Optimization viewpoint on Kalman smoothing, with applications to robust and sparse estimation.

Aleksandr Y. Aravkin and James V. Burke and Gianluigi Pillonetto

Abstract In this chapter, we present the optimization formulation of the Kalman filtering and smoothing problems, and use this perspective to develop a variety of extensions and applications. We first formulate classic Kalman smoothing as a least squares problem, highlight special structure, and show that the classic filtering and smoothing algorithms are equivalent to a particular algorithm for solving this problem. Once this equivalence is established, we present extensions of Kalman smoothing to systems with nonlinear process and measurement models, systems with linear and nonlinear inequality constraints, systems with outliers in the measurements or sudden changes in the state, and systems where the sparsity of the state sequence must be accounted for. All extensions preserve the computational efficiency of the classic algorithms, and most of the extensions are illustrated with numerical examples, which are part of an open source Kalman smoothing Matlab/Octave package.

1 Introduction

Kalman filtering and smoothing methods form a broad category of computational algorithms used for inference on noisy dynamical systems. Over the last fifty years,

Aleksandr Y. Aravkin
IBM T.J. Watson Research Center
Yorktown Heights, NY, 10598
e-mail: saravkin@us.ibm.com

James V. Burke
Department of Mathematics,
University of Washington, Seattle, WA
e-mail: jvburke@uw.edu

Gianluigi Pillonetto
Control and Dynamic Systems Department of Information Engineering,
University of Padova, Padova, Italy,
e-mail: giapi@dei.unipd.it

these algorithms have become a gold standard in a range of applications, including space exploration, missile guidance systems, general tracking and navigation, and weather prediction. In 2009, Rudolf Kalman received the National Medal of Science from President Obama for the invention of the Kalman filter. Numerous books and papers have been written on these methods and their extensions, addressing modifications for use in nonlinear systems, smoothing data over time intervals, improving algorithm robustness to bad measurements, and many other topics.
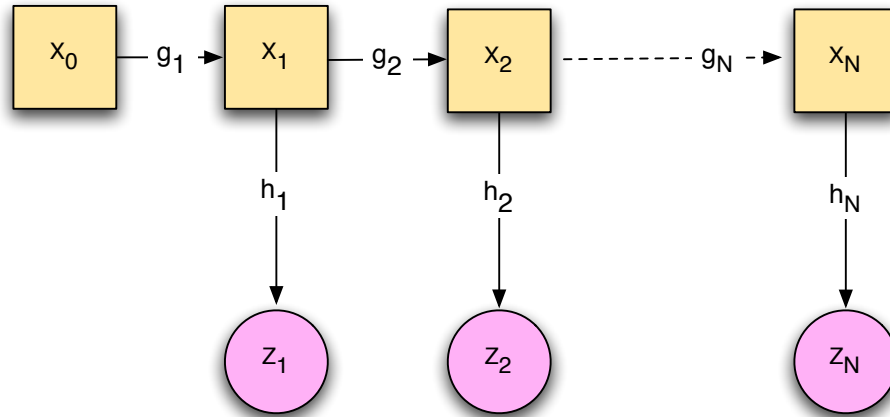
Fig. 1 Dynamic systems amenable to Kalman smoothing methods.

The classic Kalman filter [29] is almost always presented as a set of recursive equations, and the classic Rauch-Tung-Striebel (RTS) fixed-interval smoother [42] is typically formulated as two coupled Kalman filters. An elegant derivation based on projections onto spaces spanned by random variables can be found in [2]. In this chapter, we use the terms 'Kalman filter' and 'Kalman smoother' much more broadly, **including any method of inference on any dynamical system fitting the graphical representation of Figure 1.** Specific mathematical extensions we consider include

- Nonlinear process and measurement models.
- Inequality state space constraints.
- Different statistical models for process and measurement errors.
- Sparsity constraints.

We also show numerous applications of these extensions.

The key to designing tractable inference methods for the above applications is an optimization viewpoint, which we develop in the classic Kalman smoothing case and then use to formulate and solve *all* of the above extensions. Though it has been known for many years that the Kalman filter provides the maximum *a posteriori*

estimate for linear systems subject to Gaussian noise, the optimization perspective underlying this idea has not been fully deployed across engineering applications. Notably, several groups (starting in 1977) have discovered and used variants of this perspective to implement extensions to Kalman filtering and smoothing, including singular filtering ([39, 33, 40]), robust smoothing ([22, 7]), nonlinear smoothing with inequality state space constraints ([9, 11]), and sparse Kalman smoothing [1].

We focus exclusively on smoothing here, leaving online applications of these ideas to future work (see [41] for an example of using a smoother for an online application). We start by presenting the classic RTS smoothing algorithm in Section 2, and show that the well-known recursive equations are really an algorithm to solve a least squares system with special structure. Once this is clear, it becomes much easier to discuss novel extensions, since as long as special structure is preserved, their computational cost is on par with the classic smoother (or, put another way, the classic smoothing equations are viewed as a particular way to solve key subproblems in the extended approaches).

In the subsequent sections, we build novel extensions, briefly review theory, discuss the special structure, and present numerical examples for a variety of applications. In Section 3, we formulate the problem for smoothing with nonlinear process and measurement models, and show how to solve it. In Section 4, we show how state space constraints can be incorporated, and the resulting problem solved using interior point techniques. In Section 5, we review two recent Kalman smoothing formulations that are highly robust to measurement errors. Finally, in Section 6, we review recent work in sparse Kalman smoothing, and show how sparsity can be incorporated into the other extensions. We end the chapter with discussion in Section 7.

2 Optimization Formulation and RTS Smoother

2.1 Probabilistic model

The model corresponding to Figure 1 is specified as follows:

$$
\begin{aligned}
\mathbf{x_1} &= g_1(x_0) + \mathbf{w_1}, \\
\mathbf{x_k} &= g_k(\mathbf{x_{k-1}}) + \mathbf{w_k} \ k = 2, \ldots, N, \\
\mathbf{z_k} &= h_k(\mathbf{x_k}) + \mathbf{v_k} \quad k = 1, \ldots, N,
\end{aligned}
\tag{1}
$$

where $\mathbf{w_k}$, $\mathbf{v_k}$ are mutually independent random variables with known positive definite covariance matrices Q_k and R_k, respectively. We have $\mathbf{x_k}, \mathbf{w_k} \in \mathbb{R}^n$, and $\mathbf{z_k}, \mathbf{v_k} \in \mathbb{R}^{m(k)}$, so measurement dimensions can vary between time points. The classic case is obtained by making the following assumptions:

1. x_0 is known, and g_k, h_k are known *linear* functions, which we denote by

$$
g_k(x_{k-1}) = G_k x_{k-1} \quad h_k(x_k) = H_k x_k
\tag{2}
$$

where $G_k \in \mathbb{R}^{n \times n}$ and $H_k \in \mathbb{R}^{m(k) \times n}$,

2. $\mathbf{w_k}$, $\mathbf{v_k}$ are mutually independent *Gaussian* random variables.

In later sections, we will show how to relax these classic assumptions, and what gains can be achieved once they are relaxed. In this section, we will formulate estimation of the *entire* state sequence, $x_1, x_2, \ldots, x_N$, as an optimization problem, and show how the RTS smoother solves it.

2.2 *Maximum* a posteriori *formulation*

To begin, we formulate the maximum *a posteriori* (MAP) problem under linear and Gaussian assumptions. Using Bayes' theorem, we have

$$
\begin{aligned}
P\left(\{x_k\} \middle| \{z_k\}\right) &\propto P\left(\{z_k\} \middle| \{x_k\}\right) P\left(\{x_k\}\right) \\
&= \prod_{k=1}^{N} P(\{v_k\}) P(\{w_k\}) \\
&\propto \prod_{k=1}^{N} \exp\left(-\frac{1}{2}(z_k - H_k x_k)^\top R_k^{-1}(z_k - H_k x_k)\right. \\
&\qquad\qquad \left. -\frac{1}{2}(x_k - G_k x_{k-1})^\top Q_k^{-1}(x_k - G_k x_{k-1})\right).
\end{aligned}
\tag{3}
$$

A better (equivalent) formulation to (3) is minimizing its negative log posterior:

$$
\min_{\{x_k\}} f(\{x_k\}) := \sum_{k=1}^{N} \frac{1}{2}(z_k - H_k x_k)^\top R_k^{-1}(z_k - H_k x_k) + \frac{1}{2}(x_k - G_k x_{k-1})^\top Q_k^{-1}(x_k - G_k x_{k-1}).
\tag{4}
$$

To simplify the problem, we now introduce data structures that capture the entire state sequence, measurement sequence, covariance matrices, and initial conditions.

Given a sequence of column vectors $\{u_k\}$ and matrices $\{T_k\}$ we use the notation

$$
\text{vec}(\{u_k\}) = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}, \ \text{diag}(\{T_k\}) = \begin{bmatrix} T_1 & 0 & \cdots & 0 \\ 0 & T_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & T_N \end{bmatrix}.
$$

We now make the following definitions:

$$
\begin{aligned}
R &= \mathrm{diag}(\{R_k\}) & x &= \mathrm{vec}(\{x_k\}) \\
Q &= \mathrm{diag}(\{Q_k\}) & w &= \mathrm{vec}(\{g_0, 0, \ldots, 0\}) \\
H &= \mathrm{diag}(\{H_k\}) & z &= \mathrm{vec}(\{z_1, z_2, \ldots, z_N\})
\end{aligned}
\qquad
G = \begin{bmatrix} I & 0 & & \\ -G_2 & I & \ddots & \\ & \ddots & \ddots & 0 \\ & & -G_N & I \end{bmatrix},
$$

(5)

where $g_0 := g_1(x_0) = G_1 x_0$.

With definitions in (5), problem (4) can be written

$$
\min_x f(x) = \frac{1}{2}\|Hx - z\|_{R^{-1}}^2 + \frac{1}{2}\|Gx - w\|_{Q^{-1}}^2 \,,
$$

(6)

where $\|a\|_M^2 = a^\top M a$. We knew the MAP was a least squares problem already, but now the structure is fully transparent. In fact, we can write down the closed form solution by taking the gradient of (6) and setting it equal to 0:

$$
\begin{aligned}
0 &= H^\top R^{-1}(Hx - z) + G^\top Q^{-1}(Gx - w) \\
&= (H^\top R^{-1}H + G^\top Q^{-1}G)x - H^\top R^{-1}z - G^\top Q^{-1}w \,.
\end{aligned}
$$

The smoothing estimate is therefore given by solving the linear system

$$
(H^\top R^{-1}H + G^\top Q^{-1}G)x = H^\top R^{-1}z + G^\top Q^{-1}w \,.
$$

(7)

2.3 Special subproblem structure

The linear system in (7) has a very special structure: it is a symmetric positive definite block tridiagonal matrix. This can be immediately observed from the fact that both G and Q are positive definite. To be specific, it is given by

$$
C = (H^\top R^{-1}H + G^\top Q^{-1}G) = \begin{bmatrix} C_1 & A_2^\mathsf{T} & 0 & \\ A_2 & C_2 & A_3^\mathsf{T} & 0 \\ 0 & \ddots & \ddots & \ddots \\ & & 0 & A_N & C_N \end{bmatrix},
$$

(8)

with $A_k \in \mathbb{R}^{n \times n}$ and $C_k \in \mathbb{R}^{n \times n}$ defined as follows:

$$
\begin{aligned}
A_k &= -Q_k^{-1}G_k \,, \\
C_k &= Q_k^{-1} + G_{k+1}^\top Q_{k+1}^{-1}G_{k+1} + H_k^\top R_k^{-1}H_k \,.
\end{aligned}
$$

(9)

The special structure of the matrix C in (8) can be exploited to solve the linear system equivalent to the Kalman smoother. While a structure-agnostic matrix in-

version scheme has complexity $O(n^3 N^3)$, exploiting the block tridiagonal structure reduces this complexity to $O(n^3 N)$.

A straightforward algorithm for solving any symmetric positive definite block tridiagonal linear system is given in [10]. We review it here, since it is essential to build the connection to the standard viewpoint of the RTS smoother.

2.4 Block tridiagonal (BT) algorithm

Suppose for $k = 1, \dots, N$, $c_k \in \mathbf{R}^{n \times n}$, $e_k \in \mathbf{R}^{n \times \ell}$, $r_k \in \mathbf{B}^{n \times \ell}$, and for $k = 2, \dots, N$, $a_k \in \mathbf{R}^{n \times n}$. We define the corresponding block tridiagonal system of equations

$$
\begin{pmatrix}
c_1 & a_2^{\mathrm{T}} & 0 & \cdots & 0 \\
a_2 & c_2 & & & \vdots \\
\vdots & & \ddots & & 0 \\
0 & & a_{N-1} & c_{N-1} & a_N^{\mathrm{T}} \\
0 & \cdots & 0 & a_N & c_N
\end{pmatrix}
\begin{pmatrix}
e_1 \\
e_2 \\
\vdots \\
e_{N-1} \\
e_N
\end{pmatrix}
=
\begin{pmatrix}
r_1 \\
r_2 \\
\vdots \\
r_{N-1} \\
r_N
\end{pmatrix}
\tag{10}
$$

The following algorithm for (10) is given in [10, Algorithm 4].

Algorithm 2.1 *The inputs to this algorithm are* $\{a_k\}$*,* $\{c_k\}$*, and* $\{r_k\}$*. The output is a sequence* $\{e_k\}$ *that solves equation (10).*

1. Set $d_1 = c_1$ and $s_1 = r_1$.
2. For $k = 2, \dots, N$, set $d_k = c_k - a_k^{\mathrm{T}} d_{k-1}^{-1} a_k$, $\quad s_k = r_k - a_k^{\mathrm{T}} d_{k-1}^{-1} s_{k-1}$.
3. Set $e_N = d_N^{-1} s_N$.
4. For $k = N - 1, \dots, 1$, set $e_k = d_k^{-1}(s_k - a_{k+1} e_{k+1})$.

Note that after the first two steps of Algorithm 2.1, we have arrived at a linear system equivalent to (10) but upper triangular:

$$
\begin{pmatrix}
d_1 & a_2^{\mathrm{T}} & 0 & \cdots & 0 \\
0 & d_2 & & & \vdots \\
\vdots & & \ddots & & 0 \\
0 & & 0 & d_{N-1} & a_N^{\mathrm{T}} \\
0 & \cdots & 0 & 0 & d_N
\end{pmatrix}
\begin{pmatrix}
e_1 \\
e_2 \\
\vdots \\
e_{N-1} \\
e_N
\end{pmatrix}
=
\begin{pmatrix}
s_1 \\
s_2 \\
\vdots \\
s_{N-1} \\
s_N
\end{pmatrix}
\tag{11}
$$

The last two steps of the algorithm then simply back-solve for the e_k.

