{t-1}) = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{t-1}. \tag{35}$$

The covariances are $\mathbf{R}_t = \mathrm{diag}(0.2^2, 0.2^2)$ and $\mathbf{Q}_t = \mathrm{diag}(0.01, 0.01, 0.1, 0.1)$. We set $\Delta t = 0.1$, $T = 60$, $(s_x^1, s_y^1) = (0, -0.5)$, $(s_x^2, s_y^2) = (0.5, 0.6)$, $(s_x^3, s_y^3) = (0.5, 0.6)$, $\mathbf{m}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top$, and $\mathbf{P}_1 = \mathbf{I}/10$. We assume the target has many stops, which means the velocities $x_{t,3}$ and $x_{t,4}$ are sparse. We also set $\mathbf{G}_{g,t} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix}$ and use the parameters $\gamma = 1$, $\mu = 1$, $K_{\max} = 50$, and $I_{\max} = 5$. We plot the velocity variable $x_{t,3}$ corresponding to the time step t in Fig. 3, which indicates that our method (black line) can generate much more sparse results than the IEKS estimate (red dashed line).

Fig. 4 shows the relative error $\mathbf{x}_{\mathrm{err}}$ as a function of the iteration number. The values of $\mathbf{x}_{\mathrm{err}}$ generated by the regularization methods are below those generated by IEKS [1]. It also shows that the GN-mADMM, GN-IEKS-mADMM, LM-mADMM, and LM-IEKS-mADMM can find the optimal values in around 50 iterations. IEKS is the fastest method, but the relative error is the highest due to the lack of the sparsity prior (i.e., $\mu = 0$). GN-mADMM and GN-IEKS-mADMM have the same convergence results (as they are equivalent), while the latter uses less running time. Similarly, LM-mADMM and LM-IEKS-mADMM have the same convergence results, but LM-IEKS-mADMM needs less time to obtain the result than LM-mADMM. When the number of time steps T is moderate, all the running times are acceptable. But

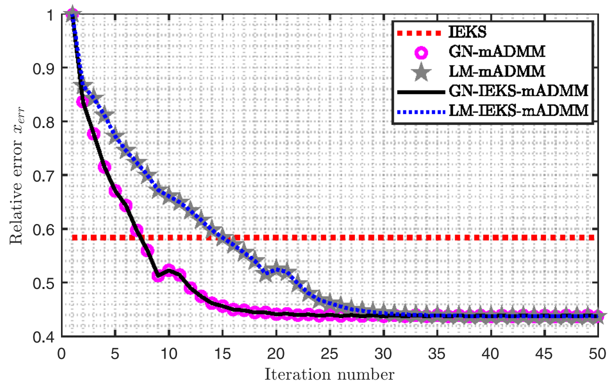

Fig. 4. Relative error $\mathbf{x}_{\mathrm{err}}$ versus iteration number.

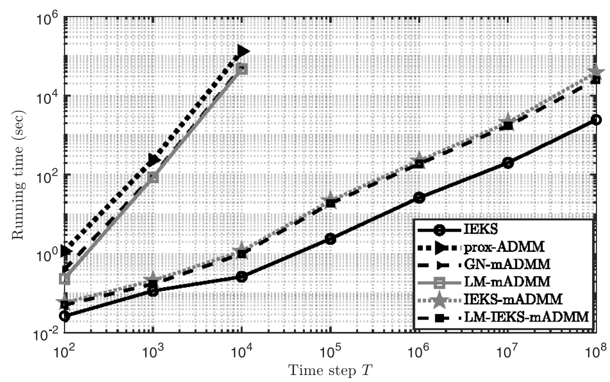

Fig. 5. Comparison of the running times in the range measurement example as function of the number of time steps (from 10^2 to 10^8).

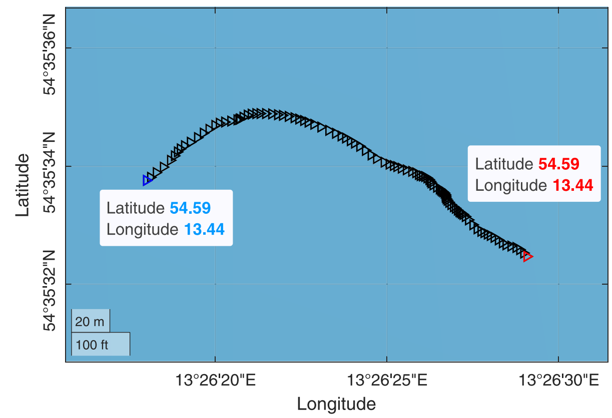

Fig. 6. Position (black markers) estimated by KS-mADMM. The starting coordinate is denoted blue marker, and the ending coordinate is a red marker. Contains data from the Danish Maritime Authority that is used in accordance with the conditions for the use of Danish public data.

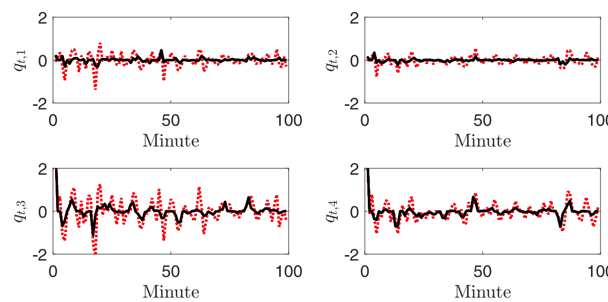

Fig. 7. Process noise estimated by KS-mADMM (black line) and KS (red dashed line).

when T is extremely large, the proposed methods provide a massive advantage.

Fig. 5 demonstrates how the running time (sec) grows when T increases. The proposed methods are compared to the state-of-the-art methods, including the prox-ADMM [46], mADMM [47], and IEKS [1]. Despite being mathematically equivalent, GN-mADMM and GN-IEKS-mADMM, LM-mADMM, and LM-IEKS-mADMM have very different running times. GN-IEKS-mADMM and LM-IEKS-mADMM are more efficient than the batch methods. Due to limited memory, we cannot report the results of the batch methods when $T > 10^4$. It is reasonable to conclude that in general, the proposed methods are competitive for extremely large-scale tracking and estimation problems. The proposed approaches are computationally inexpensive, which makes them suitable for solving real-world applications, such as ship trajectory-tracking in the next section.

