Optimization Enabled Kalman Smoothing

Jonathan Jonker

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Aleksandr Aravkin, Chair

James Burke

John Sylvester

Program Authorized to Offer Degree:
Mathematics

University of Washington

Abstract

Optimization Enabled Kalman Smoothing

Jonathan Jonker

Chair of the Supervisory Committee:
Assistant Professor Aleksandr Aravkin
Applied Mathematics

Kalman smoothing has tremendous importance in a wide range of time series analysis applications. Classic algorithms use Gaussian assumptions to simplify estimation but optimization tools can be used to unlock more modeling capabilities.

We develop techniques that allow for efficient robust estimates in the presence of singular noise. Such models arise frequently in the presence of auto-correlated noise, bias, and constraint systems. We also consider simultaneous parameter estimation and provide second-order methods for these problems, even when covariance matrices are singular. This captures ARMA and many physics based models. We also develop second-order methods for a general class of convex-composite PLQ functions and use this to solve general robust nonlinear Kalman smoothing models.

A number of applications are considered that apply the developed methods. In particular we consider problems in navigation, finance, ADCP current estimation, and flight testing using a combination of real and synthetic data sets. We also include open source implementation of our models and share these via GitHub.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First I would like to thank my advisor Professor Aleksandr Aravkin. During my time in graduate school he introduced me to the world of optimization and changed my perspective on mathematics. His constant enthusiasm kept me engaged and motivated even during the most difficult stretches of my PhD. I am extremely grateful for all that he has done to shape my graduate experience for the better.

I would also like to thank a number of my coauthors, in particular Dr. Sarah Webster and Professor Jim Burke. Both of them have been very important in motivating much of the work I have done and their guidance and insight have shaped many of the projects that make up my thesis.

I am very thankful for my office mates, Jacob Richey, Kristine Hampton, and Amy Wiebe. They all have been very supportive and helpful over the years as well as providing a welcome break from research when needed.

Finally I would like to thank my parents whose support and encouragement has been incredibly helpful through my entire school experience but especially so during my PhD.

Chapter 1

INTRODUCTION

1.1 State Space Model

This thesis presents ways to combine and extend the Kalman filter/smoother to address emerging questions using new models and modern optimization techniques. First introduced in 1960 [82], the Kalman filter has been applied to many time series models in diverse fields including tracking, navigation, and signal processing [23, 6, 7, 73, 150]. It is a method of time series analysis built on a linear state space model:

$$\begin{aligned} x_1 &= x_0 + w_1 \\ x_k &= G_k x_{k-1} + w_k, \quad k = 2, \ldots, N \\ y_k &= H_k x_k + v_k, \quad k = 1, \ldots N. \end{aligned} \tag{1.1}$$

Here x_0 is an initial estimate of the state, $x_1, \ldots, x_N$ are unknown latent states with linear process $G_1, \ldots, G_N$. $y_1, \ldots, y_N$ are observations with known linear measurement model H_k. The errors w_k and v_k we assume to be mutually independent random variables with known non-singular covariances Q_k and R_k. A visualization of this system is shown in Figure 1.1.

Our main concern is to synthesize different types of information available into the state space model in order to obtain an estimate for the unknown states. More concretely, given measurements $y_1, \ldots, y_N$, we want to estimate x_k, as well as additional auxiliary parameters. This breaks into two distinct cases; if $k = N$ this problem is referred to as a *filtering* problem (that is using available information up to time k), and if $k < N$ it is referred to as a *smoothing* problem (where we are able to use future information). The Kalman filter and Kalman smoother provide one way of solving these problems. As the smoother is built from the filter we begin our discussion there.