2.5 Equivalence of Algorithm (2.1) to Kalman Filter and RTS Smoother

Looking at the very first block, we now substitute in the Kalman data structures (9) into step 2 of Algorithm 2.1:

$$d_2 = c_2 - a_2^\mathsf{T} d_1^{-1} a_2$$

$$= Q_2^{-1} - \left(Q_2^{-1}G_2\right)^\mathsf{T} \left(\underbrace{Q_1^{-1} + H_1^\mathsf{T} R_1^{-1} H_1}_{P_{1|1}^{-1}} + G_2^\mathsf{T} Q_2^{-1} G_2\right)^{-1} \underbrace{\left(Q_2^{-1}G_2\right) + H_2^\mathsf{T} R_2^{-1} H_2 + G_3^\mathsf{T} Q_3^{-1} G_3}_{P_{2|1}^{-1}}$$

$$\underbrace{}_{P_{2|2}^{-1}}$$

(12)

These relationships can be seen quickly from [5, Theorem 2.2.7]. The matrices $P_{k|k}$, $P_{k|k-1}$ are common to the Kalman filter framework: they represent covariances of the state at time k given the the measurements $\{z_1, \ldots, z_k\}$, and the covariance of the a priori state estimate at time k given measurements $\{z_1, \ldots, z_{k-1}\}$, respectively.

From the above computation, we see that

$$d_2 = P_{2|2}^{-1} + G_3^\mathsf{T} Q_3^{-1} G_3 \ .$$

By induction, it is easy to see that in fact

$$d_k = P_{k|k}^{-1} + G_{k+1}^\mathsf{T} Q_{k+1}^{-1} G_{k+1} \ .$$

We can play the same game with s_k. Keeping in mind that $r = H^\mathsf{T} R^{-1} z + G^\mathsf{T} Q^{-1} w$, we have

$$s_2 = r_2 - a_2^\mathsf{T} d_1^{-1} r_1$$

$$= H_2^\mathsf{T} R_2^{-1} z_2 + \left(Q_2^{-1}G_2\right)^\mathsf{T} \left(\underbrace{Q_1^{-1} + H_1^\mathsf{T} R_1^{-1} H_1}_{P_{1|1}^{-1}} + G_2^\mathsf{T} Q_2^{-1} G_2\right)^{-1} \underbrace{\left(H_1^\mathsf{T} R_1^{-1} z_1 + G_1^\mathsf{T} P_{0|0}^{-1} x_0\right)}_{a_{2|1}}$$

$$\underbrace{}_{a_{2|2}}$$

(13)

These relationships also follow from [5, Theorem 2.2.7]. The quantities $a_{2|1}$ and $a_{2|2}$ are from the information filtering literature, and are less commonly known: they are preconditioned estimates

$$a_{k|k} = P_{k|k}^{-1} x_k \ , \quad a_{k|k-1} = P_{k|k-1}^{-1} x_{k|k-1} \ . \tag{14}$$

Again, by induction we have precisely that $s_k = a_{k|k}$.

When you put all of this together, you see that step 3 of Algorithm 2.1 is given by

$$e_N = d_N^{-1} s_N = \left(P_{N|N}^{-1} + 0\right)^{-1} P_{N|N}^{-1} x_{k|k} = x_{k|k} \ , \tag{15}$$

so in fact e_N is the Kalman filter estimate (and the RTS smoother estimate) for time point N.

Step 4 of Algorithm 2.1 then implements the backward Kalman filter, computing the smoothed estimates $x_{k|N}$ by back-substitution. **Therefore the RTS smoother is Algorithm 2.1 applied to** (7).

The consequences are profound — instead of working with the kinds expressions seen in (13) and (12), we can think at a high level, focusing on (6), and simply using Algorithm 2.1 (or variants) as a subroutine. As will become apparent, the key to all extensions is preserving the block tridiagonal structure in the subproblems, so that Algorithm 2.1 can be used.

2.6 Numerical Example: Tracking a Smooth Signal
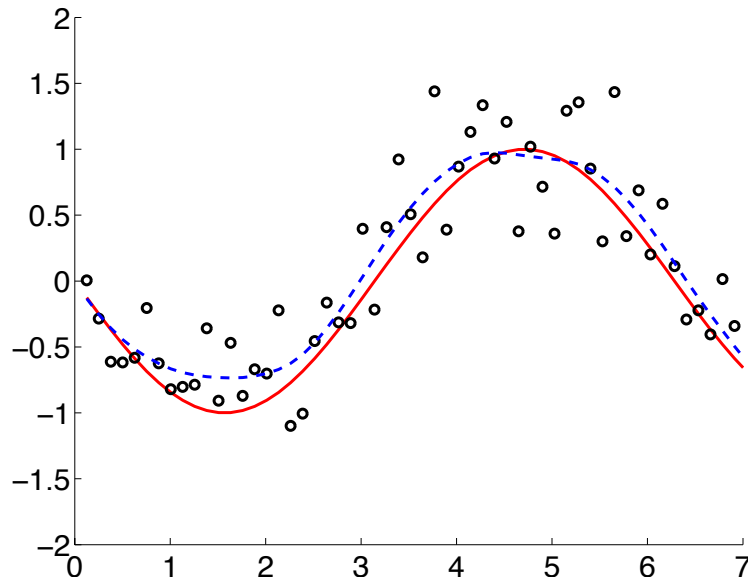
Fig. 2 Tracking a smooth signal (sine wave) using a generic linear process model (16) and direct (noisy) measurements (18). Red solid line is true signal, blue dashed line is Kalman (RTS) smoother estimate. Measurements are displayed as circles.

In this example, we focus on a very useful and simple model: the process model for a *smooth* signal. Smooth signals arise in a range of applications: physics-based models, biological data, and financial data all have some inherent smoothness.

A surprisingly versatile technique for modeling *any* such process is to treat it as integrated Brownian motion. We illustrate on a scalar time series x. We introduce a new derivative state $\dot{x}$, with process model $\dot{x}_{k+1} = \dot{x}_k + \dot{w}_k$, and then model the

signal x or interest as $x_{k+1} = x_k + \dot{x}_k \Delta t + w_k$. Thus we obtain an augmented (2D) state with process model

$$\begin{bmatrix} \dot{x}_{k+1} \\ x_{k+1} \end{bmatrix} = \begin{bmatrix} I & 0 \\ \Delta t & I \end{bmatrix} \begin{bmatrix} \dot{x}_k \\ x_k \end{bmatrix} + \begin{bmatrix} \dot{w}_k \\ w_k \end{bmatrix} . \tag{16}$$

Using a well-known connection to stochastic differential equations (see [26, 38, 11]) we use covariance matrix

$$Q_k = \sigma^2 \begin{bmatrix} \Delta t & \Delta t^2/2 \\ \Delta t^2/2 & \Delta t^3/3 \end{bmatrix} . \tag{17}$$

Model equations (16) and (17) can be applied as a process model for any smooth process. For our numerical example, we take direct measurements of the sin function, which is very smooth. Our measurement model therefore is

$$z_k = H_k x_k + v_k , \quad H_k = \begin{bmatrix} 0 & 1 \end{bmatrix} . \tag{18}$$

The resulting fit is shown in Figure 2.6. The measurements guide the estimate to the true smooth time series, giving very nice results. The figure was generated using the `ckbs` package [6], specifically using the example file `affine_ok.m`. Measurement errors were generated using $R_k = .35^2$, and this value was given to the smoother. The σ^2 in (17) was taken to be 1. The program and example are available for download from COIN-OR.

3 Nonlinear Process and Measurement Models

In the previous section, we have shown that when g_k and h_k in model (1) are linear, and $\mathbf{v_k}, \mathbf{w_k}$ are Gaussian, then the smoother is equivalent to solving a least squares problem (6). We have also shown that the filter estimates appear as intermediate results when one uses Algorithm 2.1 to solve the problem.

In this section, we turn to the case where g_k and h_k are nonlinear. We first formulate the smoothing problem as a maximum *a posteriori* (MAP) problem, and show that it is a nonlinear least squares (NLLS) problem. To set up for later sections, we also introduce the broader class of *convex composite* problems.

We then review the standard Gauss-Newton method in the broader context of convex composite models, and show that when applied to the NLLS problem, each iteration is equivalent to solving (6), and therefore to a full execution of the RTS smoother. We also show how to use a simple line search to guarantee convergence of the method to a local optimum of the MAP problem.

This powerful approach, known for at least 20 years [21, 12, 9], is rarely used in practice; instead practitioners favor the EKF or the UKF [18, 28], neither of which converge to a (local) MAP solution. MAP approaches work very well for a broad range of applications, and it is not clear why one would throw away an efficient

MAP solver in favor of another scheme. To our knowledge, the optimization (MAP) approach has never been included in a performance comparison of 'cutting edge' methods, such as [34]. While such a comparison is not in the scope of this work, we lay the foundation by providing a straightforward exposition of the optimization approach and a reproducible numerical illustration (with publicly available code) for smoothing the Van Der Pol oscillator, a well known problem where the process model is a nonlinear ODE.

3.1 Nonlinear Smoother Formulation and Structure

In order to develop a notation analogous to (6), we define functions $g : \mathbb{R}^{nN} \to \mathbb{R}^{n(N+1)}$ and $h : \mathbb{R}^{nN} \to \mathbb{R}^{M}$, with $M = \sum_{k} m_{k}$, from components g_{k} and h_{k} as follows.

$$g(x) = \begin{bmatrix} x_1 \\ x_2 - g_2(x_1) \\ \vdots \\ x_N - g_N(x_{N-1}) \end{bmatrix} , \quad h(x) = \begin{bmatrix} h_1(x_1) \\ h_2(x_2) \\ \vdots \\ h_N(x_N) \end{bmatrix} . \tag{1}$$

With this notation, the MAP problem, obtained exactly as in Section 2.2, is given by

$$\min_{x} f(x) = \frac{1}{2} \|g(x) - w\|_{Q^{-1}}^{2} + \frac{1}{2} \|h(x) - z\|_{R^{-1}}^{2} , \tag{2}$$

where z and w are exactly as in (5), so that z is the *entire* vector of measurements, and w contains the initial estimate $g_1(x_0)$ in the first n entries, and zeros in the remaining $n(N-1)$ entries.

We have formulated the nonlinear smoothing problem as a nonlinear least-squares (NLLS) problem — compare (2) with (6). We take this opportunity to note that NLLS problems are a special example of a more general structure. Objective (2) may be written as a composition of a convex function ρ with a smooth function F:

$$f(x) = \rho(F(x)) , \tag{3}$$

where

$$\rho \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \frac{1}{2} \|y_1\|_{Q^{-1}}^{2} + \frac{1}{2} \|y_2\|_{R^{-1}}^{2} , \quad F(x) = \begin{bmatrix} g(x) - w \\ h(x) - z \end{bmatrix} . \tag{4}$$

As we show in the next sub-section, problems of general form (3) can be solved using the Gauss-Newton method, which is typically associated specifically with NLLS problems. Presenting the Gauss-Newton right away in the more general setting will make it easier to understand extensions in the following sections of the chapter.

3.2 Gauss-Newton Method for Convex Composite Models

The Gauss-Newton method can be used to solve problems of the form (3), and it uses a very simple strategy: **iteratively linearizing the smooth function** F [15]. More specifically, the Gauss-Newton method is an iterative method of the form

$$x^{\nu+1} = x^\nu + \gamma^\nu d^\nu \,, \tag{5}$$

where d^ν is the Gauss-Newton search direction, and γ^ν is a scalar that guarantees

$$f(x^{\nu+1}) < f(x^\nu) \,. \tag{6}$$

The direction d^ν is obtained by solving the subproblem

$$d^\nu = \arg\min_d \tilde{f}(d) := \rho\left(F(x^\nu) + \nabla F(x^\nu)^\top d\right) \,. \tag{7}$$

We then set

$$\tilde{\Delta}f(x^\nu) = \tilde{f}(d^\nu) - f(x^\nu).$$

By [15, Lemma 2.3, Theorem 3.6],

$$f'(x^\nu; d^\nu) \le \tilde{\Delta}f(x^\nu) \le 0 \,, \tag{8}$$

with equality if and only if x^ν is a first-order stationary point for f. This implies that a suitable stopping criteria for the algorithm is the condition $\Delta f(x^\nu) \sim 0$. Moreover, x^ν is not a first-order stationary point for f, then the direction d^ν is a direction of strict descent for f at x^ν.

Once the direction d^ν is obtained with $\tilde{\Delta}f(x^\nu) < 0$, a step-size γ^ν is obtained by a standard backtracking line-search procedure: pick a values $0 < \lambda < 1$ and $0 < \kappa < 1$ (e.g., $\lambda = 0.5$ and $\kappa = 0.001$) and evaluate $f(x^\nu + \lambda^s d^\nu)$, $s = 0, 1, 2, \ldots$, until

$$f(x^\nu + \lambda^s d^\nu) \le f(x^\nu) + \kappa \lambda^s \tilde{\Delta}f(x^\nu) \tag{9}$$

is satisfied for some $\bar{s}$, then set $\gamma^\nu = \lambda^{\bar{s}}$ and make the GN update (5). The fact that there is a finite value of s for which (9) is satisfied follows from inequality $f'(x^\nu; d^\nu) \le \tilde{\Delta}f(x^\nu) < 0$. The inequality (9) is called the Armijo inequality. A general convergence theory for this algorithm as well as a wide range of others is found in [15]. For the NLLS case, the situation is simple, since ρ is a quadratic, and standard convergence theory is given for example in [27]. However, the more general theory is essential in the later sections.

3.3 Details for Kalman Smoothing

To implement the Gauss-Newton method described above, one must compute the solution d^ν to the Gauss-Newton subproblem (7) for (2). That is, one must compute

$$d^\nu = \arg\min_d \tilde{f}(d) = \frac{1}{2}\|G^\nu d - \underbrace{w - g(x^\nu)}_{w^\nu}\|^2_{Q^{-1}} + \frac{1}{2}\|H^\nu d - \underbrace{z - h(x^\nu)}_{z^\nu}\|^2_{R^{-1}}, \quad (10)$$

where

$$G^\nu = \begin{bmatrix} I & 0 & & \\ -g_2^{(1)}(x_1^\nu) & I & \ddots & \\ & \ddots & \ddots & 0 \\ & & -g_N^{(1)}(x_{N-1}^\nu) & I \end{bmatrix}, \quad H^\nu = \mathrm{diag}\{h_1^{(1)}(x_1), \ldots, h_N^{(1)}(x_N)\}.$$

$$(11)$$

However, the problem (10) has exactly the same structure as (6); a fact that we have emphasized by defining

$$w^\nu := w - g(x^\nu), \quad z^\nu = z - h(x^\nu). \quad (12)$$

Therefore, we can solve it efficiently by using Algorithm 2.1.

The linearization step in (10) should remind the reader of the EKF. Note, however, that the Gauss-Newton method is iterative, and we iterate until convergence to a local minimum of (2). We also linearize along the entire state space sequence x^ν at once in (10), rather than re-linearizing as we make our way through the x_k^ν's.

### 3.4 Numerical Example: Van Der Pol Oscillator

The Van der Pol oscillator is a popular nonlinear process for comparing Kalman filters, see [24] and [30, Section 4.1]. The oscillator is governed by a nonlinear ODE model

$$\dot{X}_1(t) = X_2(t) \quad \text{and} \quad \dot{X}_2(t) = \mu[1 - X_1(t)^2]X_2(t) - X_1(t). \quad (13)$$

In contrast to the linear model (16), which was a *generic* process for a smooth signal, we now take the Euler discretization of (13) to be the specific process model for this situation.