C. Marine Vessel Tracking

In this experiment, we utilize the Wiener velocity model [6] with a sparse noise assumption to track a marine vessel trajectory. The latitude, longitude, speed, and course of the vessel have been captured by automatic identification system (AIS) equipment, collected by Danish Maritime Authority. Similar applications can be found in [5] and [39]. The state of the ship is measured at time intervals of 1 min. Matrices $\mathbf{H}_t$, $\mathbf{A}_t$, $\mathbf{Q}_t$, and $\mathbf{R}_t$ are the same with the settings in Section VI-A with $\Delta t = 1$, $q_c = 1$, $\sigma = 0.3$, $T = 100$, $\mathbf{m}_1 = \begin{bmatrix}0.1 & 0.1 & 0 & 0\end{bmatrix}^\top$, and $\mathbf{P}_1 = 100\mathbf{I}$. We assume the process noise $\mathbf{q}_t$ is sparse, set $\mathbf{G}_{g,t}$ to an identity matrix, and use the parameters $\gamma = 1$, $\mu = 1$, and $K_{\max} = 100$. The measurement data consist of 100 time points of the vessel locations.

Our method obtains the position (latitude and longitude) estimates as shown in Fig. 6. Fig. 7 shows that our method has sparser process noise than that estimated by the KS [1]. We then highlight the computational advantage of our method. The difference in running time is dominated by the $\mathbf{x}_{1:T}$-subproblem. The running times of KS, prox-ADMM, mADMM, and KS-mADMM were 0.34, 174, 172, and 5.63 s, respectively. The running times of mADMM and prox-ADMM are similar whereas KS-mADMM has a smaller running time that resembles the plain KS.

D. Autonomous Vehicle Tracking

To further show how our methods can speed up larger scale real-world problems, we apply GN-IEKS-mADMM to a vehicle tracking problem using real-world data. Global positioning system (GPS) data were collected in urban streets and roads around Helsinki, Tuusula, and Vantaa, Finland [58]. The urban environment contained many stops to traffic lights, crossings, turns, and various other situations. We ran the experiment using a coordinate turn model [1], where the state at time step t had the positions $(x_{t,1}, x_{t,2})$, the velocities $(x_{t,3}, x_{t,4})$,

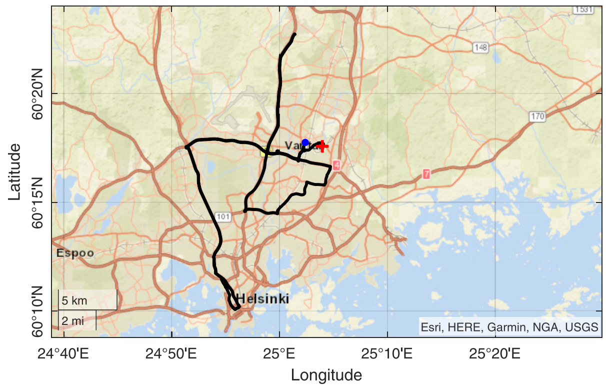

Fig. 8. Path tracking (black line) generated by GN-IEKS-mADMM. The starting position is blue point, and the ending position is red cross.

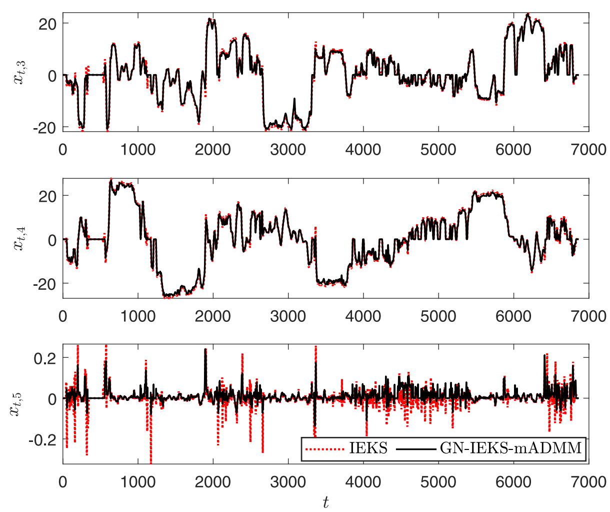

Fig. 9. Estimated velocities and angular velocities generated by IEKS (red dashed line) and the proposed method (black line).

and the angular velocity $x_{t,5}$. The number of time points T was 6865. We use the parameters $\gamma = 0.1$, $\mu = 1$, $K_{\max} = 300$, $I_{\max} = 5$, $\mathbf{m}_1 = \begin{bmatrix} 4.5 & 13.5 & 0 & 0 & 0 \end{bmatrix}^\top$, and $\mathbf{P}_1 = \mathrm{diag}(50, 50, 50, 50, 0.01)$. We utilized the matrix

$$\mathbf{G}_{g,t} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

to enforce the sparsity of the velocities and the angular velocity.

The plot in Fig. 8 demonstrates the path (black line) generated by our method. The running time of IEKS [32], GN-mADMM, and GN-IEKS-mADMM were 22, 13 520, and 2704 s, respectively. As we expected, although IEKS is the fastest, the L_2-penalized regularization methods push more of the velocities and the angle to 0, which is shown in Fig. 9. The IEKS estimate has many large peaks that appear as a result of large residuals, and GN-IEKS-mADMM has more sparse results.

E. Audio Signal Restoration

The proposed technique can be readily applied to the problem of noise reduction in audio signals. We adopt a Gabor regression model [19]

$$y(\tau) = \sum_{m=0}^{M/2} \sum_{n=0}^{N-1} c_{m,n} g_{m,n}(\tau) + r(\tau), \qquad \tau = 0, \dots, T-1$$

where signals are represented as a weighted sum of the Gabor atoms $g_{m,n}(\tau) = w_n(\tau) \exp(2\pi \mathrm{i}[m/M]\tau)$. Terms $w_n(\tau)$ correspond to a window function with bounded support centred at time instants τ_n (windows are placed so that the time axis is with tiled evenly). Sparsity is promoted through the L_2^1 pairwise grouping pattern described in Section II: $\sum_{m,n} \mu_{m,n} \|c_{m,n}\|_2$. The real representation of complex coefficients $c_{m,n}$ used in [19] is adopted. This batch problem is restated in terms of a state-space model: signal $\mathbf{y}$ is separated into P chunks $\mathbf{y}_t$ of length L and state vectors $\mathbf{x}_t = [\mathbf{c}_{2(t-1)}; \mathbf{c}_{2t-1}; \mathbf{c}_{2t}]^\top$ are defined, $\mathbf{c}_t$ being the subvector associated to each frame. Let $\mathbf{H}_0$ be a matrix containing the nonzero values of the Gabor basis functions $\mathbf{g}_{0,0}, \dots, \mathbf{g}_{M/2,0}$ as columns. Thus, atoms in subsequent frames are time-shifted replicas of this basic set and $\|\mathbf{y} - \mathbf{D}\mathbf{c}\|^2$ ($\mathbf{D}$ a *dictionary* matrix containing all atoms) can be replaced by $\sum_{t=1}^{P} \|\mathbf{y}_t - \mathbf{H}_* \mathbf{x}_t\|^2 + \sum_{t=1}^{P} \|\mathbf{x}_t - \mathbf{A}_t \mathbf{x}_{t-1}\|^2$, with $\mathbf{H}_* = \begin{bmatrix} \mathbf{H}_u & \mathbf{H}_0 & \mathbf{H}_\ell \end{bmatrix}$ and $\mathbf{A}_t = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0}; & \mathbf{0} & \mathbf{0} & \mathbf{0}; & \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix}$. Terms $\mathbf{H}_u$ and $\mathbf{H}_\ell$ are truncated versions of $\mathbf{H}_0$ corresponding to the contribution of the adjacent overlapping frames.