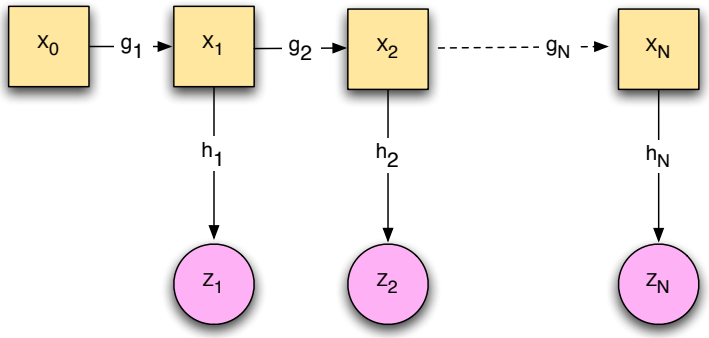


Figure 1.1: A visual representation of a state space system

1.2 Standard Kalman Algorithms

Here we present the classic Kalman filter and smoother algorithms. We will give a brief summary and intuition for the derivation but leave a fully detailed explanation to other sources [7].

1.2.1 The Kalman Filter

The derivation of the Kalman filter is centered on the following fact about jointly Gaussian distributions: If X and Y are jointly Gaussian with mean and covariance

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}, \quad \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}.$$

Then X conditioned on $Y = y$ is also a Gaussian random variable with mean and covariance

$$\bar{x} + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \bar{y}), \quad \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}.$$

Next we must introduce some notation. Let $Y_k = \{y_1, \ldots, y_k\}$ be all the measurements at times before and including k. Then let

$$\hat{x}_{k/k-1} = E[x_k|Y_{k-1}], \quad \Sigma_{k/k-1} = E[(x_k - \hat{x}_{k/k-1})(x_k - \hat{x}_{k/k-1})^T]$$

be the estimates of x_k conditioned on measurements *before* time k along with its covariance. Next define

$$\hat{x}_{k/k} = E[x_k|Y_k], \quad \Sigma_{k/k} = E[(x_k - \hat{x}_{k/k})(x_k - \hat{x}_{k/k})^T]$$

to be the estimate of state x_k updated with information *at* time k along with its covariance. Next we make an additional assumption that $\{v_k\}$ and $\{w_k\}$ are independent, mean zero, Gaussian random variables and that the initial state x_0 is a Gaussian random variable with mean $\bar{x}_0$ and covariance Q_0. Repeated application of the above fact then gives the following equations

$$\begin{aligned}
\hat{x}_{k/k} &= \hat{x}_{k/k-1} + \Sigma_{k/k-1} H_k \left(H_k^T \Sigma_{k/k-1} H_k + R_k\right)^{-1} (y_k - H_k^T \hat{x}_{k/k-1}) \\
\hat{x}_{k+1/k} &= G_k \hat{x}_{k/k} \\
\Sigma_{k/k} &= \Sigma_{k/k-1} - \Sigma_{k/k-1} H_k \left(H_k^T \Sigma_{k/k-1} H_k + R_k\right)^{-1} H_k^T \Sigma_{k/k-1} \\
\Sigma_{k+1/k} &= G_k \Sigma_{k/k} G_k^T + G_k Q_k G_k^T
\end{aligned} \tag{1.2}$$

where G_k and H_k are as in equation (1.1). The Kalman filter will be briefly used in the sequel but most of our work is on extending ideas from the Kalman smoother which we will now describe.

1.2.2 The Kalman Smoother

The smoothing problem allows us to use future information to obtain an estimate of the states. Therefore in general we expect a solution to the smoothing problem to be more accurate but more computationally taxing than a solution to the filtering problem. Before we proceed we must deal with the ambiguity present in the statement of the smoothing problem. One could fix j and estimate $x_{j/j+m}$ for all m and observe how additional information changes our current estimate. Another possibility is an online smoother where the lag time remains constant and we estimate $x_{k-N/N}$ for fixed N. The question we focus on is referred to as *fixed-interval* smoothing, where we imagine we are given a batch of measurement data and want to use all of it to estimate every state in the given time interval. More concretely, we

are given measurements $\{y_1, \ldots, y_N\}$ and we estimate $x_{k/N}$ for all k. We use 'smoothing problem' to refer to fixed interval smoothing.