Given $X(t_{k-1}) = x_{k-1}$ the Euler approximation for $X(t_{k-1} + \Delta t)$ is

$$g_k(x_{k-1}) = \begin{pmatrix} x_{1,k-1} + x_{2,k-1}\Delta t \\ x_{2,k-1} + \{\mu[1 - x_{1,k}^2]x_{2,k} - x_{1,k}\}\Delta t \end{pmatrix}. \quad (14)$$
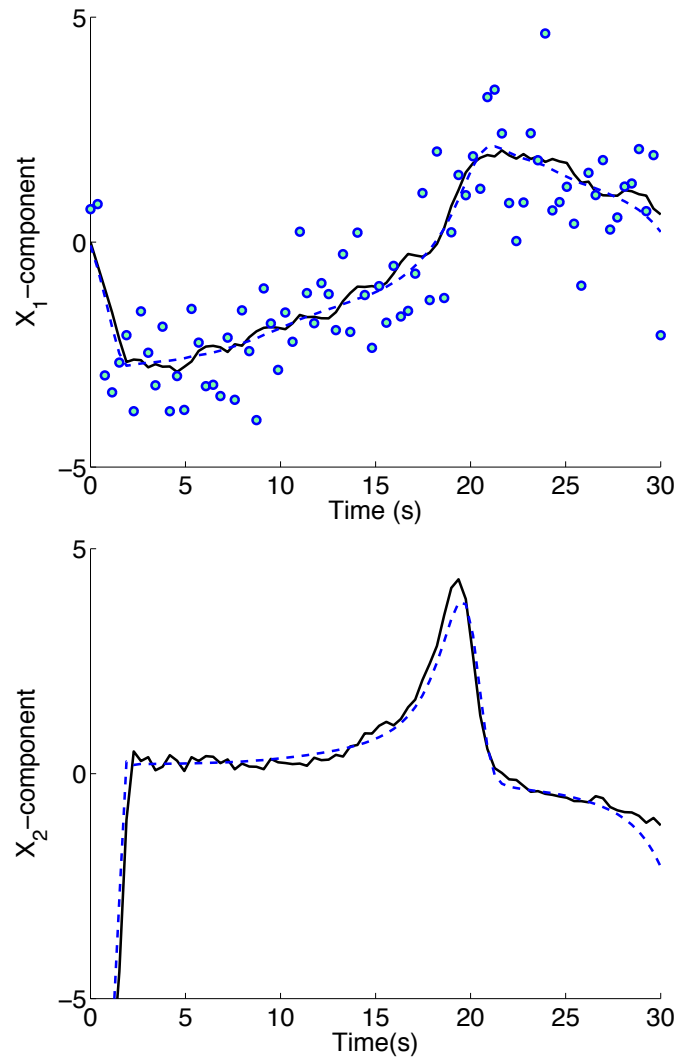
Fig. 3 Tracking the Van Der Pol Osciallator using a nonlinear process model (14) and direct (noisy) measurements (16) of X_1**-component only**. Black solid line is true signal, blue dashed line is nonlinear Kalman smoother estimate. Measurements are displayed as circles.

For the simulation, the 'ground truth' is obtained from a stochastic Euler approximation of the Van der Pol oscillator. To be specific, with $\mu = 2$, $N = 80$ and $\Delta t = 30/N$, the ground truth state vector x_k at time $t_k = k\Delta t$ is given by $x_0 = (0, -0.5)^{\mathrm{T}}$ and for $k = 1, \ldots, N$,

$$x_k = g_k(x_{k-1}) + w_k \,, \tag{15}$$

where $\{w_k\}$ is a realization of independent Gaussian noise with variance 0.01 and g_k is given in (14). Our process model for state transitions is also (16), with $Q_k = 0.01\,I$ for $k > 1$, and so is identical to the model used to simulate the ground truth $\{x_k\}$. Thus, we have precise knowledge of the process that generated the ground truth $\{x_k\}$. The initial state x_0 is imprecisely specified by setting $g_1(x_0) = (0.1, -0.4)^{\mathrm{T}} \neq x_0$ with corresponding variance $Q_1 = 0.1\,I$. For $k = 1, \ldots, N$ noisy measurements z_k direct measurements of *the first component only* were used

$$z_k = x_{1,k} + v_k \,, \tag{16}$$

with $v_k \sim N(0,1)$.

The resulting fit is shown in Figure 3.4. Despite the noisy measurements of only X_1, we are able to get a good fit for both components. The figure was generated using the `ckbs` package [6], see the file `vanderpol_experiment_simple.m`. The program and example are available for download from COIN-OR.

4 State space constraints

In almost every real-world problem, additional prior information is known about the state. In many cases, this information can be represented using *state space constraints*. For example, in tracking physical bodies, we often know (roughly or approximately) the topography of the terrain; this information can be encoded as a simple box constraint on the state. We may also know physical limitations (e.g. maximum acceleration or velocity) of objects being tracked, or hard bounds set by biological or financial systems. These and many other examples can be formulated using state space constraints. The ability to incorporate this information is particularly useful when measurements are inaccurate or far between.

In this section, we first show how to add affine inequality constraints to the affine smoother formulation in Section 2. This requires a novel methodology: *interior point (IP) methods*, an important topic in optimization [49, 32, 37]. IP methods work directly with optimality conditions, so we derive these conditions for the smoothing problem. Rather than review theoretical results about IP methods, we give a general overview and show how they specialize to the linear constrained smoother. The constrained Kalman smoother was originally proposed in [11], but we improve on that work here, and present a simplified algorithm, which is also faster and more numerically stable. We illustrate the algorithm using a numerical example, building on the example in Section 2.

Once the *linear* smoother with *linear* inequality constraints is understood, we review the constrained *nonlinear* smoother (which can have nonlinear process, measurement, and constraint functions). Using [11] and references therein, we show that the constrained nonlinear smoother is iteratively solved using linear constrained smoothing subproblems, analogously to how the nonlinear smoother in Section 3 is iteratively solved using linear smoothing subproblems from Section 2. Because of this hierarchy, the improvements to the affine algorithm immediately carry over to the nonlinear case. We end with a nonlinear constrained numerical example.

4.1 Linear Constrained Formulation

We start with the linear smoothing problem (6), and impose linear inequality constraints on the state space x:

$$B_k x_k \leq b_k \ . \tag{1}$$

By choosing the matrix B_k and b_k appropriately, one can ensure x_k lies in any polyhedral set, since such a set is defined by a finite intersection of hyperplanes. Box constraints, one of the simplest and useful tools for modeling ($l_k \leq x_k \leq u_k$) can be imposed via

$$\begin{bmatrix} I \\ -I \end{bmatrix} x_k \leq \begin{bmatrix} u_k \\ -l_k \end{bmatrix} \ .$$

In order to formulate the problem for the entire state space sequence, we define

$$B = \text{diag}(\{B_k\}) \ , \quad b = \text{vec}(\{b_k\}) \ , \tag{2}$$

and all of the constraints can be written simultaneously as $Bx \leq b$. The constrained optimization problem is now given by

$$\begin{aligned} \min_x f(x) = \quad & \frac{1}{2} \|Hx - z\|_{R^{-1}}^2 + \frac{1}{2} \|Gx - w\|_{Q^{-1}}^2 \\ \text{subject to} \quad & Bx + s = b, \quad s \geq 0 \ . \end{aligned} \tag{3}$$

Note that we have rewritten the inequality constraint as an equality constraint by introducing a new 'slack' variable s.

We derive the Karush-Kuhn-Tucker (KKT) conditions using the Lagrangian formulation. The Lagrangian corresponding to (2) is given by

$$\mathscr{L}(x, u, s) = \frac{1}{2} \|Hx - z\|_{R^{-1}}^2 + \frac{1}{2} \|Gx - w\|_{Q^{-1}}^2 + u^\top (Bx + s - b) \ . \tag{4}$$

The KKT conditions are now obtained by differentiating $\mathscr{L}$ with respect to its arguments. Recall that the gradient of (6) is given by

$$(H^\top R^{-1} H + G^\top Q^{-1} G)x - H^\top R^{-1} z - G^\top Q^{-1} w \ .$$

As in (8) set $C = H^\top R^{-1} H + G^\top Q^{-1} G$, and for convenience set

$$c = H^\top R^{-1} z + G^\top Q^{-1} w \tag{5}$$

The KKT necessary and sufficient conditions for optimality are given by

$$
\begin{aligned}
\nabla_x \mathcal{L} &= Cx + c + B^\top u = 0 \\
\nabla_q \mathcal{L} &= Bx + s - b = 0 \\
u_i s_i &= 0 \quad \forall i \, ; u_i, s_i \geq 0 \, .
\end{aligned}
\tag{6}
$$

The last set of nonlinear equations is known as *complementarity* conditions. In primal-dual interior point methods, the key idea for solving (3) is to successively solve relaxations of the system (6) that converge to a triplet $(\bar{x}, \bar{u}, \bar{s})$ which satisfy (6).

4.2 Interior Point Approach

IP methods work directly to find solutions of (6). They do so by iteratively relaxing the complementarity conditions $u_i s_i = 0$ to $u_i s_i = \mu$ as they drive the relaxation parameter μ to 0. The *relaxed* KKT system is defined by

$$
F_\mu(s, u, x) = \begin{bmatrix} s + Bx - b \\ SU\mathbf{1} - \mu\mathbf{1} \\ Cx + B^T u - c \end{bmatrix} \, . \tag{7}
$$

where S and U are diagonal matrices with s and u on the diagonal, and so the second equation in F_μ implements the relaxation $u_i s_i = \mu$ of (6). Note that the relaxation requires that $\mu_i, s_i > 0$ for all i. Since the solution to (3) is found by driving the KKT system to 0, at every iteration IP methods attempt to drive F_μ to 0 by Newton's method for root finding.

Newton's root finding method solves the linear system

$$
F_\mu^{(1)}(s, u, x) \begin{bmatrix} \Delta s \\ \Delta u \\ \Delta x \end{bmatrix} = -F_\mu(s, u, x) \, . \tag{8}
$$

It is important to see the full details of solving (8) in order to see why it is so effective for constrained Kalman smoothing. The full system is given by

$$
\begin{bmatrix} I & 0 & B \\ U & S & 0 \\ 0 & B^T & C \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta u \\ \Delta x \end{bmatrix} = - \begin{bmatrix} s + Bx - b \\ SU\mathbf{1} - \mu\mathbf{1} \\ Cx + B^T u - c \end{bmatrix} \, . \tag{9}
$$

Applying the row operations

$$\text{row}_2 \leftarrow \quad \text{row}_2 - U\text{row}_1$$
$$\text{row}_3 \leftarrow \text{row}_3 - B^T S^{-1}\text{row}_2 \quad,$$

we obtain the equivalent system

$$\begin{bmatrix} I & 0 & B \\ 0 & S & -UB \\ 0 & 0 & C+B^T S^{-1}UB \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta u \\ \Delta x \end{bmatrix} = - \begin{bmatrix} s+Bx-b \\ -U(Bx-b)-\mu\mathbf{1} \\ Cx+B^T u-c+B^T S^{-1}(U(Bx-b)+\mu\mathbf{1}) \end{bmatrix}.$$

In order to find the update for Δx, we have to solve the system

$$\left(C+B^T S^{-1}UB\right)\Delta x = Cx+B^T u-c+B^T S^{-1}\left(U(Bx-b)+\mu\mathbf{1}\right) \qquad (10)$$

Note the structure of the matrix in the LHS of (10). The matrix C is the same as in (6), so it is positive definite symmetric block tridiagonal. The matrices S^{-1} and U are diagonal, and we always ensure they have only positive elements. The matrices B and $B^\top$ are both block diagonal. Therefore, $C+B^T S^{-1}UB$ has the same structure as C, and we can solve (10) using Algorithm 2.1.

Once we have Δx, the remaining two updates are obtained by back-solving:

$$\Delta u = US^{-1}(B(x+\Delta x)-b) + \frac{\mu}{s} \qquad (11)$$

and

$$\Delta s = -s+b-B(x+\Delta x). \qquad (12)$$

This approach improves the algorithm presented in [11] solely by changing the order of variables and equations in (7). This approach simplifies the derivation while also improving speed and numerical stability.

It remains to explain how μ is taken to 0. There are several strategies, see [49, 32, 37]. For the Kalman smoothing application, we use one of the simplest: for two out of every three iterations μ is aggressively taken to 0 by the update $\mu = \mu/10$; while in the remaining iterations, μ is unchanged. In practice, one seldom needs more than 10 interior point iterations; therefore the constrained linear smoother performs at a constant multiple of work of the linear smoother.

4.3 Two Linear Numerical Examples

In this section, we present to simple examples, both with linear constraints.

Constant Box Constraints

In the first example, we impose box constraints in the example of Section 2.6. Specifically, we take advantage of the fact the state is bounded:
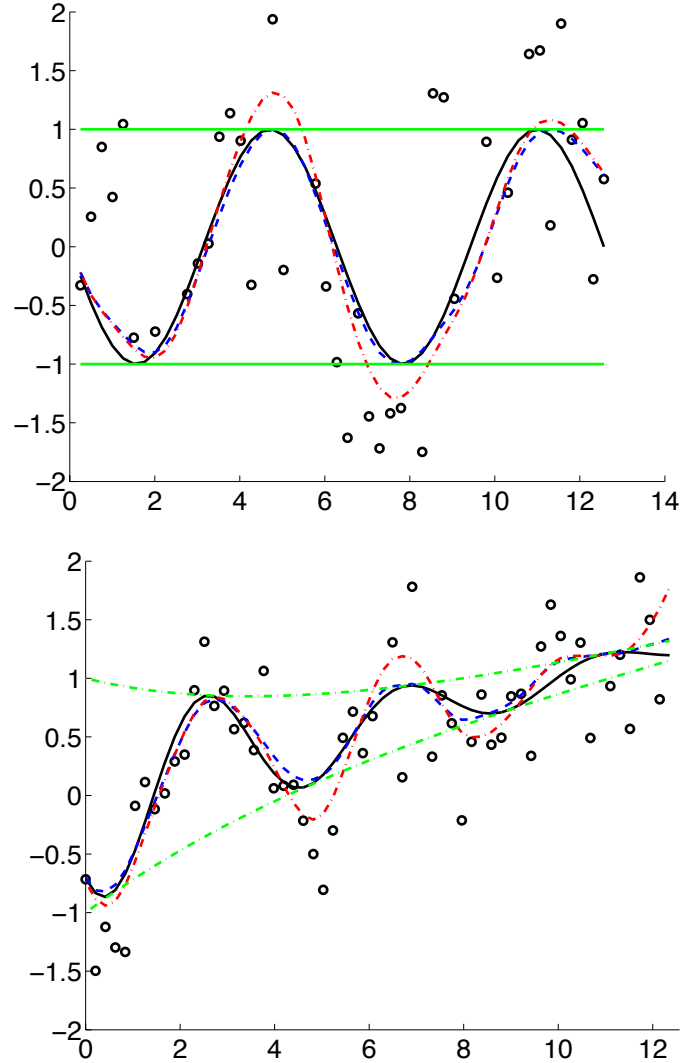
Fig. 4 Two examples of linear constraints. Black solid line is true signal, magenta dash-dot line is unconstrained Kalman smoother, and blue dashed line is the constrained Kalman smoother. Measurements are displayed as circles, and bounds are shown as green horizontal lines. In the left panel, note that performance of the bounded smoother is significantly better around time 4-10 — the unconstrained is fooled by the measurements at times 4 and 8. In the right panel, as the oscillations die down due to damping, the measurement variance remains unchanged, so it becomes much more difficult to track the signal without the bound constraints.

$$\begin{bmatrix} -1 \end{bmatrix} \leq \begin{bmatrix} x \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} \tag{13}$$

We can encode this information in form (1) with

$$B_k = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad b_k = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{14}$$

We contrast the performance of the constrained linear smoother with that of the linear smoother without constraints. To show the advantages of modeling with constraints, we increase the measurement noise in both situations to $\sigma^2 = 1$. The results are show in Figure 4.2. The constrained smoother avoids some of the problems encountered by the unconstrained smoother. Of particular interest are the middle and end parts of the track, where the unconstrained smoother goes far afield because of bad measurement. The constrained smoother is able track portions of the track extremely well, having avoided the bad measurements with the aid of the bound constraints. The figure was generated using the `ckbs` package [6], specifically using the example file `affine_ok_boxC.m`.