The algorithm is tested on a ∼3-s long glockenspiel excerpt sampled at 22050 [Hz] and contaminated with artificial background noise with signal-to-noise ratio (SNR) 5dB. Experiments are carried out in an Intel Core i7 @ 2.50 GHz, 16-GB RAM, with parameters $\gamma = 5$, $\mu = 2.6$, and $K_{\max} = 500$, and a window length $L = 512$. Kalman gain matrices are precomputed. Reflecting the power spectrum of typical audio signals, which decays with frequency, penalization is made frequency-dependent by setting $\mu_{m,n} = \mu / f(m)$, with $f(m)$ a decreasing modulating function (e.g., a Butterworth filter gain), in a similar fashion to [18]. Coefficients are initialised at 0. The average output SNR is 12.4 with an average running time of 64.6 s in 20 realizations. Fig. 10 shows the visual reconstruction results.

In comparison, the Gibbs sampling schemes for models (e.g., [19]) yield noisier restorations with comparable computing times. We analyzed the same example using the Gibbs sampler with 500 iterations, 250 burn-in periods. Hyperparameters and initial values are chosen to ensure a fair comparison with the KS-mADMM method (unfavorable initialization may induce longer convergence times). With a runtime of ∼180 s, the Gibbs algorithm yields an output SNR of ∼15 dB. The perceptual evaluation of audio quality (PEAQ) [59], a measure that incorporates psycho-acoustic criteria to assess audio signals, is adopted. The objective difference grade (ODG) indicator derived from PEAQ is used to compare the reconstructions with respect to the clean reference signal, obtaining ODG $= -3.910$ for clean signal against noisy input, ODG $= -3.846$ for clean signal against the Gibbs reconstruction, and ODG $= -3.637$ for clean signal against

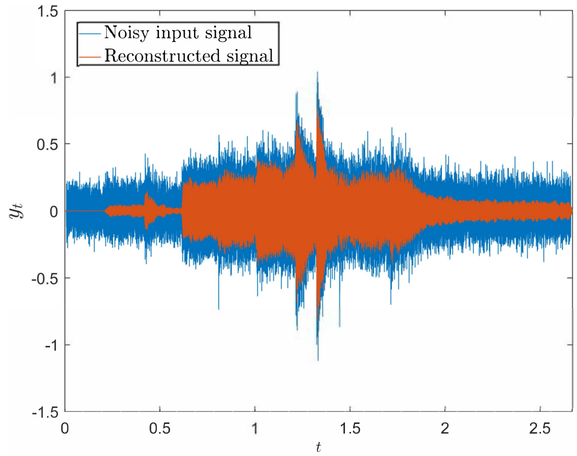

Fig. 10. Reconstructed glockenspiel excerpt.

KS-mADMM reconstruction (the closer to 0, the better). Despite the lower SNR (12.4 dB), the KS-ADMM reconstruction sounds cleaner (i.e., has fewer audio artifacts) than its Gibbs counterpart, which is consistent with the ODG values obtained. Devising appropriate temporal evolution models for the audio synthesis coefficients over time and investigating self-adaptive schemes for the estimation of μ (here, tuned empirically) are topics of future research.

VII. Conclusion and Discussion

In this article, we have presented efficient smoothing-and-splitting methods for solving regularized autonomous tracking and state estimation problems. We formulated the problem as a generalized L_2-penalized dynamic group Lasso minimization problem. The problem can be solved using batch methods when the number of time steps is moderate. For the case with a large number of time steps, new KS-mADMM, GN-IEKS-mADMM, and LM-IEKS-mADMM methods were developed. We also proved the convergence of the proposed methods. We applied the developed methods to simulated tracking, real-world tracking, and audio signal restoration problems, where methods resulted in improved localization and estimation performance and significantly reduced computation load.

A disadvantage of the smoothing-and-splitting methods is that although the methods significantly improve the tracking and estimation performance, their reliability depends on user-defined penalty parameters [e.g., the parameter γ in (6)]. See [46] and [47] for further details on choosing the appropriate values of the parameters. The use of adaptive penalty parameters may improve the performance in dynamic systems, even while stronger conditions of convergence need to be guaranteed [60]. It would be interesting to develop fully automated solvers with adaptive parameters. The convergence and the convergence rate of our methods are based on the Bayesian smoothers and ADMM, and we have established the convergence rate of the convex case. Possible future work includes discussing the convergence rate for nonconvex variants.

Although we only consider the autonomous tracking and state estimation problems in this article, it is possible to apply our framework to a wide class of control problems. For example, in linear optimal control problems, we could introduce splitting variables to decompose the nonsmooth terms and then use the Riccati equations to compute the subproblems arising in the optimal control problems [61]. In cooperative control of multiple target systems [62]–[64], we can consider a reformulation of dynamic models of group targets into classes with different characteristics. Based on the framework, we address the subproblems in implementing optimization-based methods such as receding horizon methods [63]. The proposed framework can be extended to other variable splitting methods [48] as well as other recursive smoothers [1]. Future work also includes developing other variants, for example, sigma-point-based variable splitting methods.