We now present the equations for the Kalman smoother. The most famous of these consists of two parts: a forward and a backward pass. The forward pass is the traditional Kalman filter (equation (1.2)), the second is called the Rach-Tung-Striebel (RTS) smoother [124]. The RTS smoother can be thought of as a backwards pass through the data where information from the future propagates backwards and updates the filtered estimates. We begin at the final time point taking the smoothed estimate to be the one already computed by the filter, then proceed backwards using the following recursive equations:

$$\begin{aligned} \Gamma_k &= \Sigma_{k/k} G_k^T \left[G_k \Sigma_{k/k} G_k^T + Q_k \right]^{-1} \\ \hat{x}_{k/N} &= \hat{x}_{k/k} + \Gamma_k \left[\hat{x}_{k+1/N} - G_k \hat{x}_{k/k} \right] \\ \Sigma_{k/N} &= \Sigma_{k/k} + \Gamma_k \left[\Sigma_{k+1/N} - G_k \Sigma_{k/k} G_k^T - Q_k \right] \Gamma_k^T \end{aligned} \tag{1.3}$$

where $\hat{x}_{k/k}$ and $\Sigma_{k/k}$ are stored from (1.2).

1.3 An Optimization Perspective

Numerous extensions of the Kalman filter have been proposed such as the unscented Kalman filter [156], the extended Kalman filter (EKF; see e.g. [34]) and particle filters [65]. The EKF linearizes nonilnear process and/or measurement functions at the current estimate, while both particle and unscented filters approximate the statistics of the current state and propagate these approximations through the nonlinear dynamics.

Our extensions are based on the optimization perspective. This viewpoint allows us to create extensions that can be efficiently solved using optimization algorithms. For most extensions, the convenient statistical properties that are inherited from linear models with Gaussian errors no longer hold. In spite of this, the optimization perspective allows useful estimates to be efficiently obtained for a range of applications of current interest to practitioners who use state space models.

When models are linear and errors are assumed to be Gaussian, the expected value of the

state space sequence exactly aligns with the maximum a posteriori estimate, and solves a least squares problem. Classic algorithms exploit the underlying structure of this problem, and solve smaller least squares problems sequentially, interpreting these iterations as information moving forward or backward one step at a time. To view the Kalman smoother as a single optimization problem we re-derive it as the maximum *a posteriori* or MAP estimate, given by $\max_{\{x_k\}} \mathbf{p}\left(\{x_k\}|Y_N\right)$, where Y_N is as before and $\{x_k\}$ indicates the entire unknown state sequence. In the Gaussian case this agrees with the conditional mean estimate used above, but the MAP estimate makes it much more clear how to extend to non-Gaussian cases. Along with a change of variables suggested by Aravkin, Burke, and Pillonetto [17], we can write the Kalman smoother as a single least squares problem.

We begin by using Bayes' theorem

$$\mathbf{p}\left(\{x_k\}|Y_N\right) = \frac{\mathbf{p}\left(Y_N|\{x_k\}\right)\mathbf{p}\left(\{x_k\}\right)}{\mathbf{p}\left(Y_N\right)}$$

$$= \frac{\prod_k \mathbf{p}(y_k|x_k)\prod_k \mathbf{p}(x_{k+1}|x_k)}{\mathbf{p}\left(Y_N\right)} \propto \prod_k \mathbf{p}(w_k)\prod_k \mathbf{p}(v_k).$$

Under Gaussian assumptions these probability functions can be explcitly written. We then take a negative log to arrive at the MAP estimate

$$\arg\max_{\{x_k\}} \mathbf{p}\left(\{x_k\}|Y_N\right) = \arg\min_{\{x_k\}} \sum_{k=1}^{N} ||Q_k^{-1/2}(x_k - G_k x_{k-1})||^2 + ||R_k^{-1/2}(y_k - H_k x_k)||^2. \quad (1.4)$$