Variable Box Constraints

In the second example, we impose time-varying constraints on the state. Specifically, we track an exponentially bounded signal with a linear trend:

$$\exp(-\alpha t)\sin(\beta t) + .1t$$

using the 'smooth signal' process model and direct measurements, as in Section 2.6. The challenge here is that as the oscillations start to die down because of the exponential damping, the variance of the measurements remains the same. We can improve the performance by giving the smoother the exponential damping terms as constraints.

We included the second example to emphasize that 'linearity' of constraints means 'with respect to the state'; in fact, the constraints in the second example are simply box constraints which are time dependent. The second example is no more complicated than the first one for the constrained smoother.

4.4 Nonlinear Constrained Smoother

We now consider the nonlinear constrained smoother, where we allow process functions g_k, measurement functions h_k to be nonlinear, and also allow nonlinear smooth constraints $\xi_k(x_k) \leq b_k$. To be consistent with the notation we use throughout the paper, we define a new function

$$\xi(x) = \begin{bmatrix} \xi_1(x_1) \\ \xi_2(x_2) \\ \vdots \\ \xi_N(x_N) \end{bmatrix} , \tag{15}$$

so that all the constraints can be written simultaneously as $\xi(x) \leq b$.

The problem we would like to solve now is a constrained reformulation of (2)

$$\min_x \quad f(x) = \frac{1}{2}\|g(x) - w\|_{Q^{-1}}^2 + \frac{1}{2}\|h(x) - z\|_{R^{-1}}^2 \tag{16}$$
$$\text{subject to} \quad \xi(x) - b \leq 0 .$$

At this point, we come back to the convex-composite representation described in Section 3.1. The constraint $\xi(x) - b \leq 0$ may be represented using an additional term in the objective function:

$$\delta(\xi(x) - b \mid \mathbb{R}_-) , \tag{17}$$

where $\delta(x \mid C)$ is the convex indicator function:

$$\delta(x \mid C) = \begin{cases} 0 & x \in C \\ \infty & x \notin C \end{cases} . \tag{18}$$

Therefore, the objective (16) can be represented as follows:

$$\begin{aligned} f(x) &= \rho(F(x)) \\ \rho\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} &= \frac{1}{2}\|y_1\|_{Q^{-1}}^2 + \frac{1}{2}\|y_2\|_{R^{-1}}^2 + \delta(y_3 \mid \mathbb{R}_-) \\ F(x) &= \begin{bmatrix} g(x) - w \\ h(x) - z \\ \xi(x) - b \end{bmatrix} . \end{aligned} \tag{19}$$

The approach to nonlinear smoothing in [11] is essentially the Gauss-Newton method described in Section 3.2, applied to (19). In other words, at each iteration ν, the function F is linearized, and the direction finding subproblem is obtained by solving

$$\min_d \quad \frac{1}{2}\|G^\nu d - \underbrace{w - g(x^\nu)}_{w^\nu}\|_{Q^{-1}}^2 + \frac{1}{2}\|H^\nu d - \underbrace{z - h(x^\nu)}_{z^\nu}\|_{R^{-1}}^2 ,$$
$$\text{subject to} \quad B^\nu d \leq \underbrace{b - \xi(x^\nu)}_{b^\nu} , \tag{20}$$

where G^v and H^v are exactly as in (10), $B^v = \nabla_x \xi(x^v)$ is a block diagonal matrix because of the structure of ξ (15), and we have written the indicator function in (19) as an explicit constraint to emphasize the structure of the subproblem.

Note that (20) has exactly the same structure as the linear constrained smoothing problem (3), and therefore can be solved using the interior point approach in the previous section. Because the convex-composite objective (19) is not finite valued (due to the indicator function of the feasible set), to prove convergence of the non-linear smoother, [11] uses results from [14]. We refer the interested reader to [11, Lemma 8, Theorem 9] for theoretical convergence results, and to [11, Algorithm 6] for the full algorithm, including line search details.

Because of the hierarchical dependence of the nonlinear constrained smoother on the linear constrained smoother, the simplified improved approach we presented in Section 4.2 pays off even more in the nonlinear case, where it is used repeatedly as a subroutine.

4.5 Nonlinear Constrained Example
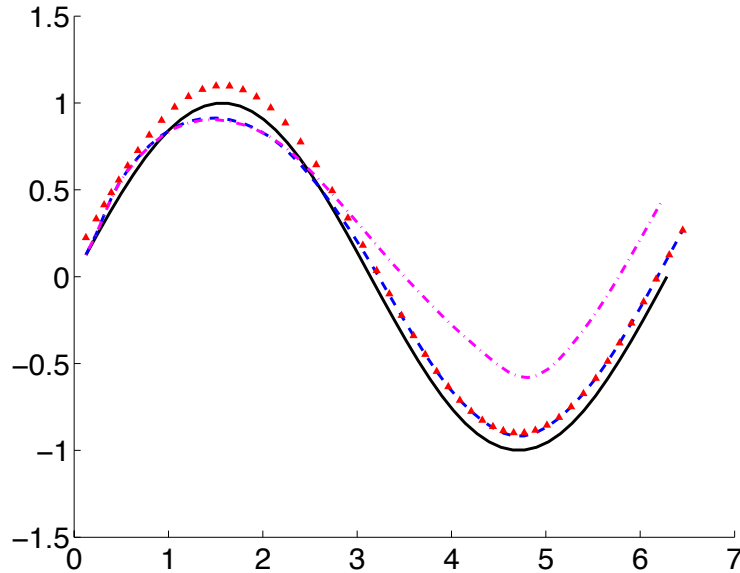
Fig. 5 Smoother results for ship tracking example with linear process model, nonlinear measurement model, and nonlinear constraints (with respect to the state). Black solid line is true state, red triangles denote the constraint, magenta dash-dot line is the unconstrained estimate, and blue dashed line gives the constrained nonlinear smoothed estimate.

The example in this section is reproduced from [11]. Consider the problem of tracking a ship traveling close to shore where we are given distance measurements from two fixed stations to the ship as well as the location of the shoreline. Distance to fixed stations is a nonlinear function, so the measurement model here is nonlinear.

In addition, the corresponding constraint functions $\{f_k\}$ are not affine because the shoreline is not a straight line. For the purpose of simulating the measurements $\{z_k\}$, the ship velocity $[X_1(t), X_3(t)]$ and the ship position $[X_2(t), X_4(t)]$ are given by

$$X(t) = [\, 1 \,,\, t \,,\, -\cos(t) \,,\, 1.3 - \sin(t) \,]^\top$$

Both components of the ship's position are modeled using the smooth signal model in Section 2.6. Therefore we introduce two velocity components, and the process model is given by

$$G_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \Delta t & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \Delta t & 0 \end{bmatrix} , \quad Q_k = \begin{bmatrix} \Delta t & \Delta t^2/2 & 0 & 0 \\ \Delta t^2/2 & \Delta t^3/3 & 0 & 0 \\ 0 & 0 & \Delta t & \Delta t^2/2 \\ 0 & 0 & \Delta t^2/2 & \Delta t^3/3 \end{bmatrix} .$$

The initial state estimate is given by $g_1(x_0) = X(t_1)$ and $Q_1 = 100I_4$ where I_4 is the four by four identity matrix. The measurement variance is constant for this example and is denoted by σ^2. The distance measurements are made from two stationary locations on shore. One is located at $(0,0)$ and the other is located at $(2\pi, 0)$. The measurement model is given by

$$h_k(x_k) = \begin{pmatrix} \sqrt{x_{2,k}^2 + x_{4,k}^2} \\ \sqrt{(x_{2,k} - 2\pi)^2 + x_{4,k}^2} \end{pmatrix} , \quad R_k = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix} .$$

We know that the ship does not cross land, so $X_4(t) \geq 1.25 - \sin[X_2(t)]$. This information is encoded by the constraints

$$\xi_k(x_k) = 1.25 - \sin(x_{2,k}) - x_{4,k} \leq 0 .$$

The initial point for the smoother is $[0,0,0,1]^\top$, which is not feasible. The results are plotted in Figure 4.5. The constrained smoother performs significantly better than the unconstrained smoother in this example. The experiment was done using the `ckbs` program, specifically see `sine_wave_example.m`.

5 Robust Kalman smoothing

In many applications, the probalistic model for the dynamics and/or the observations (1) is not well described by a Gaussian distribution. This occurs in the model for the observations when they are contaminated by outliers, or more generally,

when the measurement noise v_k is heavy tailed [44], and it occurs in the model for the dynamics when tracking systems with rapidly changing dynamics, or jumps in the state values [31]. A *robust* Kalman filter or smoother is one that can obtain an acceptable estimate of the state when Gaussian assumptions are violated, and which continues to perform well when they are not violated.

We show how to accommodate non-Gaussian densities by starting with a simple case of non-Gaussian heavy tailed measurement noise v_k [7]. However, this general approach can be extended to w_k as well. Heavy tailed measurement noise occurs in applications related to glint noise [25], turbulence, asset returns, and sensor failure or machine malfunction. It can also occur in the presence of secondary noise sources or other kinds of data anomalies. Although it is possible to estimate a minimum variance estimate of the state using stochastic simulation methods such as Markov chain Monte-Carlo (MCMC) or particle filters [24, 35], these methods are very computationally intensive, and convergence often relies on heuristic techniques and is highly variable. The approach taken here is very different. It is based on the optimization perspective presented in the previous sections. We develop a method for computing the MAP estimate of the state sequence under the assumption that the observation noise comes from the ℓ_1-Laplace density often used in robust estimation, e.g., see [23, equation 2.3]. As we will see, the resulting optimization problem will again be one of convex composite type allowing us to apply a Gauss-Newton strategy for computing the MAP estimate. Again, the key to a successful computational strategy is the preservation of the underlying tri-diagonal structure.

## 5.1 An ℓ_1-*Laplace Smoother*

For $u \in \mathbb{R}^m$ we use the notation $\|u\|_1$ for the ℓ_1 norm of u; i.e., $\|u\|_1 = |u_1| + \ldots + |u_m|$. The multivariate ℓ_1-Laplace distribution with mean μ and covariance R has the following density:

$$\mathbf{p}(v_k) = \det(2R)^{-1/2} \exp\left[-\sqrt{2} \left\| R^{-1/2}(v_k - \mu) \right\|_1 \right] , \qquad (1)$$

where $R^{1/2}$ denotes a Cholesky factor of the positive definite matrix R; i.e., $R^{1/2}(R^{1/2})^{\mathrm{T}} = R$. One can verify that this is a probability distribution with covariance R using the change of variables $u = R^{-1/2}(v_k - \mu)$. A comparison of the Gaussian and Laplace distributions is displayed in Figure 6. This comparison includes the densities, negative log densities, and influence functions, for both distributions.

5.1.1 Maximum *a posteriori* formulation

Assume that the model for the dynamics and the observations is given by (1), where w_k is assumed to be Gaussian and v_k is modeled by the ℓ_1-Laplace density (1). Under these assumptions, the MAP objective function is given by
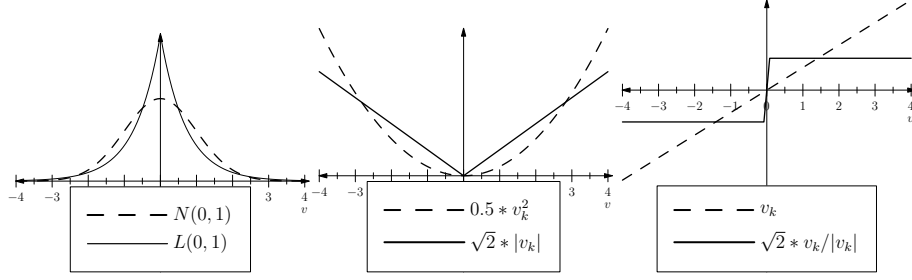
Fig. 6 Gaussian and Laplace Densities, Negative Log Densities, and Influence Functions (for scalar v_k)

$$P\left(\{x_k\}\big|\{z_k\}\right) \propto P\left(\{z_k\}\big|\{x_k\}\right) P\left(\{x_k\}\right)$$

$$= \prod_{k=1}^{N} P(\{v_k\})P(\{w_k\})$$

$$\propto \prod_{k=1}^{N} \exp\left(-\sqrt{2}\left\|R^{-1/2}(z_k - h_k(x_k))\right\|_1 - \frac{1}{2}(x_k - g_k(x_{k-1}))^\top Q_k^{-1}(x_k - g_k(x_{k-1}))\right).$$

(2)

Dropping terms that do not depend on $\{x_k\}$, minimizing this MAP objective with respect to $\{x_k\}$ is equivalent to minimizing

$$f(\{x_k\}) := \sqrt{2}\sum_{k=1}^{N}\left\|R_k^{-1/2}[z_k - h_k(x_k)]\right\|_1 + \frac{1}{2}\sum_{k=1}^{N}[x_k - g_k(x_{k-1})]^\mathsf{T} Q_k^{-1}[x_k - g_k(x_{k-1})],$$

where, as in (1), x_0 is known and $g_0 = g_1(x_0)$. Setting

$$\begin{aligned}
R &= \mathrm{diag}(\{R_k\}) \\
Q &= \mathrm{diag}(\{Q_k\}) \\
x &= \mathrm{vec}(\{x_k\}) \\
w &= \mathrm{vec}(\{g_0, 0, \dots, 0\}) \\
z &= \mathrm{vec}(\{z_1, z_2, \dots, z_N\})
\end{aligned} \qquad , \qquad
g(x) = \begin{bmatrix} x_1 \\ x_2 - g_2(x_1) \\ \vdots \\ x_N - g_N(x_{N-1}) \end{bmatrix}, \quad
h(x) = \begin{bmatrix} h_1(x_1) \\ h_2(x_2) \\ \vdots \\ h_N(x_N) \end{bmatrix},$$

(3)

as in (5) and (1), the MAP estimation problem is equivalent to

$$\underset{x \in \mathbf{R}^{Nn}}{\text{minimize}} \ f(x) = \frac{1}{2}\|g(x) - w\|_{Q^{-1}} + \sqrt{2}\left\|R^{-1/2}(h(x) - z)\right\|_1.$$