Appendix A
Proof of Lemma 1

For proving Lemma 1, we define $\boldsymbol{\zeta} = \begin{bmatrix}\mathbf{x} & \mathbf{w}\end{bmatrix}^\top$ and then write variables $\mathbf{x}$ and $\mathbf{w}$ into the function $\Xi(\boldsymbol{\zeta})$, which is

$$\Xi(\boldsymbol{\zeta}) = \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x} - \mathbf{e}\|^2_{\mathbf{R}^{-1}} + \frac{1}{2}\|\mathbf{m} - \boldsymbol{\Phi}\mathbf{x} - \mathbf{b}\|^2_{\mathbf{Q}^{-1}} + \mu\|\mathbf{w}\|_2. \tag{36}$$

Using the optimality conditions of the subproblems in (7), we can write

$$\begin{aligned}&\Xi(\boldsymbol{\zeta}) - \Xi\left(\boldsymbol{\zeta}^{(k+1)}\right) + (\boldsymbol{\zeta} - \boldsymbol{\zeta}^{(k+1)})^\top \\ &\quad\times\left(\begin{bmatrix}\boldsymbol{\Phi} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}^\top \boldsymbol{\eta}^{(k)} + \gamma\begin{bmatrix}\boldsymbol{\Phi} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}^\top\right.\\ &\qquad\left.\times\left(\boldsymbol{\zeta}^{(k+1)} - \begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}\mathbf{v}^{(k)} - \begin{bmatrix}\mathbf{d}\\ \mathbf{0}\end{bmatrix}\right)\right) \geq \mathbf{0}\end{aligned} \tag{37a}$$

$$\begin{aligned}&\left(\mathbf{v} - \mathbf{v}^{(k+1)}\right)^\top\left(-\begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}^\top \boldsymbol{\eta}^{(k)} - \gamma\begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}^\top\right.\\ &\qquad\left.\times\left(\boldsymbol{\zeta}^{(k+1)} - \begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}\mathbf{v}^{(k+1)} - \begin{bmatrix}\mathbf{d}\\ \mathbf{0}\end{bmatrix}\right)\right) \geq \mathbf{0}\end{aligned} \tag{37b}$$

$$\begin{aligned}&\left(\boldsymbol{\eta} - \boldsymbol{\eta}^{(k+1)}\right)^\top\left(-\left(\begin{bmatrix}\boldsymbol{\Phi} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\mathbf{z}^{(k+1)} - \begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}\mathbf{v}^{(k+1)} - \begin{bmatrix}\mathbf{d}\\ \mathbf{0}\end{bmatrix}\right)\right.\\ &\qquad\left.+\frac{1}{\gamma}(\boldsymbol{\eta}^{(k+1)} - \boldsymbol{\eta}^{(k)})\right) \geq \mathbf{0}.\end{aligned} \tag{37c}$$

For simplicity of notation, we also define

$$\begin{aligned}\boldsymbol{\xi} &= \begin{bmatrix}\boldsymbol{\zeta} & \mathbf{v} & \boldsymbol{\eta}\end{bmatrix}^\top\\ F(\boldsymbol{\xi}) &= \begin{bmatrix}\begin{bmatrix}\boldsymbol{\Phi} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}^\top \boldsymbol{\eta}^{(k+1)}\\ -\begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}^\top \boldsymbol{\eta}^{(k+1)}\\ -\begin{bmatrix}\boldsymbol{\Phi} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\boldsymbol{\zeta}^{(k+1)} + \begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}\mathbf{v}^{(k+1)} + \begin{bmatrix}\mathbf{d}\\ \mathbf{0}\end{bmatrix}\end{bmatrix}.\end{aligned} \tag{38}$$

We group all the variables $\boldsymbol{\zeta}$, $\mathbf{v}$, and $\boldsymbol{\eta}$ into a single vector $\boldsymbol{\xi}$ and then rewrite (37) as follows:

$$\Xi(\boldsymbol{\zeta}) - \Xi\left(\boldsymbol{\zeta}^{(k+1)}\right) + \left[\boldsymbol{\xi} - \boldsymbol{\xi}^{(k+1)}\right]^\top$$

$$\left(F(\boldsymbol{\xi}) + \gamma \begin{bmatrix} \begin{bmatrix}\boldsymbol{\Phi} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}^\top \\ -\begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}^\top \\ \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}\left(\mathbf{v}^{(k)} - \mathbf{v}^{(k+1)}\right) + \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \gamma\begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}^\top & \mathbf{0}\\ \mathbf{0} & \frac{1}{\gamma}\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{v}^{(k+1)}-\mathbf{v}^{(k)}\\ \boldsymbol{\eta}^{(k+1)}-\boldsymbol{\eta}^{(k)}\end{bmatrix}\right) \geq \mathbf{0}$$

$$\begin{aligned}&\Xi(\boldsymbol{\zeta}) - \Xi(\boldsymbol{\zeta}^{(k+1)}) + \left[\boldsymbol{\xi} - \boldsymbol{\xi}^{(k+1)}\right]^\top F(\boldsymbol{\xi})\\ &\quad + \gamma\left[\boldsymbol{\xi} - \boldsymbol{\xi}^{(k+1)}\right]^\top \begin{bmatrix} \begin{bmatrix}\boldsymbol{\Phi} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}^\top \\ -\begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}^\top \\ \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}\left(\mathbf{v}^{(k)} - \mathbf{v}^{(k+1)}\right)\\ &\geq \begin{bmatrix}\mathbf{v}-\mathbf{v}^{(k+1)}\\ \boldsymbol{\eta}-\boldsymbol{\eta}^{(k+1)}\end{bmatrix}^\top \begin{bmatrix}\gamma\begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}^\top & \mathbf{0}\\ \mathbf{0} & \frac{1}{\gamma}\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{v}^{(k+1)}-\mathbf{v}^{(k)}\\ \boldsymbol{\eta}^{(k+1)}-\boldsymbol{\eta}^{(k)}\end{bmatrix}.\end{aligned} \tag{39}$$

Since the mapping $F(\boldsymbol{\xi})$ is affine with a skew-symmetric matrix, it is monotonic [52]. Then, we have the inequality

$$\begin{aligned}&\Xi\left(\boldsymbol{\zeta}^{(k+1)}\right) - \Xi(\boldsymbol{\zeta}^\star) + \left(\boldsymbol{\xi}^{(k+1)} - \boldsymbol{\xi}^\star\right)^\top F\left(\boldsymbol{\xi}^{(k+1)}\right)\\ &\quad\geq \Xi\left(\boldsymbol{\zeta}^{(k+1)}\right) - \Xi(\boldsymbol{\zeta}^\star) + \left(\boldsymbol{\xi}^{(k+1)} - \boldsymbol{\xi}^\star\right)^\top F(\boldsymbol{\xi}^\star) \geq \mathbf{0}.\end{aligned} \tag{40}$$

Meanwhile, using (7d), the inequality can be written as

$$\left(\boldsymbol{\eta}^{(k)} - \boldsymbol{\eta}^{(k+1)}\right)^\top \left(-\begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}\right)\left(\mathbf{v}^{(k)} - \mathbf{v}^{(k+1)}\right) \geq \mathbf{0}. \tag{41}$$