We condense this notation by defining the following stacked vectors

$$\eta = \begin{bmatrix} x_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{Nn}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^{Nn}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^{Nm}$$

and matrices

$$G = \begin{bmatrix} I & 0 & & \\ -G_2 & I & \ddots & \\ & \ddots & \ddots & 0 \\ & & -G_N & I \end{bmatrix} \in \mathbb{R}^{Nn \times Nn}, \qquad Q = \begin{bmatrix} Q_1 & 0 & & \\ 0 & \ddots & \ddots & \\ & \ddots & \ddots & 0 \\ & & 0 & Q_N \end{bmatrix} \in \mathbb{R}^{Nn \times Nn},$$

$$H = \begin{bmatrix} H_1 & 0 & & \\ 0 & \ddots & \ddots & \\ & \ddots & \ddots & 0 \\ & & 0 & H_N \end{bmatrix} \in \mathbb{R}^{Nm \times Nn}, \qquad R = \begin{bmatrix} R_1 & 0 & & \\ 0 & \ddots & \ddots & \\ & \ddots & \ddots & 0 \\ & & 0 & R_N \end{bmatrix} \in \mathbb{R}^{Nm \times Nm}.$$

The smoothing problem can now be written

$$\min_x ||Q^{-1/2}(Gx - \eta)||^2 + ||R^{-1/2}(Hx - y)||^2 \tag{1.5}$$

with x giving the optimal state pace sequence. Problem (1.5) has optimality conditions

$$\left(G^T Q^{-1} G + H^T R^{-1} H\right) \hat{x} = G^T Q^{-1} \eta + H^T R^{-1} y. \tag{1.6}$$

The matrix in (1.6) is a block tridiagonal matrix, a fact which allows (1.6) to be solved efficiently. If a standard forward block tridiagonal scheme is used to solve (1.6), then the same equations as (1.3) are derived [17]. While this is extremely useful in many fields it does have some shortcomings. In particular it cannot track fast changes in the state and the estimate is heavily influenced by outliers in the measurement data. We now examine two examples showcasing these problems as an illustration of why modeling extensions are necessary.

1.3.1 DC Motor Example

Consider an example of a dynamical system modeling a DC motor [98]. The state we take to be angular velocity and angle of the motor shaft and the outputs are noisy measurements of the angle of the motor shaft. Following a discretization from [113], we have the following

model:

$$x_{k+1} = \begin{pmatrix} 0.7 & 0 \\ 0.084 & 1 \end{pmatrix} x_k + \begin{pmatrix} 11.81 \\ 0.62 \end{pmatrix} d_k$$

$$y_k = \begin{pmatrix} 0 & 1 \end{pmatrix} x_k + v_k$$

where d_k is a disturbance process. This disturbance process is modeled as an independent random variable with

$$d_k = \begin{cases} 0 & \text{with probability } 1-\alpha \\ \mathcal{N}(0,1) & \text{with probability } \alpha. \end{cases}$$

This corresponds to a zero-mean (non-Gaussian) noise w_k, with covariance[1]

$$Q_k = \alpha \begin{pmatrix} 11.81 \\ 0.62 \end{pmatrix} \begin{pmatrix} 11.81 & 0.62 \end{pmatrix}.$$

We consider trying to reconstruct d_k from noisy measurements under the assumptions:

$$x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \alpha = 0.01, \quad v_k \sim \mathcal{N}(0, 0.1^2).$$

Figure 1.2 shows the smoothed estimates from solving (1.5). The estimate of d_k is rather poor and the large peak is drastically underestimated.

1.3.2 Outliers Corrupting Measurement Data

Now consider the case where the disturbance d_k is actually Gaussian. Setting $d_k \sim \mathcal{N}(0, 0.1^2)$ we then turn our attention to the measurement noise, v_k. We take outliers to occur with probability $\alpha = 0.1$ and assume they are generated from a distribution with standard deviation 100 times greater than that of nominal. This can be explicitly written as a mixture of two normal distributions:

$$v_k \sim (1-\alpha)\mathcal{N}(0, \sigma^2) + \alpha\mathcal{N}(0, (100\sigma)^2).$$

[1]Note that this is singular and as such will not work immediately in (1.5). Later we will allow for singular covariance but for now the RTS equations ((1.3)) can be used even with singular Q_k.