(4)

5.1.2 The Convex Composite Structure

The objective in (4) can again be written as a the composition of a convex function ρ with a smooth function F:

$$f(x) = \rho(F(x)) , \tag{5}$$

where

$$\rho \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \frac{1}{2}\|y_1\|_{Q^{-1}}^2 + \sqrt{2}\|R^{-1/2}y_2\|_1 , \quad F(x) = \begin{bmatrix} g(x) - w \\ h(x) - z \end{bmatrix} . \tag{6}$$

Consequently, the generalized Gauss-Newton methodology described in Section 3.2 again applies. That is, given an approximate solution x^ν to (4), we compute a new approximate solution of the form

$$x^{\nu+1} = x^\nu + \gamma^\nu d^\nu,$$

where d^ν solves the subproblem

$$\underset{d \in \mathbb{R}^n}{\text{minimize}}\, \rho(F(x^\nu) + F'(x^\nu)d), \tag{7}$$

and γ^ν is computed using the backtracking line-search procedure described in Section 3.2. Following the pattern described in (10), the subproblem (7), where ρ and F are given in (6), has the form

$$d^\nu = \arg\min_d \tilde{f}(d) = \frac{1}{2}\|G^\nu d - \underbrace{w - g(x^\nu)}_{w^\nu}\|_{Q^{-1}}^2 + \sqrt{2}\|R^{-1/2}(H^\nu d - \underbrace{z - h(x^\nu)}_{z^\nu})\|_1 ,$$
$$\tag{8}$$

where

$$G^\nu = \begin{bmatrix} I & 0 & & \\ -g_2^{(1)}(x_1^\nu) & I & \ddots & \\ & \ddots & \ddots & 0 \\ & & -g_N^{(1)}(x_{N-1}^\nu) & I \end{bmatrix} , \quad H^\nu = \text{diag}\{h_1^{(1)}(x_1),\dots,h_N^{(1)}(x_N)\} . \tag{9}$$

5.1.3 Solving the Subproblem by Interior Point Methods

By (8), the basic subproblem that must be solved takes the form

$$\min_d \frac{1}{2}\|Gd - w\|_{Q^{-1}}^2 + \sqrt{2}\|R^{-1/2}(Hd - z)\|_1 , \tag{10}$$

where, as in (5),

$$
\begin{aligned}
R &= \mathrm{diag}(\{R_k\}) & x &= \mathrm{vec}(\{x_k\}) \\
Q &= \mathrm{diag}(\{Q_k\}) & w &= \mathrm{vec}(\{w_1, w_2, \ldots, w_N\}) \\
H &= \mathrm{diag}(\{H_k\}) & z &= \mathrm{vec}(\{z_1, z_2, \ldots, z_N\})
\end{aligned}
\qquad
G = \begin{bmatrix} I & 0 & & \\ -G_2 & I & \ddots & \\ & \ddots & \ddots & 0 \\ & & -G_N & I \end{bmatrix} .
\tag{11}
$$

Using standard optimization techniques, one can introduce a pair of auxiliary non-negative variables $p^+, p^- \in \mathbb{R}^M$ ($M = \sum_{k=1}^N m(k)$) so that this problem can be rewritten as

$$
\begin{aligned}
&\text{minimize} && \tfrac{1}{2} d^\top C d + c^\top d + \sqrt{\mathbf{2}}^\top (p^+ + p^-) \\
&\text{w.r.t.} && d \in \mathbb{R}^{nN}, \ p^+, p^- \in \mathbb{R}^M \\
&\text{subject to} && Bd + b = p^+ - p^- ,
\end{aligned}
\tag{12}
$$

where

$$
C = G^\top Q^{-1} G = \begin{bmatrix} C_1 & A_2^\top & 0 & \\ A_2 & C_2 & A_3^\top & 0 \\ 0 & \ddots & \ddots & \ddots \\ & 0 & A_N & C_N \end{bmatrix},
\qquad
\begin{aligned}
A_k &= -Q_k^{-1} G_k \\
C_k &= Q_k^{-1} + G_{k+1}^\top Q_{k+1}^{-1} G_{k+1} \\
c &= G^\top w \\
B &= R^{-1/2} H \\
b &= -R^{-1/2} z
\end{aligned}
.
$$

The problem (12) is a convex quadratic program. If we define

$$
F_\mu(p^+, p^-, s^+, s^-, d) = \begin{bmatrix} p^+ - p^- - b - Bd \\ \mathrm{diag}(p^-)\mathrm{diag}(s^-)\mathbf{1} - \mu\mathbf{1} \\ s^+ + s^- - 2\sqrt{\mathbf{2}} \\ \mathrm{diag}(p^+)\mathrm{diag}(s^+)\mathbf{1} - \mu\mathbf{1} \\ Cd + c + B^{\mathrm{T}}(s^- - s^+)/2 \end{bmatrix},
\tag{13}
$$

for $\mu \geq 0$, then the KKT conditions for (12) can be written as

$$
F_0(p^+, p^-, s^+, s^-, d) = 0 .
$$

The set of solutions to $F_\mu(p^+, p^-, s^+, s^-, d) = 0$ for $\mu > 0$ is called the central path. We solve the system for $\mu = 0$ by an interior point strategy which, as described earlier, is a Newton based predictor-corrector method for following the central path as $\mu \downarrow 0$. At each iteration of the interior point method we need to solve a system of the form

$$
F_\mu(p^+, p^-, s^+, s^-, d) + F_\mu'(p^+, p^-, s^+, s^-, d) \begin{bmatrix} \Delta p^+ \\ \Delta p^- \\ \Delta s^+ \\ \Delta s^- \\ \Delta y \end{bmatrix} = 0,
$$

where the vectors p^+, p^-, s^+, and s^- are componentwise strictly positive. Using standard methods of Gaussian elimination (as in Section 4.2), we obtain the solution

$$\begin{aligned}
\Delta y &= [C + B^\mathsf{T} T^{-1} B]^{-1} (\bar{e} + B^\mathsf{T} T^{-1} \bar{f}) \\
\Delta s^- &= T^{-1} B \Delta y - T^{-1} \bar{f} \\
\Delta s^+ &= -\Delta s^- + 2\sqrt{2} - s^+ - s^- \\
\Delta p^- &= \operatorname{diag}(s^-)^{-1} [\tau \mathbf{1} - \operatorname{diag}(p^-) \Delta s^-] - p^- \\
\Delta p^+ &= \Delta p^- + B \Delta y + b + By - p^+ + p^-,
\end{aligned}$$

where

$$\begin{aligned}
\bar{d} &= \tau \mathbf{1}/s^+ - \tau \mathbf{1}/s^- - b - By + p^+ \\
\bar{e} &= B^\mathsf{T} (\sqrt{2} - s^-) - Cy - c \\
\bar{f} &= \bar{d} - \operatorname{diag}(s^+)^{-1} \operatorname{diag}(p^+)(2\sqrt{2} - s^-) \\
T &= \operatorname{diag}(s^+)^{-1} \operatorname{diag}(p^+) + \operatorname{diag}(s^-)^{-1} \operatorname{diag}(p^-) \, .
\end{aligned}$$

Since the matrices T and B are block diagonal, the matrix $B^\top T B$ is also block diagonal. Consequently, the key matrix $C + B^\mathsf{T} T^{-1} B$ has exactly the same form as the block tri-diagonal matrix in (10) with

$$\begin{aligned}
c_k &= Q_k^{-1} + G_{k+1}^\top Q_{k+1}^{-1} G_{k+1} + H_k^\top T_k^{-1} H_k \quad k = 1, \dots, N, \\
a_k &= -Q_k^{-1} G_k \qquad\qquad\qquad\qquad\quad k = 2, \dots, N,
\end{aligned}$$

where $T_k = \operatorname{diag}(s_k^+)^{-1} \operatorname{diag}(p_k^+) + \operatorname{diag}(s_k^-)^{-1} \operatorname{diag}(p_k^-)$. Algorithm 2.1 can be applied to solve this system accurately and stably with $O(n^3 N)$ floating point operations which preserves the efficiency of the classical Kalman Filter algorithm.

Further discussion on how to incorporate approximate solutions to the quadratic programming subproblems can be found in [7, Section V].

5.1.4 A Linear Example

In the linear case, the functions g_k and h_k is (1) are affine so that they equal their linearizations. In this case, the problems (4) and (7) are equivalent and only one subproblem of the form (10), or equivalently (12), needs to be solved. We illustrate the ℓ_1-Laplace smoother described in Section 5.1.1 by applying it to the example studied in Section 2.6, except now the noise term v_k is modeled using the ℓ_1-Laplace density. The numerical experiment described below is take from [7, Section VI].

The numerical experiment uses two full periods of $X(t)$ generated with $N = 100$ and $\Delta t = 4\pi/N$; i.e., discrete time points equally spaced over the interval $[0, 4\pi]$. For $k = 1, \dots, N$ the measurements z_k were simulated by $z_k = X_2(t_k) + v_k$. In order to test the robustness of the ℓ_1 model to measurement noise containing outlier data, we generate v_k as a mixture of two normals with p denoting the fraction of outlier

Table 1 Median MSE and 95% confidence intervals for the different estimation methods

p	ϕ	GKF	IGS	ILS
0	–	.34 (.24, .47)	.04(.02, .1)	.04(.01, .1)
.1	1	.41(.26, .60)	.06(.02, .12)	.04(.02, .10)
.1	4	.59(.32, 1.1)	.09(.04, .29)	.05(.02, .12)
.1	10	1.0(.42, 2.3)	.17(.05, .55)	.05(.02, .13)
.1	100	6.8(1.7, 17.9)	1.3(.30, 5.0)	.05(.02, .14)

contamination; i.e.,

$$v_k \sim (1 - p)\mathbf{N}(0, 0.25) + p\mathbf{N}(0, \phi) \,. \tag{14}$$

This was done for $p \in \{0, 0.1\}$ and $\phi \in \{1, 4, 10, 100\}$. The model for the mean of z_k given x_k is $h_k(x_k) = (0, 1)x_k = x_{2,k}$. Here $x_{2,k}$ denotes the second component of x_k. The model for the variance of z_k given x_k is $R_k = 0.25$. This simulates a lack of knowledge of the distribution for the outliers; i.e, $p\mathbf{N}(0, \phi)$. Note that we are recovering estimates for the smooth function $-\sin(t)$ and its derivative $-\cos(t)$ using noisy measurements (with outliers) of the function values.
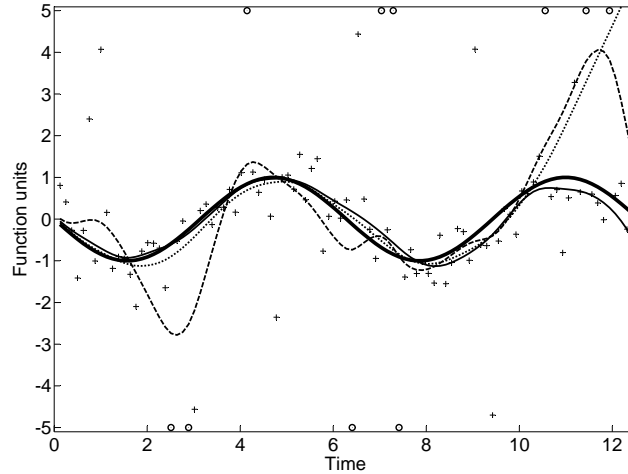
Fig. 7 Simulation: measurements (+), outliers (o) (absolute residuals more than three standard deviations), true function (thick line), ℓ_1-Laplace estimate (thin line), Gaussian estimate (dashed line), Gaussian outlier removal estimate (dotted line)

We simulated 1000 realizations of the sequence $\{z_k\}$ keeping the ground truth fixed, and for each realization, and each estimation method, we computed the corresponding state sequence estimate $\{\hat{x}_k\}$. The Mean Square Error (MSE) corresponding to such an estimate is defined by

$$\text{MSE} = \frac{1}{N} \sum_{k=1}^{N} [x_{1,k} - \hat{x}_{1,k}]^2 + [x_{2,k} - \hat{x}_{2,k}]^2 , \tag{15}$$

where $x_k = X(t_k)$. In Table 1, the Gaussian Kalman Filter is denoted by (GKF), the Iterated Gaussian Smoother (IGS), and the Iterated ℓ_1-Laplace Smoother (ILS). For each of these estimation techniques, each value of p, and each value of ϕ, the corresponding table entry is the median MSE followed by the centralized 95% confidence interval for the MSE. For this problem, the model functions $\{g_k(x_{k-1})\}$ and $\{h_k(x_k)\}$ are linear so the iterated smoothers IGS and ILS only require one iteration to estimate the sequence $\{\hat{x}_k\}$.

Note the ℓ_1-Laplace smoother performs nearly as well as the Gaussian smoother at the nominal conditions ($p = 0$). The ℓ_1-Laplace smoother performs better and more consistently in cases with data contamination ($p \geq .1$ and $\phi \geq 1$). It is also apparent that the smoothers perform better than the filters.

Outlier detection and removal followed by refitting is a simple approach to robust estimation and can be applied to the smoothing problem. An inherent weakness of this approach is that the outlier detection is done using an initial fit which assumes outliers are not present. This can lead to good data being classified as outliers and result in over fitting the remaining data. An example of this is illustrated in Figure 7 which plots the estimation results for a realization of $\{z_k\}$ where $p = 0.1$ and $\phi = 100$. Outlier removal also makes critical review of the model more difficult. A robust smoothing method with a consistent model, such as the ℓ_1-Laplace smoother, does not suffer from these difficulties.

5.1.5 Stochastic Nonlinear Process Example

We now illustrate the behavior of the ℓ_1-Laplace smoother on the Van Der Pol Oscillator described in Section 3.4. The numerical experiment we describe is taken from [7, Section VI]. The corresponding nonlinear differential equation is

$$\dot{X}_1(t) = X_2(t) \quad \text{and} \quad \dot{X}_2(t) = \mu[1 - X_1(t)^2]X_2(t) - X_1(t) .$$

Given $X(t_{k-1}) = x_{k-1}$ the Euler approximation for $X(t_{k-1} + \Delta t)$ is

$$g_k(x_{k-1}) = \begin{pmatrix} x_{1,k-1} + x_{2,k-1}\Delta t \\ x_{2,k-1} + \{\mu[1 - x_{1,k}^2]x_{2,k} - x_{1,k}\}\Delta t \end{pmatrix} .$$

For this simulation, the 'ground truth' is obtained from a stochastic Euler approximation of the Van der Pol oscillator. To be specific, with $\mu = 2$, $N = 164$ and $\Delta t = 16/N$, the ground truth state vector x_k at time $t_k = k\Delta t$ is given by

$x_0 = (0, -0.5)^{\mathrm{T}}$ and for $k = 1, \ldots, N$,

$$x_k = g_k(x_{k-1}) + w_k , \tag{16}$$

where $\{w_k\}$ is a realization of independent Gaussian noise with variance 0.01. Our model for state transitions (1) uses $Q_k = 0.01\,I$ for $k > 1$, and so is identical to the model used to simulate the ground truth $\{x_k\}$. Thus, we have precise knowledge of the process that generated the ground truth $\{x_k\}$. The initial state x_0 is imprecisely specified by setting $g_1(x_0) = (0.1, -0.4)^{\mathrm{T}} \neq x_0$ with corresponding variance $Q_1 = 0.1\,I$.