Combining (40) and (41), we can derive (39) as

$$\begin{aligned}&\left(\begin{bmatrix}\mathbf{v}^{(k+1)}\\ \boldsymbol{\eta}^{(k+1)}\end{bmatrix} - \begin{bmatrix}\mathbf{v}^\star\\ \boldsymbol{\eta}^\star\end{bmatrix}\right)^\top \begin{bmatrix}\gamma\mathbf{I} + \mathbf{G}^\top\mathbf{G} & \mathbf{0}\\ \mathbf{0} & \frac{1}{\gamma}\mathbf{I}\end{bmatrix}\\ &\quad\times\left(\begin{bmatrix}\mathbf{v}^{(k)}\\ \boldsymbol{\eta}^{(k)}\end{bmatrix} - \begin{bmatrix}\mathbf{v}^{(k+1)}\\ \boldsymbol{\eta}^{(k+1)}\end{bmatrix}\right)\\ &\quad\geq \gamma\left[\boldsymbol{\xi}^{(k+1)} - \boldsymbol{\xi}^\star\right]^\top \begin{bmatrix} \begin{bmatrix}\boldsymbol{\Phi} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}^\top \\ -\begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}^\top \\ \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}\left(\mathbf{v}^{(k)} - \mathbf{v}^{(k+1)}\right)\\ &\quad\geq (\boldsymbol{\eta}^{(k)} - \boldsymbol{\eta}^{(k+1)})^\top\left(-\begin{bmatrix}\mathbf{I}\\ \mathbf{G}\end{bmatrix}\right)\left(\mathbf{v}^{(k)} - \mathbf{v}^{(k+1)}\right) \geq \mathbf{0}.\end{aligned} \tag{42}$$

Let $\boldsymbol{\Omega} = \begin{bmatrix}\gamma\mathbf{I} + \mathbf{G}^\top\mathbf{G} & \mathbf{0}\\ \mathbf{0} & (1/\gamma)\mathbf{I}\end{bmatrix}$. We then conclude that

$$\begin{aligned}&\left\|\begin{bmatrix}\mathbf{v}^{(k)}\\ \boldsymbol{\eta}^{(k)}\end{bmatrix} - \begin{bmatrix}\mathbf{v}^\star\\ \boldsymbol{\eta}^\star\end{bmatrix}\right\|_{\boldsymbol{\Omega}}^2\\ &\quad= \left\|\begin{bmatrix}\mathbf{v}^{(k+1)}\\ \boldsymbol{\eta}^{(k+1)}\end{bmatrix} - \begin{bmatrix}\mathbf{v}^\star\\ \boldsymbol{\eta}^\star\end{bmatrix}\right\|_{\boldsymbol{\Omega}}^2 + \left\|\begin{bmatrix}\mathbf{v}^{(k)}\\ \boldsymbol{\eta}^{(k)}\end{bmatrix} - \begin{bmatrix}\mathbf{v}^{(k+1)}\\ \boldsymbol{\eta}^{(k+1)}\end{bmatrix}\right\|_{\boldsymbol{\Omega}}^2\\ &\qquad + 2\left(\begin{bmatrix}\mathbf{v}^{(k+1)}\\ \boldsymbol{\eta}^{(k+1)}\end{bmatrix} - \begin{bmatrix}\mathbf{v}^\star\\ \boldsymbol{\eta}^\star\end{bmatrix}\right)^\top \boldsymbol{\Omega}\left(\begin{bmatrix}\mathbf{v}^{(k)}\\ \boldsymbol{\eta}^{(k)}\end{bmatrix} - \begin{bmatrix}\mathbf{v}^{(k+1)}\\ \boldsymbol{\eta}^{(k+1)}\end{bmatrix}\right)\\ &\quad\geq \left\|\begin{bmatrix}\mathbf{v}^{(k+1)}\\ \boldsymbol{\eta}^{(k+1)}\end{bmatrix} - \begin{bmatrix}\mathbf{v}^\star\\ \boldsymbol{\eta}^\star\end{bmatrix}\right\|_{\boldsymbol{\Omega}}^2 + \left\|\begin{bmatrix}\mathbf{v}^{(k)}\\ \boldsymbol{\eta}^{(k)}\end{bmatrix} - \begin{bmatrix}\mathbf{v}^{(k+1)}\\ \boldsymbol{\eta}^{(k+1)}\end{bmatrix}\right\|_{\boldsymbol{\Omega}}^2.\end{aligned} \tag{43}$$

Appendix B
Proof of Lemma 2

To simplify the notation, we replace the $(k+1)$: iteration with the $+$:th iteration and drop the iteration counter k in this proof. Due to the strongly amenability, $s(\mathbf{x})$ is prox-regular with a positive constant M. Now, we compute

$$\begin{aligned}&\mathcal{L}_\gamma(\mathbf{x}, \mathbf{w}, \mathbf{v}; \boldsymbol{\eta}) - \mathcal{L}_\gamma\left(\mathbf{x}^{(+)}, \mathbf{w}^{(+)}, \mathbf{v}; \boldsymbol{\eta}\right)\\ &\quad= s(\mathbf{x}) - s\left(\mathbf{x}^{(+)}\right) + \left\langle\overline{\boldsymbol{\eta}}, \boldsymbol{\Phi}\mathbf{x}^{(+)} - \boldsymbol{\Phi}\mathbf{x}\right\rangle\\ &\qquad + \left\langle\gamma\left(\boldsymbol{\Phi}\mathbf{x}^{(+)} - \mathbf{d} - \mathbf{v}\right), \boldsymbol{\Phi}\mathbf{x}^{(+)} - \boldsymbol{\Phi}\mathbf{x}\right\rangle\\ &\qquad + \frac{\gamma}{2}\left\|\boldsymbol{\Phi}\mathbf{x}^{(+)} - \boldsymbol{\Phi}\mathbf{x}\right\|^2\\ &\qquad + g(\mathbf{w}) - g\left(\mathbf{w}^{(+)}\right) + \left\langle\underline{\boldsymbol{\eta}}, \mathbf{w}^{(+)} - \mathbf{w}\right\rangle\\ &\qquad + \left\langle\gamma\left(\mathbf{w}^{(+)} - \mathbf{G}\mathbf{v}\right), \mathbf{w}^{(+)} - \mathbf{w}\right\rangle + \frac{\gamma}{2}\left\|\mathbf{w}^{(+)} - \mathbf{w}\right\|^2\\ &\quad> \frac{\gamma\delta_+\left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\right) - M}{2}\left\|\mathbf{x}^{(+)} - \mathbf{x}\right\|^2 + \frac{\gamma}{2}\left\|\mathbf{w}^{(+)} - \mathbf{w}\right\|^2\end{aligned} \tag{44}$$

where $\boldsymbol{\eta} = \mathrm{vec}(\overline{\boldsymbol{\eta}}, \underline{\boldsymbol{\eta}})$. We then have