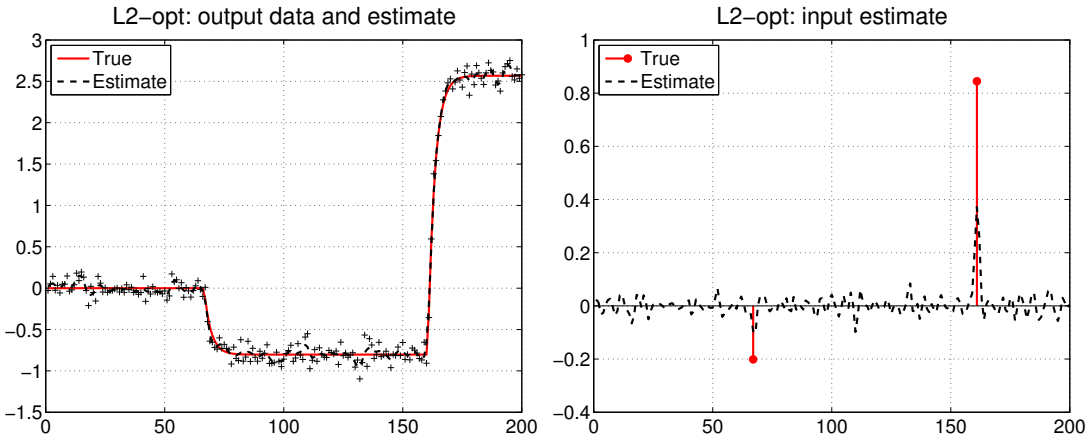


Figure 1.2: *Left:* true noiseless output and measurements (+) shown with estimate from solving (1.6). *Right:* estimated impulsive disturbance (d_k) along with its true value.

Taking $\sigma = 0.1$ we attempt to reconstruct the angle of the motor shaft (second component of the state). Two instances of this are shown in Figure 1.3, one using the nominal variance and one using the true variance of the mixed normal model. As seen in the plot neither one performs well. This is due to the fact that the least squares penalty is not robust and is heavily biased by outliers. Simply changing the scaling of the variance can not fix this underlying problem. This example shows that extensions of the base model of (1.5) are indeed required. In the next section we examine some of the work that has already been done.

1.4 Related Works

Originally the optimization perspective was used with the same idea of passing information forward and backwards. This can be done when nonlinear process and measurement maps are present by using a Gauss-Newton algorithm at each time step [30, 25]. A similar technique can be used when the loss functions are allowed to be robust [58, 59]. Paige and Saunders noted that the least squres structure can be exploited to allow for singular covariance in the case of Gaussian noise [119, 118]. Others have explored a more general perspective

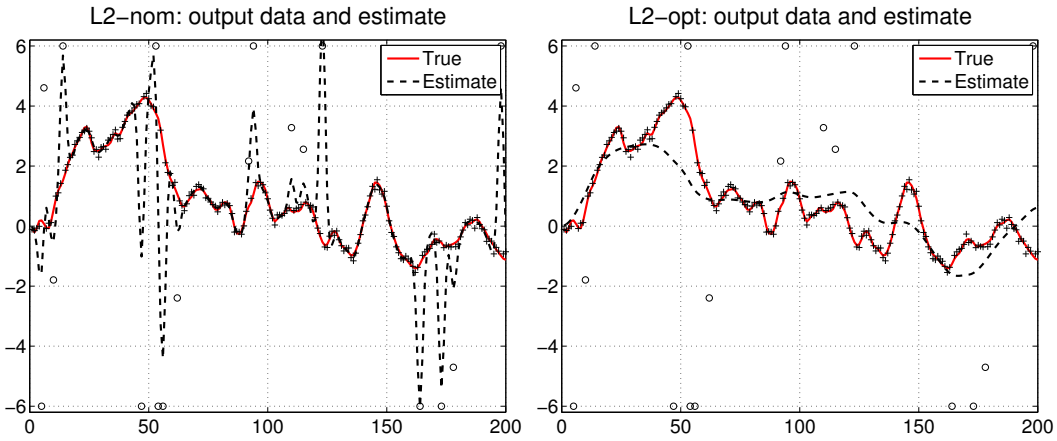


Figure 1.3: *Left:* smoothed estimates with assumed nominal measurement variance. *Right:* smoothed estimates with true measurement variance.

using statistics [56], but they still use an algorithm that only passes information forward and backward one time step.