Table 2 Median MSE over 1000 runs and confidence intervals containing 95% of MSE results

| p	ϕ	IGS	ILS
0	–	0.07 (0.06, 0.08)	0.07 (0.06, 0.09)
.1	10	0.07 (0.06, 0.10)	0.07 (0.06, 0.09)
.2	10	0.08 (0.06, 0.11)	0.08 (0.06, 0.11)
.3	10	0.08 (0.06, 0.11)	0.08 (0.06, 0.11)
.1	100	0.10 (0.07, 0.14)	0.07 (0.06, 0.10)
.2	100	0.12 (0.07, 0.40)	0.08 (0.06, 0.11)
.3	100	0.13 (0.09, 0.64)	0.08 (0.07, 0.10)
.1	1000	0.17 (0.11, 1.50)	0.08 (0.06, 0.11)
.2	1000	0.21 (0.14, 2.03)	0.08 (0.06, 0.11)
.3	1000	0.25 (0.17, 2.66)	0.09 (0.07, 0.12)

For $k = 1, \ldots, N$ the measurements z_k were simulated by $z_k = x_{1,k} + v_k$. The measurement noise v_k was generated as follows:

$$v_k \sim (1 - p)\mathbf{N}(0, 1.0) + p\mathbf{N}(0, \phi) . \tag{17}$$

This was done for $p \in \{0, 0.1, 0.2, 0.3\}$ and $\phi \in \{10, 100, 1000\}$. The model for the mean of z_k given x_k is $h_k(x_k) = (1, 0)x_k = x_{1,k}$. As in the previous simulation, we simulated a lack of knowledge of the distribution for the outliers; i.e, $p\mathbf{N}(0, \phi)$. In (1), the model for the variance of z_k given x_k is $R_k = 1.0$.

We simulated 1000 realizations of the ground truth state sequence $\{x_k\}$ and the corresponding measurement sequence $\{z_k\}$. For each realization, we computed the corresponding state sequence estimate $\{\hat{x}_k\}$ using both the IGS and IKS procedures. The Mean Square Error (MSE) corresponding to such an estimate is defined by equation (15), where x_k is given by equation (16). The results of the simulation appear in Table 2. As the proportion and variance of the outliers increase, the Gaussian smoother degrades, but the ℓ_1-Laplace smoother is not affected.

Figure 8 provides a visual illustration of one realization $\{x_k\}$ and its corresponding estimates $\{\hat{x}_k\}$. The left two panels demonstrate that, when no outliers are present, both the IGS and ILS generate accurate estimates. Note that we only observe the first component of the state and that the variance of the observation is relatively large (see top two panels). The right two panels show what can go wrong
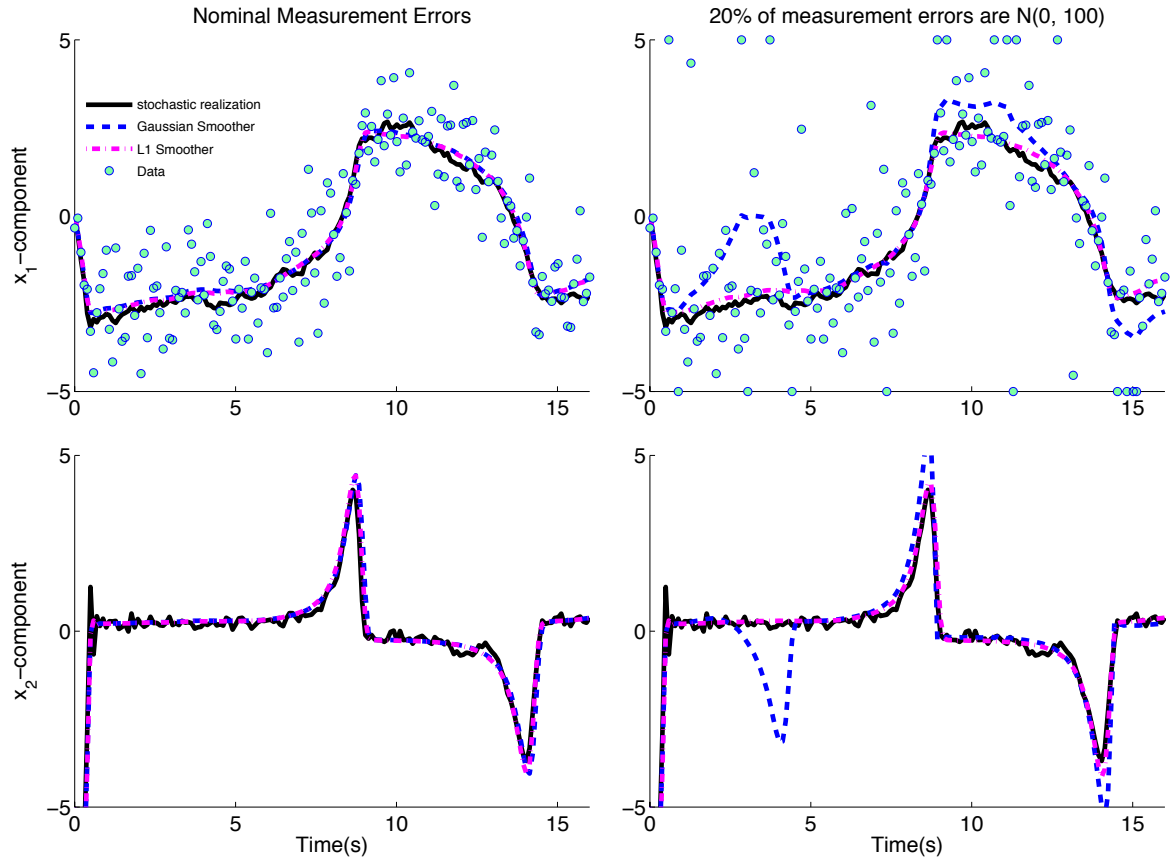
Fig. 8 The left two panels show estimation of x_1, (top) and x_2 (bottom) with errors from the nominal model. The stochastic realization is represented by a thick black line; the Gaussian smoother is the blue dashed line, and the ℓ_1-smoother is the magenta dash-dotted line. Right two panels show the same stochastic realization but with measurement errors now from $(p, \phi) = (.2, 100)$. Outliers appear on the top and bottom boundary in the top right panel.

when outliers are present. The Van der Pol oscillator can have sharp peaks as a result of the nonlinearity in its process model, and outliers in the measurements can 'trick' the IGS into these modes when they are not really present. In contrast, the Iterated ℓ_1-Laplace Smoother avoids this problem.

5.2 Further Extensions with Log-Concave Densities

Let us step back for a moment and examine a theme common to all of the variations on the Kalman smoother that we have examined thus far and compare the objective functions in (4), (6), (2), (3), and (16). In all cases, the objective function takes the form

$$\sum_{k=1}^{N} V_k\left(h(x_k) - z_k; R_k\right) + J_k\left(x_k - g(x_{k-1}); Q_k\right) , \tag{18}$$

where the mappings V_k and J_k are associated with log-concave densities of the form

$$p_{v,k}(z) \propto \exp\left(-V_k(z : R_k)\right)) \quad \text{and} \quad p_{w,k}(x) \propto \exp\left(-J_k(x; Q_k)\right)$$

with $p_{v,k}$ and $p_{w,k}$ having covariance matrices R_k and Q_k, respectively. The choice of the penalty functions V_k and J_k reflect the underlying model for distribution of the observations and the state, respectively. In many applications, the functions V_k and J_k are a members of the class of extended piecewise linear-quadratic penalty functions.

5.2.1 Extended Linear-Quadratic Penalties

Definition 1. For a nonempty polyhedral set $U \subset \mathbb{R}^m$ and a symmetric positive-semidefinite matrix $M \in \mathbb{R}^{m \times m}$ (possibly $M = 0$), define the function $\theta_{U,M} : \mathbb{R}^m \to \{\mathbb{R} \cup \infty\} := \overline{\mathbb{R}}$ by

$$\theta_{U,M}(w) := \sup_{u \in U}\left\{\langle u, w\rangle - \frac{1}{2}\langle u, Mu\rangle\right\}. \tag{19}$$

Given and injective matrix $B \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^m$, define $\rho : \mathbb{R}^n \to \overline{\mathbb{R}}$ as $\theta_{U,M}(b + By)$:

$$\rho_{U,M,b,B}(y) := \sup_{u \in U}\left\{\langle u, b + By\rangle - \frac{1}{2}\langle u, Mu\rangle\right\} . \tag{20}$$

All functions of the type specified in (19) are called *piecewise linear-quadratic* (PLQ) penalty functions, and those of the form (20) are called *extended piecewise linear-quadratic* (EPLQ) penalty functions.

Remark 1. PLQ penalty functions are extensively studied by Rockafellar and Wets in [43]. In particular, they present a full duality theory for optimizations problems based on these functions.

It is easily seen that the penalty functions arising from both the Gaussian and ℓ_1-Laplace distributions come from this EPLQ class. But so do other important densities such as the Huber and Vapnik densities.

Examples: The ℓ_2, ℓ_1, Huber, and Vapnik penalties are representable in the notation of Definition 1.
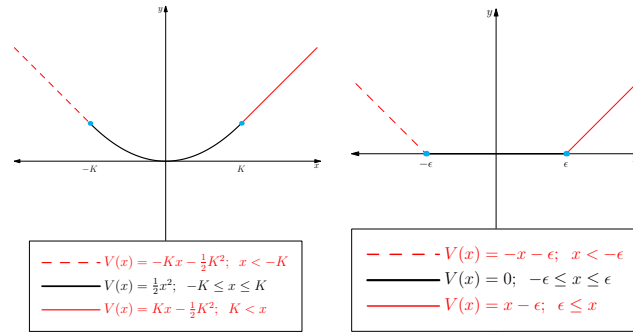
Fig. 9 Huber (left) and Vapnik (right) Penalties

1. L_2: Take $U = \mathbb{R}$, $M = 1$, $b = 0$, and $B = 1$. We obtain $\rho(y) = \sup_{u \in \mathbb{R}} \left\langle uy - \frac{1}{2}u^2 \right\rangle$.

   The function inside the sup is maximized at $u = y$, whence $\rho(y) = \frac{1}{2}y^2$.

2. ℓ_1: Take $U = [-1, 1]$, $M = 0$, $b = 0$, and $B = 1$. We obtain $\rho(y) = \sup_{u \in [-1,1]} \langle uy \rangle$.

   The function inside the sup is maximized by taking $u = \text{sign}(y)$, whence $\rho(y) = |y|$.

3. Huber: Take $U = [-K, K]$, $M = 1$, $b = 0$, and $B = 1$. We obtain $\rho(y) = \sup_{u \in [-K,K]} \left\langle uy - \frac{1}{2}u^2 \right\rangle$.

   Take the derivative with respect to u and consider the following cases:

   a. If $y < -K$, take $u = -K$ to obtain $-Ky - \frac{1}{2}K^2$.
   b. If $-K \leq y \leq K$, take $u = y$ to obtain $\frac{1}{2}y^2$.
   c. If $y > K$, take $u = K$ to obtain a contribution of $Ky - \frac{1}{2}K^2$.

   This is the Huber penalty with parameter K, shown in the left panel of Fig. 1.

4. Vapnik: take $U = [0, 1] \times [0, 1]$, $M = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right]$, $B = \left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right]$, and $b = \left[\begin{smallmatrix} -\varepsilon \\ -\varepsilon \end{smallmatrix}\right]$, for some $\varepsilon > 0$. We obtain $\rho(y) = \sup_{u_1, u_2 \in [0,1]} \left\langle \left[\begin{smallmatrix} y - \varepsilon \\ -y - \varepsilon \end{smallmatrix}\right], \left[\begin{smallmatrix} u_1 \\ u_2 \end{smallmatrix}\right] \right\rangle$. We can obtain an explicit representation by considering three cases:

   a. If $|y| < \varepsilon$, take $u_1 = u_2 = 0$. Then $\rho(y) = 0$.
   b. If $y > \varepsilon$, take $u_1 = 1$ and $u_2 = 0$. Then $\rho(y) = y - \varepsilon$.
   c. If $y < -\varepsilon$, take $u_1 = 0$ and $u_2 = 1$. Then $\rho(y) = -y - \varepsilon$.

   This is the Vapnik penalty with parameter ε, shown in the right panel of Fig. 5.2.1.

5.2.2 PLQ Densities

We caution that not every EPLQ function is the negative log of a density function. For an ELQP function ρ to be associated with a density, the function $\exp(-\rho(x))$ must be integrable on $\mathbb{R}^n$. The integrability of $\exp(-\rho(x))$ can be established under a coercivity hypothesis.

Definition 2. A function $\rho : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\} = \overline{\mathbb{R}}$ is said to be coercive (or 0-coercive) if $\lim_{\|x\| \to \infty} \rho(x) = +\infty$.

Since the functions $\rho_{U,M,b,B}$ defined in (20) are not necessarily finite-valued, their calculus must be treated with care. An important tool in this regard is the essential dominion. The essential domain of $\rho : \mathbb{R}^n \to \overline{\mathbb{R}}$ is the set

$$\mathrm{dom}(\rho) := \{x : \rho(x) < +\infty\}.$$

The affine hull of $\mathrm{dom}(\rho)$ is the smallest affine set containing $\mathrm{dom}(\rho)$, where a set is affine if it is the translate of a subspace.

Theorem 1. *[4, Theorem 6] (PLQ Integrability). Let $\rho := \rho_{U,M,b,B}$ be defined as in* (20). *Suppose $\rho(y)$ is coercive, and let n_{aff} denote the dimension of* $\mathrm{aff}(\mathrm{dom}\,\rho)$. *Then the function $f(y) = \exp(-\rho(y))$ is integrable on* $\mathrm{aff}(\mathrm{dom}\,\rho)$ *with the n_{aff}-dimensional Lebesgue measure.*

∎

Theorem 2. *[4, Theorem 7] (Coercivity of ρ). The function $\rho_{U,M,b,B}$ defined in* (20) *is coercive if and only if $[B^{\mathrm{T}}\mathrm{cone}(U)]^{\circ} = \{0\}$.*

∎

If $\rho := \rho_{U,M,b,B}$ is coercive, then, by Theorem 1, then the function $f(y) = \exp(-\rho(y))$ is integrable on $\mathrm{aff}(\mathrm{dom}\,\rho)$ with the n_{aff}-dimensional Lebesgue measure. If we define

$$\mathbf{p}(y) = \begin{cases} c_1^{-1}\exp(-\rho(y)) & y \in \mathrm{dom}\,\rho \\ 0 & \text{else,} \end{cases} \tag{21}$$

where

$$c_1 = \left(\int_{y \in \mathrm{dom}\,\rho} \exp(-\rho(y)) dy \right),$$

and the integral is with respect to the Lebesgue measure with dimension n_{aff}, then $\mathbf{p}$ is a probability density on $\mathrm{dom}(\rho)$. We call these PLQ densities.