$$\begin{aligned}&\mathcal{L}_\gamma\left(\mathbf{x}^{(+)}, \mathbf{w}^{(+)}, \mathbf{v}^{(+)}; \boldsymbol{\eta}^{(+)}\right) - \mathcal{L}_\gamma(\mathbf{x}, \mathbf{w}, \mathbf{v}; \boldsymbol{\eta})\\ &\quad< \frac{1}{\gamma}\|\boldsymbol{\eta}^{(+)} - \boldsymbol{\eta}\|^2 + \frac{M - \gamma\delta_+\left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\right)}{2}\left\|\mathbf{x}^{(+)} - \mathbf{x}\right\|^2\\ &\qquad + \frac{\gamma}{2}\left\|\mathbf{w}^{(+)} - \mathbf{w}\right\|^2\end{aligned} \tag{45}$$

which will be non-negative provided when $\gamma > (M/[\delta_+(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})]) > 0$ is satisfied. In particular, when $\boldsymbol{\Phi} = \mathbf{I}$, $\delta_+(\boldsymbol{\Phi}^\top\boldsymbol{\Phi}) = 1$.

In our case, $\mathcal{L}_\gamma(\mathbf{x}^{(k)}, \mathbf{w}^{(k)}, \mathbf{v}^{(k)}; \boldsymbol{\eta}^{(k)})$ is upper bounded by $\mathcal{L}_\gamma(\mathbf{x}^{(0)}, \mathbf{w}^{(0)}, \mathbf{v}^{(0)}; \boldsymbol{\eta}^{(0)})$, and is also lower bounded by $\mathcal{L}_\gamma(\mathbf{x}^{(k)}, \mathbf{w}^{(k)}, \mathbf{v}^{(k)}; \boldsymbol{\eta}^{(k)}) \geq s(\mathbf{x}^{(k)}) + \sum_{t=1}^{T}\sum_{g=1}^{N_g}\mu\|\mathbf{w}_{g,t}\|_2$. Thus, we obtain the conclusion.

Appendix C
Proof of Lemma 3

We use the smallest nonzero eigenvalue of $\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$ and $\mathbf{S}^{-1}$ to yield the inequality

$$\left\|\mathbf{J}_\theta^\top\mathbf{J}_\theta\left(\mathbf{x}^{(i)}\right)\right\| \geq \max\left\{\gamma\delta_+\left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\right), \lambda^{(i)}\delta_+\left(\left[\mathbf{S}^{(i)}\right]^{-1}\right)\right\} \tag{46}$$

where

$$\mathbf{J}_\theta = \left[\mathbf{R}^{-\frac{1}{2}}\mathbf{J}_h(\mathbf{x}) \quad \mathbf{Q}^{-\frac{1}{2}}\mathbf{J}_a(\mathbf{x}) \quad \gamma^{\frac{1}{2}}\boldsymbol{\Phi} \quad \lambda^{\frac{1}{2}}\mathbf{S}^{-\frac{1}{2}}\right]^\top.$$

We then have

$$\begin{aligned}\left\|\mathbf{x}^{(i+1)} - \mathbf{x}^\star\right\| &\leq \frac{M}{2}\left\|\left[\mathbf{J}_\theta^\top\mathbf{J}_\theta\left(\mathbf{x}^{(i)}\right)\right]^{-1}\right\|\left\|\mathbf{x}^{(i)} - \mathbf{x}^\star\right\|^2\\ &\quad + \left\|\left[\mathbf{J}_\theta^\top\mathbf{J}_\theta\left(\mathbf{x}^{(i)}\right)\right]^{-1}\mathbf{H}_\theta\left(\mathbf{x}^{(i)}\right)\right\|\left\|\mathbf{x}^{(i)} - \mathbf{x}^\star\right\|.\end{aligned} \tag{47}$$

When $\|\mathbf{H}_\theta(\mathbf{x})\| \leq \kappa$ and $\kappa \to 0$, the convergence is quadratic. The linear convergence can be established when the inequality $\|[\mathbf{J}_\theta^\top \mathbf{J}_\theta(\mathbf{x}^{(i)})]^{-1}\mathbf{H}_\theta(\mathbf{x}^{(i)})\| \leq \kappa/\max\{\gamma\delta_+(\mathbf{\Phi}^\top\mathbf{\Phi}), \lambda^{(i)}\delta_+([\mathbf{S}^{(i)}]^{-1})\} < 1$ is satisfied.

REFERENCES

[1] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[2] F. Dornaika and B. Raducanu, "Three-dimensional face pose detection and tracking using monocular videos: Tool and application," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 4, pp. 935–944, Aug. 2009.

[3] G. Rogez, J. Rihan, J. J. Guerrero, and C. Orrite, "Monocular 3-D gait tracking in surveillance scenes," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 894–909, Jun. 2014.

[4] S. J. Godsill and P. J. Rayner, *Digital Audio Restoration: A Statistical Model Based Approach*. London, U.K.: Springer, 1998.

[5] B. I. Ahmad, J. K. Murphy, P. Langdon, S. Godsill, R. Hardy, and L. Skrypchuk, "Intent inference for hand pointing gesture-based interactions in vehicles," *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 878–889, Apr. 2016.

[6] Y. B. Shalom, X. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation*. New York, NY, USA: Wiley, 2001.

[7] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic system," *AIAA J.*, vol. 3, no. 8, pp. 1445–1450, Aug. 1965.

[8] B. Bell, "The iterated Kalman smoother as a Gauss–Newton method," *SIAM J. Optim.*, vol. 4, no. 3, pp. 626–636, Aug. 1994.

[9] T. D. Barfoot, *State Estimation for Robotics*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[10] S. Särkkä and L. Svensson, "Levenberg–Marquardt and line–search extended Kalman smoothers," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Barcelona, Spain, May 2020, pp. 5875–5879.

[11] S. Särkkä, "Unscented Rauch–Tung–Striebel smoother," *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 845–849, Apr. 2008.

[12] A. F. García-Fernández, L. Svensson, and S. Särkkä, "Iterated posterior linearization smoother," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 2056–2063, Apr. 2017.

[13] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Stat. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.

[14] A. Doucet, N. De Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York, NY, USA: Springer, 2001.

[15] S. Seifzadeh, B. Khaleghi, and F. Karray, "Distributed soft-data-constrained multi-model particle filter," *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 384–394, Mar. 2015.

[16] R. Gao, F. Tronarp, and S. Särkkä, "Iterated extended Kalman smoother-based variable splitting for L1-regularized state estimation," *IEEE Trans. Signal Process.*, vol. 67, no. 19, pp. 5078–5092, Oct. 2019.