We now will consider in more detail some of the most relevant advances where the smoother is viewed as a single optimization problem.

1.4.1 Nuisance Parameter Estimation

Consdier the case when measurements have a static bias. The state space equations for such a model are:

$$\begin{aligned} x_1 &= x_0 + w_1 \\ x_k &= G_k x_{k-1} + w_k, \quad k = 2, \ldots, N \\ y_k &= H_k x_k + P_k b + v_k, \quad k = 1, \ldots, N. \end{aligned}$$

The states can then solved for by minimizing the negative of the log posterior:

$$\min_z (y - \bar{H}z)^T R^{-1} (y - \bar{H}z) + (\eta - \bar{G}z)^T Q^{-1} (\eta - \bar{G}z)$$

where $\bar{H} = \begin{bmatrix} H & P \end{bmatrix}, \bar{G} = \begin{bmatrix} G & 0 \end{bmatrix}$, and $z = \begin{bmatrix} x \\ b \end{bmatrix}$. This is again a least squares problem but the Hessian is no longer block tridiagonal. Authors of [12] call this structure *augmented block tridiagonal* and show how it can be used to efficiently solve the resulting linear system.

1.4.2 Least Squares Penalty with Nonlinear Models and Constraints

Here we consider a slightly more general state space model where we allow for nonlinear process and measurement models, as well as nonlinear constraints. The MAP estimate then becomes

$$\begin{aligned} \min_x \; & ||Q^{-1/2}(G(x) - \eta)||^2 + ||R^{-1/2}(H(x) - y)||^2 \\ & \text{s.t. } F(x) \leq 0 \end{aligned} \tag{1.7}$$

where F is a given (possibly nonlinear) constraint function. [29] provides an algorithm to solve (1.7) by iteravely solving local Gauss-Newton approximations. Each local subproblemn is created with a linear approximation to all nonlinear functions, that is each has the form:

$$\begin{aligned} \min_x \; & ||Q^{-1/2}(G(\bar{x}) + \nabla G(\bar{x})(x - \bar{x})||^2 + ||R^{-1/2}(H(\bar{x}) + \nabla H(\bar{x})(x - \bar{x})||^2 \\ & \text{s.t. } F(\bar{x}) + \nabla F(\bar{x})(x - \bar{x}) \leq 0. \end{aligned} \tag{1.8}$$

This Quadratic Program (QP) is then solved via an interior point scheme (see for instance [162]) where a log-barrier term is added and driven to zero to ensure strict feasibility. Each iteration of the interior point method can be efficiently implemented, since the approximating QP preserves the underlying structure of the Kalman smoothing problem.

1.4.3 Robust Smoother with Nonlinear Models and Nonsmooth Loss

When outliers are present in the measurement data, any approach based on Gaussian errors (or, correspondingly, least squares) can easily fail to produce reliable estimates. Authors of [13] instead model the measurements noise as coming from a ℓ_1-Laplacian distribution. The MAP approach then leads to the problem

$$\min_x \; ||Q^{-1/2}(G(x) - \eta)||^2 + |R^{-1/2}(H(x) - y)|_1. \tag{1.9}$$

Similarly to method described above, [13] uses local Gauss-Newton approximations with an interior point scheme. However in this case the presence of the 1-norm requires additional developments in order to solve the local approximation using an efficient interior point method.

1.4.4 Fully Robust Estimates for any Convex PLQ Loss

The approach of [13] uses specific information about the 1-norm. The authors of [15] develop a general framework called IPSolve where the loss functions are allowed to be any piecewise linear quadratic (PLQ) function (for a definition and discussion see [126]). However, [15] only consider linear process and measurement maps:

$$\min_x \rho_1(Q^{-1/2}(Gx - \eta)) + \rho_2(R^{-1/2}(Hx - y)), \tag{1.10}$$

where ρ_1, ρ_2 are any PLQ penalties. The method detailed in [15] uses the dual representation of PLQ penalties [126]. For any given PLQ penalty ρ, there is a symmetric positive semidefinite (psd) matrix M and nonempty polyhedral set $\mathcal{U}$ with $0 \in \mathcal{U}$, so that for any $z \in \text{dom } \rho$ we can write

$$\rho(z) = \sup_{u \in \mathcal{U}} \{\langle u, z\rangle - \frac{1}{2}\langle u, Mu\rangle\}. \tag{1.11}$$