5.2.3 PLQ Densities and Kalman Smoothing

We now show how to build up the penalty functions V_k and J_k in (18) using PLQ densities. We will do this for the linear model (1)-(2) for simplicity. The nonlin-

ear case can be handled as before by applying the Gauss-Newton strategy to the underlying convex composite function.

Using the notion given in (5), the linear model (1)-(2) can be written as

$$
\begin{aligned}
w &= Gx + \mathbf{w} \\
z &= Hx + \mathbf{v} \,.
\end{aligned}
\tag{22}
$$

A general Kalman smoothing problem can be specified by assuming that the noises $\mathbf{w}$ and $\mathbf{v}$ in the model (22) have PLQ densities with means 0, variances Q and R (5). Then, for suitable $\{U_k^w, M_k^w, b_k^w, B_k^w\}$ and $\{U_k^v, M_k^v, b_k^v, B_k^v\}$, we have

$$
\begin{aligned}
\mathbf{p}(w) &\propto \exp(-\theta_{U^w, M^w}(b^w + B^w Q^{-1/2} w)) \\
\mathbf{p}(v) &\propto \exp(-\theta_{U^v, M^v}(b^v + B^v R^{-1/2} v)) \,,
\end{aligned}
\tag{23}
$$

where

$$
\begin{array}{c}
U^w = \prod_{k=1}^{N} U_k^w \subset \mathbb{R}^{nN} \\[2mm]
U^v = \prod_{k=1}^{N} U_k^v \subset \mathbb{R}^{M}
\end{array}
\,,\quad
\begin{array}{c}
M^w = \mathrm{diag}(\{M_k^w\}) \\[1mm]
M^v = \mathrm{diag}(\{M_k^v\})
\end{array}
\,,\quad
\begin{array}{c}
B^w = \mathrm{diag}(\{B_k^w\}) \\[1mm]
B^v = \mathrm{diag}(\{B_k^v\}) \\[1mm]
b^w = \mathrm{vec}(\{b_k^w\}) \\[1mm]
b^v = \mathrm{vec}(\{b_k^v\})
\end{array}
\,.
$$

Then the MAP estimator for x in the model (22) is

$$
\arg \min_{x \in \mathbb{R}^{nN}} \left\{ \begin{array}{c} \theta_{U^w, M^w}(b^w + B^w Q^{-1/2}(Gx - w)) \\ + \theta_{U^v, M^v}(b^v + B^v R^{-1/2}(Hx - z)) \end{array} \right\} \,.
\tag{24}
$$

Note that since w_k and v_k are independent, problem (24) is decomposable into a sum of terms analogous to (18). This special structure follows from the block diagonal structure of H, Q, R, B^v, B^w, the bidiagonal structure of G, and the product structure of sets U^w and U^v, and is key in proving the linear complexity of the solution method we propose.

5.2.4 Solving the Kalman Smoother Problem with PLQ Densities

Recall that, when the sets U^w and U^v are polyhedral, (24) is an Extended Linear Quadratic program (ELQP), described in [43, Example 11.43]. We solve (24) by working directly with its associated Karush-Kuhn-Tucker (KKT) system.

Lemma 1. *[4, Lemma 3.1] Suppose that the sets U_k^w and U_k^v are polyhedral, that is, they can be given the representation*

$$
U_k^w = \{u | (A_k^w)^T u \le a_k^w\}, \quad U_k^v = \{u | (A_k^v)^T u \le a_k^v\} \,.
$$

Then the first-order necessary and sufficient conditions for optimality in (24) *are given by*

$$0 = (A^w)^{\mathrm{T}} u^w + s^w - a^w \; ; \quad 0 = (A^v)^{\mathrm{T}} u^v + s^v - a^v$$
$$0 = (s^w)^{\mathrm{T}} q^w \; ; \qquad\qquad 0 = (s^v)^{\mathrm{T}} q^v$$

$$0 = \tilde{b}^w + B^w Q^{-1/2} Gx - M^w u^w - A^w q^w$$
$$0 = \tilde{b}^v - B^v R^{-1/2} Hx - M^v u^v - A^v q^v \qquad\qquad , \qquad (25)$$
$$0 = G^{\mathrm{T}} Q^{-\mathrm{T}/2} (B^w)^{\mathrm{T}} u^w - H^{\mathrm{T}} R^{-\mathrm{T}/2} (B^v)^{\mathrm{T}} u^v$$
$$0 \leq s^w, s^v, q^w, q^v.$$

where $\tilde{b}^w = b^w - B^w Q^{-1/2} w$ and $\tilde{b}^v = b^v - B^v R^{-1/2} z$.

∎

We propose solving the KKT conditions (25) by an Interior Point (IP) method. IP methods work by applying a damped Newton iteration to a relaxed version of (25) where the relaxation is to the complementarity conditions. Specifically, we replace the complementarity conditions by

$$(s^w)^{\mathrm{T}} q^w = 0 \rightarrow Q^w S^w \mathbf{1} - \mu \mathbf{1} = 0$$
$$(s^v)^{\mathrm{T}} q^v = 0 \; \rightarrow Q^v S^v \mathbf{1} - \mu \mathbf{1} = 0 \,,$$

where Q^w, S^w, Q^v, S^v are diagonal matrices with diagonals q^w, s^w, q^v, s^v respectively. The parameter μ is aggressively decreased to 0 as the IP iterations proceed. Typically, no more than 10 or 20 iterations of the relaxed system are required to obtain a solution of (25), and hence an optimal solution to (24). The following theorem shows that the computational effort required (per IP iteration) is linear in the number of time steps whatever PLQ density enters the state space model.

Theorem 3. *[4, Theorem 3.2] (PLQ Kalman Smoother Theorem) Suppose that all w_k and v_k in the Kalman smoothing model* (1)-(2) *come from PLQ densities that satisfy* $\mathrm{Null}(M) \cap U^\infty = \{0\}$. *Then an IP method can be applied to solve* (24) *with a per iteration computational complexity of $O(Nn^3 + Nm)$.*

∎

The proof, which can be found in [4], shows that IP methods for solving (24) preserve the key block tridiagonal structure of the standard smoother. General smoothing estimates can therefore be computed in $O(Nn^3)$ time, as long as the number of IP iterations is fixed (as it usually is in practice, to 10 or 20).
It is important to observe that the motivating examples all satisfy the conditions of Theorem 3.

Corollary 1. *[4, Corollary 3.3] The densities corresponding to L^1, L^2, Huber, and Vapnik penalties all satisfy the hypotheses of Theorem 3.*

Proof: We verify that $\mathrm{Null}(M) \cap \mathrm{Null}(A^{\mathrm{T}}) = 0$ for each of the four penalties. In the L^2 case, M has full rank. For the L^1, Huber, and Vapnik penalties, the respective sets U are bounded, so $U^\infty = \{0\}$.

5.2.5 Numerical example: Vapnik penalty and functional recovery
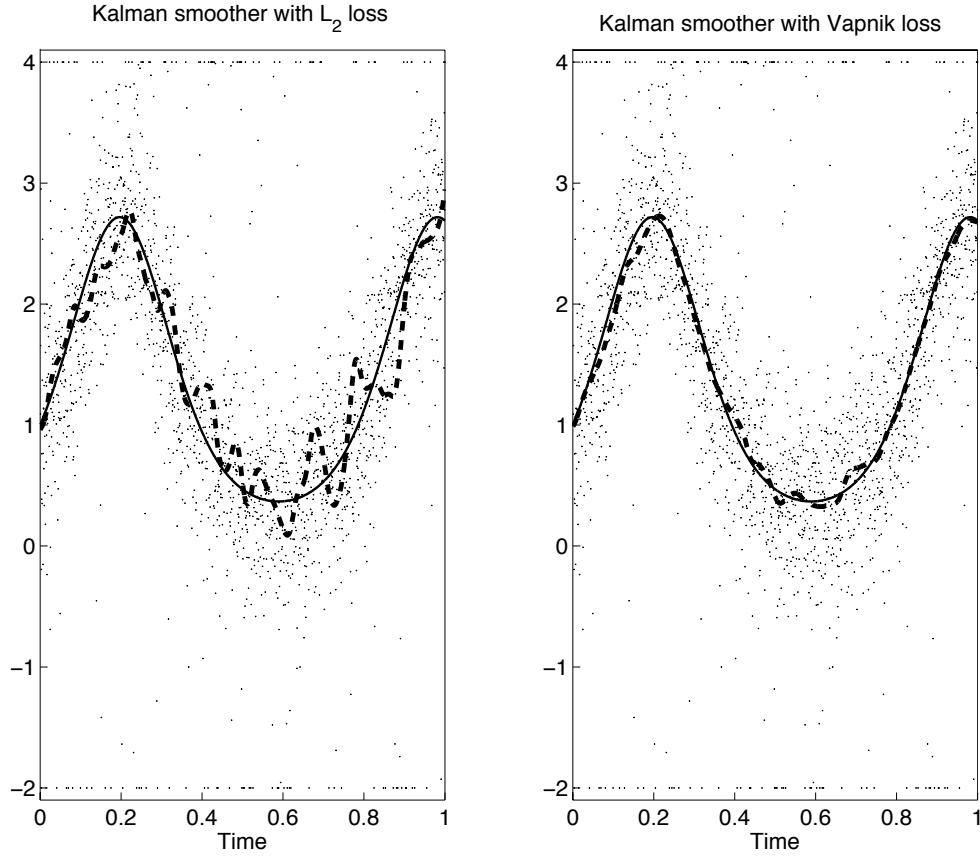
Fig. 10 Simulation: measurements (·) with outliers plotted on axis limits (4 and −2), true function (continuous line), smoothed estimate using either the quadratic loss (dashed line, left panel) or the Vapnik's ε-insensitive loss (dashed line, right panel)

In this section we present a numerical example to illustrate the use of the Vapnik penalty (see Figure 5.2.1) in the Kalman smoothing context, for a functional recovery application.

We consider the following function

$$f(t) = \exp\left[\sin(8t)\right]$$

taken from [19]. Our aim is to reconstruct f starting from 2000 noisy samples collected uniformly over the unit interval. The measurement noise v_k was generated

using a mixture of two normals with $p = 0.1$ denoting the fraction from each normal; i.e.,

$$v_k \sim (1-p)\mathbf{N}(0,0.25) + p\mathbf{N}(0,25),$$

where $\mathbf{N}$ refers to the Normal distribution. Data are displayed as dots in Fig. 10. Note that the purpose of the second component of the normal mixture is to simulate outliers in the output data and that all the measurements exceeding vertical axis limits are plotted on upper and lower axis limits (4 and -2) to improve readability.

The initial condition $f(0) = 1$ is assumed to be known, while the difference of the unknown function from the initial condition (i.e. $f(\cdot) - 1$) is modeled as a Gaussian process given by an integrated Wiener process. This model captures the Bayesian interpretation of cubic smoothing splines [48], and admits a 2-dimensional state space representation where the first component of $x(t)$, which models $f(\cdot) - 1$, corresponds to the integral of the second state component, modelled as Brownian motion. To be more specific, letting $\Delta t = 1/2000$, the sampled version of the state space model (see [26, 38] for details) is defined by

$$G_k = \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix}, \qquad k = 2, 3, \ldots, 2000$$

$$H_k = \begin{bmatrix} 0 & 1 \end{bmatrix}, \qquad k = 1, 2, \ldots, 2000$$

with the autocovariance of w_k given by

$$Q_k = \lambda^2 \begin{bmatrix} \Delta t & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & \frac{\Delta t^3}{3} \end{bmatrix}, \qquad k = 1, 2, \ldots, 2000 \, ,$$

where λ^2 is an unknown scale factor to be estimated from the data.

The performance of two different Kalman smoothers are compared. The first (classical) estimator uses a quadratic loss function to describe the negative log of the measurement noise density and contains only λ^2 as unknown parameter. The second estimator is a Vapnik smoother relying on the ε-insensitive loss, and so depends on two unknown parameters λ^2 and ε. In both of the cases, the unknown parameters are estimated by means of a cross validation strategy where the 2000 measurements are randomly split into a training and a validation set of 1300 and 700 data points, respectively. The Vapnik smoother was implemented by exploiting the efficient computational strategy described in the previous section, see [8] for specific implementation details. In this way, for each value of λ^2 and ε contained in a 10×20 grid on $[0.01, 10000] \times [0, 1]$, with λ^2 logarithmically spaced, the function estimate was rapidly obtained by the new smoother applied to the training set. Then, the relative average prediction error on the validation set was computed, see Fig. 11. The parameters leading to the best prediction were $\lambda^2 = 2.15 \times 10^3$ and $\varepsilon = 0.45$, which give a sparse solution defined by fewer than 400 support vectors. The value of λ^2 for the classical Kalman smoother was then estimated following the same strategy described above. In contrast to the Vapnik penalty, the quadratic loss does not induce any sparsity, so that, in this case, the number of support vectors

equals the size of the training set.

The left and right panels of Fig. 10 display the function estimate obtained using the quadratic and the Vapnik losses, respectively. It is clear that the Gaussian estimate is heavily affected by the outliers. In contrast, as expected, the estimate coming from the Vapnik based smoother performs well over the entire time period, and is virtually unaffected by the presence of large outliers.
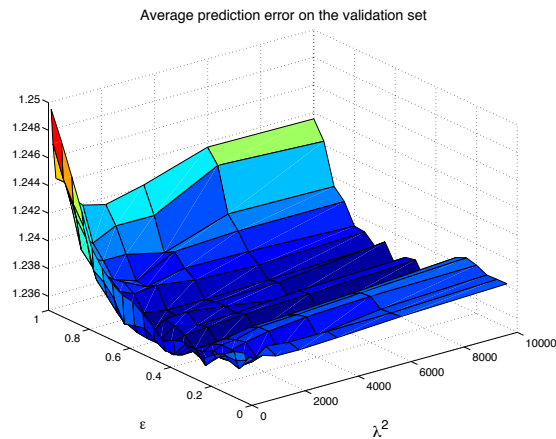
Fig. 11 Estimation of the smoothing filter parameters using the Vapnik loss. Average prediction error on the validation data set as a function of the variance process λ^2 and ε.

6 Sparse Kalman smoothing

In recent years, sparsity promoting formulations and algorithms have made a tremendous impact in signal processing, reconstruction algorithms, statistics, and inverse problems (see e.g. [13] and the references therein). In some contexts, rigorous mathematical theory is available that can guarantee recovery from under-sampled sparse signals [20]. In addition, for many inverse problems, sparsity promoting optimization provides a way to exploit prior knowledge of the signal class as a way to improve the solution to an ill-posed problem, but conditions for recoverability have not yet been derived [36].

   In the context of dynamic models, several sparse Kalman filters have been recently proposed [17, 16, 47, 1]. In the applications considered, in addition to process and measurement models, the state space is also known to be sparse. The aim is to improve recovery by incorporating sparse optimization techniques. Reference [1] is very close to the work presented in this section, since they formulate a sparsity

promoting optimization problem over the whole measurement sequence and solve it with an optimization technique shown to preserve computational efficiency.

In this section, we formulate the sparse Kalman smoothing problem as an optimization problem over the entire state space sequence, and suggest two new approaches for the solution of such problems. The first approach is based on the interior point methodology, and is a natural extension of the mathematics presented in earlier sections.