[17] J. Tisdale, Z. Kim, and J. K. Hedrick, "Autonomous UAV path planning and estimation," *IEEE Robot. Autom. Mag.*, vol. 16, no. 2, pp. 35–42, Jun. 2009.

[18] C. Févotte, B. Torrésani, L. Daudet, and S. J. Godsill, "Sparse linear regression with structured priors and application to denoising of musical audio," *IEEE Trans. Audio Speech Lang. Process.*, vol. 16, no. 1, pp. 174–185, Jan. 2008.

[19] P. J. Wolfe, S. J. Godsill, and W.-J. Ng, "Bayesian variable selection and regularization for time–frequency surface estimation," *J. Royal Stat. Soc. B, Stat. Methodol.*, vol. 66, no. 3, pp. 575–589, 2004.

[20] Z. Zhao, S. Särkkä, and A. B. Rad, "Spectro-temporal ECG analysis for atrial fibrillation detection," in *Proc. 28th Int. Workshop Mach. Learn. Signal Process.*, Aalborg, Denmark, Sep. 2018, pp. 1–6.

[21] R. Gao, F. Tronarp, Z. Zhao, and S. Särkkä, "Regularized state estimation and parameter learning via augmented Lagrangian Kalman smoother method," in *Proc. 29th Int. Workshop Mach. Learn. Signal Process.*, Pittsburgh, PA, USA, Oct. 2019, pp. 1–6.

[22] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Royal Stat. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.

[23] L. Meier, S. van de Geer, and P. Buhlmann, "The group LASSO for logistic regression," *J. Royal Stat. Soc. B*, vol. 70, no. 1, pp. 53–71, 2008.

[24] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation-based noise removal algorithms," *Physica D Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, 1992.

[25] R. Gao, F. Tronarp, and S. Särkkä, "Combined analysis-L1 and total variation ADMM with applications to MEG brain imaging and signal reconstruction," in *Proc. IEEE 26th Eur. Signal Process. Conf.*, Rome, Italy, Feb. 2018, pp. 1930–1934.

[26] A. Carmi, P. Gurfil, and D. Kanevsky, "Methods for sparse signal recovery using Kalman filtering pseudo-measurement norms and quasi-norms," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2405–2409, Apr. 2010.

[27] N. Vaswani, "Kalman filtered compressed sensing," in *Proc. IEEE Int. Conf. Image Process.*, San Diego, CA, USA, Dec. 2008, pp. 893–896.

[28] N. Vaswani and J. Zhan, "Recursive recovery of sparse signal sequences from compressive measurements: A review," *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3523–3549, Jul. 2016.

[29] J. H. Yoo, S. H. Lim, and B. Shim, and J. W. Choi, "Estimation of dynamically varying support of sparse signals via sequential Monte-Carlo method," *IEEE Trans. Signal Process.*, vol. 68, pp. 4135–4145, Jul. 2020.

[30] A. Y. Aravkin, B. M. Bell, J. V. Burke, and G. Pillonetto, "An L1 Laplace robust Kalman smoother," *IEEE Trans. Autom. Control*, vol. 56, no. 12, pp. 2898–2911, Dec. 2011.

[31] M. R. O'Shaughnessy, M. A. Davenport, and C. J. Rozell, "Sparse Bayesian learning with dynamic filtering for inference of time-varying sparse signals," *IEEE Trans. Signal Process.*, vol. 68, pp. 388–403, Jan. 2020.

[32] A. Aravkin, J. V. Burke, L. Ljung, A. Lozano, and G. Pillonetto, "Generalized Kalman smoothing: Modeling and algorithms," *Automatica*, vol. 86, pp. 63–86, Dec. 2017.

[33] A. S. Charles, A. Balavoine, and C. J. Rozell, "Dynamic filtering of time-varying sparse signals via L1 minimization," *IEEE Trans. Signal Process.*, vol. 64, no. 21, pp. 5644–5656, Nov. 2016.

[34] A. Simonetto and E. Dall'Anese, "Prediction-correction algorithms for time-varying constrained optimization," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5481–5494, Oct. 2017.

[35] S. Farahmand, G. Giannakis, and D. Angelosante, "Doubly robust smoothing of dynamical processes via outlier sparsity constraints," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4529–4543, Oct. 2011.

[36] R. Gao and S. Särkkä, "Augmented sigma-point Lagrangian splitting method for sparse nonlinear state estimation," in *Proc. IEEE 28th Eur. Signal Process. Conf.*, Amsterdam, The Netherlands, Jan. 2021, pp. 2090–2094.

[37] B. S. Park and S. J. Yoo, "An error transformation approach for connectivity-preserving and collision-avoiding formation tracking of networked uncertain underactuated surface vessels," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2955–2966, Aug. 2019.

[38] Z. Peng, D. Wang, T. Li, and M. Han, "Output-feedback cooperative formation maneuvering of autonomous surface vehicles with connectivity preservation and collision avoidance," *IEEE Trans. Cybern.*, vol. 6, no. 50, pp. 2527–2535, Jun. 2020.

[39] W. Caharija *et al.*, "Integral line-of-sight guidance and control of underactuated marine vehicles: Theory, simulations, and experiments," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 5, pp. 1623–1642, Sep. 2016.

[40] R. Rout, R. Cui, and W. Yan, "Sideslip-compensated guidance-based adaptive neural control of marine surface vessels," *IEEE Trans. Cybern.*, early access, Oct. 15, 2020, doi: 10.1109/TCYB.2020.3023162.

[41] L. Wu, Y. Wang, and S. Pan, "Exploiting attribute correlations: A novel trace Lasso-based weakly supervised dictionary learning method," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4497–4508, Dec. 2017.

[42] T. Bai, Y.-F. Li, and X. Zhou, "Learning local appearances with sparse representation for robust and fast visual tracking," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 663–675, Apr. 2015.

[43] D. Du, L. Zhang, H. Lu, X. Mei, and X. Li, "Discriminative hash tracking with group sparsity," *IEEE Trans. Cybern.*, vol. 46, no. 8, pp. 1914–1925, Aug. 2016.

[44] M. Xu and M. H, "Adaptive elastic echo state network for multivariate time series prediction," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2173–2183, Oct. 2016.

[45] X. Cheng, Y. Zhang, J. Cui, and L. Zhou, "Object tracking via temporal consistency dictionary learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 4, pp. 628–638, Apr. 2017.

[46] S. J. Wright and J. Nocedal, *Numerical Optimization*. New York, NY, USA: Springer, 2006.

[47] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[48] R. Glowinski, S. J. Osher, and W. Yin, *Splitting Methods in Communication, Imaging, Science, and Engineering*. Cham, Switzerland: Springer, 2016.