This representation is also used in Chapter 4 of this thesis. It will be convenient to characterize the polyhedral set $\mathcal{U}$ explicitly by $\mathcal{U} = \{v \text{ s.t. } Cv \leq c\}$ for a given matrix C and vector c. Then (1.11) becomes

$$\rho(z) = \sup_{Cu \leq c} \{\langle u, z\rangle - \frac{1}{2}\langle u, Mu\rangle\}. \tag{1.12}$$

Authors of [15] then uses this dual form for (1.10) to solve the resulting problem using an interior point method, generalizing the work of [13] for linear state space models.

1.4.5 Student's T Kalman Smoothers

While the class of PLQ penalties is fairly large and encompasses many of commonly used loss functions, one notable exception is the Student't T loss shown in Figure 1.4, which is not log-concave, and is used in robust statistical modeling for this reason [104].

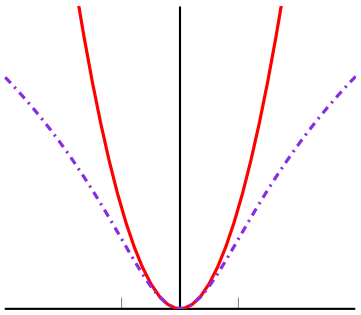


Figure 1.4: Comparison of least squares (red solid) and Student's T (violet dash dot).

The main attraction of the Student's T loss is that it has *sublinear* tails, making it more robust to outliers than any PLQ (or more generally any log-concave density). In the Kalman setting this allows for tracking sudden changes in the state (Student's T as process loss), or making the smoother robust to measurement outliers (Student's T as measurement loss). However as Student's T is not convex, great care must be taken. Authors of [18] develop methods to solve (1.6) where the loss functions are either least squares or Student's T. The technique used is a more general type of Gauss-Newton method, that develops a customized Hessian approximation adapted to the Student's T loss function.

1.4.6 Summary

The modeling capabilities of existing models are summarized in Table 1.1. There are a few notable gaps the most obvious of which is presence of singular noise in either the process or measurement error terms. This corresponds to additional knowledge that the error terms, w_k, v_k, live in a smaller dimensional subspace and thus the covariance matrices Q_k, R_k will

Method	Nonlinear Maps	Outliers	Sudden Changes	Singular Covariance	Nonconvex	Loss Functions
Classic	No	No	No	No[2]	No	$\|\|\cdot\|\|^2$
Nuisance Parameter	No	No	No	No	No	$\|\|\cdot\|\|^2$
Inequality	Yes	No	No	No	No	$\|\|\cdot\|\|^2$
Laplace	Yes	Yes	No	No	No	$\|\|\cdot\|\|^2$ or $\|\cdot\|_1$
IPSolve	No	Yes	Yes	No	No	Any PLQ
Student't T	Yes	Yes	Yes	No	Yes	T or $\|\|\cdot\|\|^2$

Table 1.1: Summary of relevent models.

not be full rank. Many models have naturally occurring singular covariance such as the DC motor example [113] and basic navigation models [7]. Additionally a number of modeling techniques such as fitting and removing constant bias, modeling time-dependent noise, and auto-regressive state dependence can be captured using a singular covariance matrix. A number of such scenarios will be explored in chapters 2,3 and 6.

Another gap in Table 1.1 is there is no general framework that allows for any PLQ loss when nonlinear maps are present. Finally estimating either dynamic parameters that govern the system or parameters that couple the states breaking traditional Kalman structure has not been explored in general Kalman smoothing problems.