The second approach is geared towards problems where the dimension n (state at a single time point) is large. For this case, we propose a matrix free approach, using a different (constrained) Kalman smoothing formulation, together with the projected gradient method. In both methods, the structure of the Kalman smoothing problem is exploited to achieve computational efficiency.

We present theoretical development for the two approaches, leaving applications and numerical results to future work.

6.1 Penalized Formulation and Interior Point Approach

We consider only the linear smoother (6). A straight forward way to impose sparsity on the state is to augment this formulation with a 1-norm penalty:

$$\min_x f(x) := \frac{1}{2}\|Hx - z\|_{R^{-1}}^2 + \frac{1}{2}\|Gx - w\|_{Q^{-1}}^2 + \lambda \|Wx\|_1 , \tag{1}$$

where W is a diagonal weighting matrix included for modeling convenience. For example, the elements of W can be set to 0 to exclude certain parts of the state dimension from the sparse penalty. A straightforward constrained reformulation of (1) is

$$\min_x \quad \frac{1}{2}\|Hx - z\|_{R^{-1}}^2 + \frac{1}{2}\|Gx - w\|_{Q^{-1}}^2 + \lambda \mathbf{1}^T y \tag{2}$$
$$\text{s.t.} \quad -y \leq Wx \leq y .$$

Note that this is different from the constrained problem (3), because we have introduced a new variable y, with constraints in x and y. Nonetheless, an interior point approach may still be used to solve the resulting problem. We rewrite the constraint in (8) using non-negative slack variables s, r:

$$Wx - y + s = 0$$
$$-Wx - y + r = 0 , \tag{3}$$

and form the Lagrangian for the corresponding system:

$$L(s, r, q, p, y, x) = x^T Cx + c^T x + \lambda \mathbf{1}^T y + q^T (Wx - y + s) + p^T (-Wx - y + r) , \tag{4}$$

with C as in (8) and c as in (5)., and where q and p are the dual variables corresponding to the inequality constraints $Wx \leq y$ and $-Wx \leq -y$, respectively. The (relaxed)

KKT system is therefore given by

$$F_\mu(s,r,q,p,y,x) := \begin{pmatrix} s - y + Wx \\ r - y - Wx \\ D(s)D(q)\mathbf{1} - \mu\mathbf{1} \\ D(r)D(p)\mathbf{1} - \mu\mathbf{1} \\ \lambda\mathbf{1} - q - p \\ Wq - Wp + Cx + c \end{pmatrix} = 0 \,. \tag{5}$$

The derivative matrix $F_\mu^{(1)}$ is given by

$$F_\mu^{(1)} = \begin{bmatrix} I & 0 & 0 & 0 & -I & W \\ 0 & I & 0 & 0 & -I & -W \\ D(q) & 0 & D(s) & 0 & 0 & 0 \\ 0 & D(p) & 0 & D(r) & 0 & 0 \\ 0 & 0 & -I & -I & 0 & 0 \\ 0 & 0 & W & -W & 0 & C \end{bmatrix} \,, \tag{6}$$

and it is row equivalent to the system

$$\begin{bmatrix} I & 0 & 0 & 0 & -I & W \\ 0 & I & 0 & 0 & -I & -W \\ 0 & 0 & D(s) & 0 & D(q) & -D(q)W \\ 0 & 0 & 0 & D(r) & D(p) & D(p)W \\ 0 & 0 & 0 & 0 & \Phi & -\Psi W \\ 0 & 0 & 0 & 0 & 0 & C + W\Phi^{-1}\left(\Phi^2 - \Psi^2\right)W \end{bmatrix}$$

where

$$\begin{aligned} \Phi &= D(s)^{-1}D(q) + D(r)^{-1}D(p) \\ \Psi &= D(s)^{-1}D(q) - D(r)^{-1}D(p) \,, \end{aligned} \tag{7}$$

and the matrix $\Phi^2 - \Psi^2$ is diagonal, with the iith entry given by $4q_ir_i$. Therefore, the modified system preserves the structure of C; specifically it is symmetric, block tridiagonal, and positive definite. The Newton iterations required by the interior point method can therefore be carried out, with each iteration having complexity $O(n^3N)$.

6.2 Constrained Formulation and Projected Gradient Approach

Consider again the linear smoother (6), but now impose a 1-norm constraint rather than a penalty:

$$\min_x f(x) := \frac{1}{2}\|Hx - z\|^2_{R^{-1}} + \frac{1}{2}\|Gx - w\|^2_{Q^{-1}}$$
$$\text{s.t.} \quad \|Wx\|_1 \leq \tau . \tag{8}$$

This problem, which equivalent to (1) for certain values of λ and τ, is precisely the LASSO problem [45], and can be written

$$\min \frac{1}{2}x^T Cx + c^T x \quad \text{s.t.} \quad \|Wx\|_1 \leq \tau . \tag{9}$$

with $C \in \mathbb{R}^{nN \times nN}$ as in (8) and $c \in \mathbb{R}^{nN}$ as in (5). When n is large, the interior point method proposed in the previous section may not be feasible, since it requires exact solutions of the system

$$(C + W\Phi^{-1}\left(\Phi^2 - \Psi^2\right)W)x = r ,$$

and the block-tridiagonal algorithm 2.1 requires the inversion of $n \times n$ systems.

The problem (9) can be solved without inverting such systems, using the spectral projected gradient method, see e.g. [46, Algorithm 1]. Specifically, the gradient $Cx + c$ must be repeatedly computed, and then $x^v - (Cx^v + c)$ is projected onto the set $\|Wx\|_1 \leq \tau$. (the word 'spectral' refers to the fact that the Barzilai-Borwein line search is used to get the step length).

In the case of the Kalman smoother, the gradient $Cx + c$ can be computed in $O(n^2 N)$ time, because of the special structure of C. Thus for large systems, the projected gradient method that exploits the structure of C affords significant savings per iteration relative to the interior point approach, $O(n^2 N)$ vs. $O(n^3 N)$, and relative to a method agnostic to the structure of C, $O(n^2 N)$ vs. $O(n^2 N^2)$. The projection onto the feasible set $\|Wx\|_1 \leq \tau$ can be done in $O(nN \log(nN))$ time.

7 Conclusions

In this chapter, we have presented an optimization approach to Kalman smoothing, together with a survey of applications and extensions. In Section 2.5, we showed that the recursive Kalman filtering and smoothing algorithm is equivalent to algorithm 2.1, an efficient method to solve block tridiagonal positive definite systems. In the following sections, we used this algorithm as a subroutine, allowing us to present new ideas on a high level, without needing to explicitly write down modified Kalman filtering and smoothing equations.

We have presented extensions to nonlinear process and measurement models in Section 3, described constrained Kalman smoothing (both the linear and nonlinear cases) in Section 4, and presented an entire class of robust Kalman smoothers (derived by considering log-linear-quadratic densities) in Section 5. For all of these applications, nonlinearity in the process, measurements, and constraints can be handled by a generalized Gauss-Newton method that exploits the convex composite

structure discussed in Sections 3.1 and 4.4. The GN subproblem can be solved either in closed form or via an interior point approach; in both cases algorithm 2.1 was used. For all of these extensions, numerical illustrations have also been presented, and most are available for public release through the `ckbs` package [6].

In the case of the robust smoothers, it is possible to extend the density modeling approach by considering densities outside the log-concave class [3], but we do not discuss this work here.

We ended the survey of extensions by considering two novel approaches to Kalman smoothing of sparse systems, for applications where modeling the sparsity of the state space sequence improves recovery. The first method built on the readers' familiarity with the interior point approach as a tool for the constrained extension in Section 4. The second method is suitable for large systems, where exact solution of the linear systems is not possible. Numerical illustrations of the methods have been left to future work.

References

1. D. Angelosante, S.I. Roumeliotis, and G.B. Giannakis. Lasso-kalman smoother for tracking sparse signals. In *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*, pages 181–185, nov. 2009.
2. C.F. Ansley and R. Kohn. A geometric derivation of the fixed interval smoothing algorithm. *Biometrika*, 69:486–487, 1982.
3. A. Aravkin, James Burke, and Gianluigi Pillonetto. Robust and trend-following kalman smoothers using student's t. In *International Federation of Automaic Control (IFAC), 16th Symposium of System Identification*, oct. 2011.
4. A. Aravkin, James Burke, and Gianluigi Pillonetto. A statistical and computational theory for robust and sparse Kalman smoothing. In *International Federation of Automaic Control (IFAC), 16th Symposium of System Identification*, oct. 2011.
5. A.Y. Aravkin. *Robust Methods with Applications to Kalman Smoothing and Bundle Adjustment.* PhD thesis, University of Washington, Seattle, WA, June 2010.
6. A.Y. Aravkin, B.M. Bell, J.V. Burke, and G. Pillonetto. Matlab/Octave package for constrained and robust Kalman smoothing.
7. A.Y. Aravkin, B.M. Bell, J.V. Burke, and G. Pillonetto. An ℓ_1-laplace robust kalman smoother. *Automatic Control, IEEE Transactions on*, 56(12):2898–2911, dec. 2011.
8. A.Y. Aravkin, B.M. Bell, J.V. Burke, and G. Pillonetto. Learning using state space kernel machines. In *Proc. IFAC World Congress 2011*, Milan, Italy, 2011.
9. B.M. Bell. The iterated Kalman smoother as a Gauss-Newton method. *SIAM J. Optimization*, 4(3):626–636, August 1994.
10. B.M. Bell. The marginal likelihood for parameters in a discrete Gauss-Markov process. *IEEE Transactions on Signal Processing*, 48(3):626–636, August 2000.
11. B.M. Bell, J.V. Burke, and G. Pillonetto. An inequality constrained nonlinear kalman-bucy smoother by interior point likelihood maximization. *Automatica*, 45(1):25–33, January 2009.
12. B.M. Bell and F. Cathey. The iterated Kalman filter update as a Gauss-Newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, February 1993.
13. Alfred M. Bruckstein, David L. Donoho, and Michael Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev.*, 51(1):34–81, February 2009.
14. J. V. Burke and S-P. Han. A robust sequential quadratic programming method. *Mathematical Programming*, 43:277–303, 1989. 10.1007/BF01582294.

15. James V. Burke. Descent methods for composite nondifferentiable optimization problems. *Mathematical Programming*, 33:260–279, 1985.
16. A. Carmi, P. Gurfil, and D. Kanevsky. Methods for sparse signal recovery using kalman filtering with embedded pseudo-measurement norms and quasi-norms. *IEEE Transactions on Signal Processing*, In print.
17. A. Carmi, P. Gurfil, and D. Kanevsky. A simple method for sparse signal recovery from noisy observations using kalman filtering. Technical Report RC24709, Human Language Technologies, IBM, 2008.
18. Rudoph Van der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, OGI School of Science and Engineering, Oregon Health and Science University, April 2004.
19. F. Dinuzzo, M. Neve, G. De Nicolao, and U. P. Gianazza. On the representer theorem and equivalent degrees of freedom of SVR. *Journal of Machine Learning Research*, 8:2467–2495, 2007.
20. D.L. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289 –1306, april 2006.
21. L. Fahrmeir and H. Kaufmann. On Kalman filtering, posterior mode estimation, and Fisher scoring in dynamic exponential family regression. *Metrika*, pages 37–60, 1991.
22. Ludwig Fahrmeir and Rita Kunstler. Penalized likelihood smoothing in robust state space models. *Metrika*, 49:173–191, 1998.
23. Junbin Gao. Robust l1 principal component analysis and its Bayesian variational inference. *Neural Computation*, 20(2):555–572, February 2008.
24. S. Gillijns, O. Barrero Mendoza, J. Chandrasekar, B. L. R. De Moor, D. S. Bernstein, and A. Ridley. What is the ensemble Kalman filter and how well does it work? In *Proceedings of the American Control Conference*, pages 4448–4453. IEEE, 2006.
25. G.A. Hewer, R.D. Martin, and Judith Zeh. Robust preprocessing for Kalman filtering of glint noise. *IEEE Transactions on Aerospace and Electronic Systems*, AES-23(1):120–128, January 1987.
26. Andrew Jazwinski. *Stochastic Processes and Filtering Theory*. Dover Publications, Inc, 1970.
27. JR J.E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimiation and Nonlinear Equations*. Computational Mathematics. Prentice-Hall, 1983.
28. Simon Julier, Jeffrey Uhlmann, and Hugh Durrant-White. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, March 2000.
29. R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the AMSE - Journal of Basic Engineering*, 82(D):35–45, 1960.
30. R. Kandepu, B. Foss, and L. Imsland. Applying the unscented Kalman filter for nonlinear state estimation. *J. of Process Control*, 18:753–768, 2008.
31. Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky. ℓ_1 trend filtering. *Siam Review*, 51(2):339–360, 2009.
32. M. Kojima, N. Megiddo, T. Noma, and A. Yoshise. *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*, volume 538 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, Germany, 1991.
33. S. Kourouklis and C. C. Paige. A constrained least squares approach to the general Gauss-Markov linear model. *Journal of the American Statistical Association*, 76(375):620–625, September 1981.
34. T. Lefebvre, H. Bruyninckx, and J. De Schutter. Kalman filters for nonlinear systems: A comparison of performance. *Intl. J. of Control*, 77(7):639–653, May 2004.
35. Jun S. Liu and Rong Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.
36. Hassan Mansour, Haneet Wason, Tim T.Y. Lin, and Felix J. Herrmann. Randomized marine acquisition with compressive sampling matrices. *Geophysical Prospecting*, 60(4):648–662, 2012.
37. A. Nemirovskii and Y. Nesterov. *Interior-Point Polynomial Algorithms in Convex Programming*, volume 13 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, USA, 1994.

38. Bernt Oksendal. *Stochastic Differential Equations*. Springer, sixth edition, 2005.
39. C. C. Paige and M. A. Saunders. Least squares estimation of discrete linear dynamic systems using orthogonal transformations. *Siam J. Numer. Anal*, 14(2):180–193, April 1977.
40. Christopher C. Paige. Covariance matrix representation in linear filtering. *Contemporary Mathematics*, 47, 1985.
41. Gianluigi Pillonetto, Aleksandr Y. Aravkin, and Stefano Carpin. The unconstrained and inequality constrained moving horizon approach to robot localization. In *IROS*, pages 3830–3835, 2010.
42. H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA J.*, 3(8):1145–1150, 1965.
43. R. Tyrrell Rockafellar and Roger J-B. Wets. *Variational Analysis*, volume 317 of *A Series of Comprehensive Studies in Mathematics*. Springer, 1998.
44. Irvin C. Schick and Sanjoy K. Mitter. Robust recursive estimation in the presence of heavy-tailed observation noise. *The Annals of Statistics*, 22(2):1045–1080, June 1994.
45. R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.
46. E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008.
47. N. Vaswani. Kalman filtered compressed sensing. Proceedings of the IEEE International Conference on Image Processing (ICIP), October 2008.
48. G. Wahba. *Spline models for observational data*. SIAM, Philadelphia, 1990.
49. S.J. Wright. *Primal-dual interior-point methods*. Siam, Englewood Cliffs, N.J., USA, 1997.