[49] A. T. Puig, A. Wiesel, G. Fleury, and A. O. Hero, “Multidimensional shrinkage-thresholding operator and group LASSO penalties,” *IEEE Signal Process. Lett.*, vol. 18, no. 6, pp. 363–366, Jun. 2011.

[50] J. J. Moré, “The Levenberg–Marquardt algorithm: Implementation and theory,” in *Proc. Number. Anal.*, 1978, pp. 105–116.

[51] R. Gao, “Recursive smoother type variable splitting methods for state estimation,” Ph.D. dissertation, Dept. Electr. Eng. Autom., Aalto Univ., Espoo, Finland, 2020.

[52] B. He and X. Yuan, “On the $O(1/t)$ convergence rate of the Douglas-Rachford alternating direction method,” *SIAM J. Numer. Anal.*, vol. 50, no. 2, pp. 700–709, 2012.

[53] Y. Wang, W. Yin, and J. Zeng, “Global convergence of ADMM in nonconvex nonsmooth optimization,” *J. Sci. Comput.*, vol. 78, no. 1, pp. 29–63, Jan. 2019.

[54] M. Hong, M. Razaviyayn, and Z.-Q. Luo, “Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems,” *SIAM J. Optim.*, vol. 26, no. 1, pp. 337–364, Jan. 2016.

[55] R. T. Rockafellar and R. Wets, *Variational Analysis*. Berlin, Germany: Springer, 1998.

[56] R. A. Poliquin and R. T. Rockafellar, “Prox-regular functions in variational analysis,” *Trans. Amer. Math. Soc.*, vol. 348, no. 5, pp. 1805–1838, May 1996.

[57] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[58] J. Hartikainen and S. Särkkä, “Sequential inference for latent force models,” in *Proc. 27th Conf. Uncertainty Artif. Intell.*, Jul. 2011, pp. 311–318.

[59] P. Kabal, “An examination and interpretation of ITU-R BS.1387: Perceptual evaluation of audio quality,” Dept. Electr. Comput. Eng., McGill Univ., Montreal, QC, Canada, Rep., 2003.

[60] H. Zhou, X. Zeng, and Y. Hong, “Adaptive exact penalty design for constrained distributed optimization,” *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4661–4667, Nov. 2019.

[61] D. Axehill and M. Morari, “An alternative use of the Riccati recursion for efficient optimization,” *Syst. Control Lett.*, vol. 61, no. 1, pp. 37–40, Jan. 2012.

[62] S. Wen and G. Guo, “Sampled-data control for connected vehicles with Markovian switching topologies and communication delay,” *IEEE Trans. Veh. Technol.*, vol. 21, no. 7, pp. 2930–2942, Jul. 2020.

[63] G. Guo and P. Zhang, “Asymptotic stabilization of USVs with actuator dead-zones and yaw constraints based on fixed-time disturbance observer,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 302–316, Jan. 2020.

[64] G. Guo and Q. Wang, “Fuel-efficient en route speed planning and tracking control of truck platoons,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 3091–3103, Aug. 2019.

Rui Gao received the Master of Science degree in computer software and theory from Northeastern University, Shenyang, China, in 2012, and the Doctor of Science (Tech.) degree in automation, systems and control engineering from Aalto University, Espoo, Finland, in 2020.

She is a Postdoctoral Researcher with the Department of Electrical Engineering and Automation, Aalto University. Her research interests include state estimation, sparsity, convex optimization, and image fusion.

Simo Särkkä (Senior Member, IEEE) received the Master of Science (Tech.) degree (with Distinction) in engineering physics and mathematics, and the Doctor of Science (Tech.) degree (with Distinction) in electrical and communications engineering from the Helsinki University of Technology, Espoo, Finland, in 2000 and 2006, respectively.

He is currently an Associate Professor with Aalto University and an Adjunct Professor with the Tampere University of Technology, Tampere, Finland, and the Lappeenranta University of Technology, Lappeenranta, Finland. He has authored or coauthored over 150 peer-reviewed scientific articles and his books *Bayesian Filtering and Smoothing* and *Applied Stochastic Differential Equations* along with the Chinese translation of the former were recently published via Cambridge University Press. His research interests are in multisensor data processing systems with applications in location sensing, health and medical technology, machine learning, inverse problems, and brain imaging.

Dr. Särkkä serves as a Senior Area Editor for IEEE SIGNAL PROCESSING LETTERS.

Rubén Claveria-Vega received the B.Eng. degree in computer science and the M.Sc. degree in electrical engineering from the University of Chile, Santiago, Chile, in 2015 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Signal Processing and Communications Laboratory, Department of Engineering, University of Cambridge, Cambridge, U.K.

His research interests include statistical signal analysis, Bayesian modeling and methodology, and audio and music processing.

Simon Godsill received the B.A. and Ph.D. degrees from the University of Cambridge, Cambridge, U.K., in 1989 and 1994, respectively.

He is a Professor of Statistical Signal Processing with the Engineering Department, University of Cambridge, where he is also a Professorial Fellow and a Tutor with Corpus Christi College Cambridge. He coordinates an Active Research Group in Signal Inference and its Applications. He is the Head of the Information Engineering Division, Cambridge. His group specializes in Bayesian computational methodology, multiple object tracking and intentionality analysis, audio and music processing, Lévy processes, and time-series modeling. A particular methodological theme over recent years has been the development of novel techniques for optimal Bayesian filtering and smoothing, using Sequential Monte Carlo or particle filtering methods. He has authored or coauthored journals, books, and international conference proceedings, and has given a number of high-profile invited and plenary addresses at conferences. In 1998, he authored or coauthored *Digital Audio Restoration* with Prof. P. Rayner. He was the Theme Leader in Tracking and Reasoning over time for the U.K's Data and Information Fusion Defence Technology Centre and the Principal Investigator on many grants funded by the EU, EPSRC, QinetiQ, General Dynamics, MOD, Microsoft U.K., Citibank, Mastercard, Google, DSO Singapore, Huawei, and Jaguar Land Rover. From 2009 to 2010, he was a Co-Organizer of an 18-month research program in Sequential Monte Carlo Methods with the SAMSI Institute, Durham, NC, USA. In 2014, he co-organized a research programme with the Isaac Newton Institute on Sequential Monte Carlo methods. He continues to be the Director of CEDAR Audio Ltd. (which has received numerous accolades over the years, including a technical Oscar).

Prof. Godsill was an Associate Editor for IEEE TRANSACTION ON SIGNAL PROCESSING and the *Bayesian Analysis*. He was a Founding Staff Member in 1988.