1.5 Contributions

We present several modeling extensions of the Kalman filter/smoother that seek to fill the gaps highlighted in the previous section. We also develop numerical methods and demonstrate their usefulness on relevant examples using a number of synthetic and real data sets. In particular:

- We extend equation (1.10) to allow for singular Q_k, R_k. This has the added benefit of expanding the nuisance parameter estimation of [12] to a more general setting. Instead of an interior point method we use splitting methods and discuss advantages of this

approach for Kalman smoothing[3]. Additionally we demonstrate the increased flexibility of this new model by way of an extended example on a data set from an underwater autonomous vehicle. A number of problems with the data set are dealt with using the newly developed tools that are not be possible with existing methods.

- We develop a method for simultaneous state inference and parameter estimation when unknown parameters govern either the process or measurement dynamics[4]. Our technique here relies on smooth loss functions, but can be used with the nonconvex Student's T loss.

- We present a second order method on a general class of convex-composite PLQ functions and show how this can be used to solve the smoothing problem in a non-linear state space model[5]. This is a general model that captures all modeling capabilities in Table 1.1.

- We show how the Kalman filter can be combined with machine learning techniques to solve relevant problems in anomoly detection. We consider examples in aviation with several synthetic and real data sets.

- We demonstrate how the Kalman smoother in (1.5) can be blended with other models that break the traditional structure of the Kalman smoother. We use this to solve problems in underwater current estimation and compare our method with standard techniques in the field.

These results appear in the following articles and preprints which appear in full in the remaining chapters:

---

[3]Code is available at github.com/jonkerjo/Singular_Kalman_Python

[4]Code is available at github.com/UW-AMO/SSM_ParamsEstimation

[5]Code is available at github.com/UW-AMO/NC-PLQ

- J. Jonker, A.Y. Aravkin, J.V. Burke, G. Pillonetto, S. Webster. Fast Robust Methods for Singular State-Space Models. *Automatica*, 105:399-405, 2019

- J. Jonker, A.Y. Aravkin, J.V. Burke, G. Pillonetto, S. Webster. Roubst Singular Smoothers For Tracking Using Low-Fidelity Data. *Proceedings of Robotics: Science and Systems*, 2019

- J. Jonker, A. Shcherbina, R. Krishfield, L. Van Uffelen, A.Y. Aravkin, S. Webster. Preliminary Results in Current Profile Estimation and Doppler-aided Navigation for Autonomous Underwater Gliders. *OCEANS* 2019-Marseille. IEEE, 2019

- J. Jonker, P. Zheng, A.Y. Aravkin. Efficient Robust Parameter Identification in Generalized Kalman Smoothing Models. (submitted and under review), 2020

- B. M. de Silva, J. Callaham, J. Jonker, N. Goebel, J. Klemisch, D. McDonald, N. Hicks, J. N. Kutz, S. L. Brunton, A.Y. Aravkin. Physics-Informed Machine Learning for Sensor Fault Detection With Flight Test Data. (submitted and under review), 2020

1.6 Outline

In Chapter 2 we show how to find robust smoothing estimates when Q_k and R_k are singular. We also suggest using a DRS algorithm to exploit the unique structure of the smoothing problem and show that the convergence rate does not depend on the conditioning of the system. Because of this, the locally linear convergence rate of DRS can compete even with second order methods on problems where both are applicable. Furthermore, in Chapter 6 we apply this to an autonomous underwater vehicle with biased and discretized acceleration data to showcase the flexibility of the new model.

In Chapter 3 we allow process and measurement maps to $G(\theta)$ and $H(\theta)$ to depend on parameters, show how to estimate these parameters simultaneously with solving the robust singular smoothing problem. In the case of smooth loss functions we calculate second order

information using a variant of the implicit function theorem, use it to implement second order methods that exploit problem structure, and compare their numerical performance.

In Chapter 4 we show how a non-linear smoothing problem can be formulated as a specific case of a general class of convex-composite PLQ functions. We propose an adaptive quasi-Newton method in the general case and discuss how it applies to the smoothing problem specifically.

In Chapter 5 presents and application of Kalman filtering to a model developed using a data driven dynamic mode decomposition (DMD) method. The output is then used as a feature in a machine learning model for anomaly detection. We demonstrate this using a decision tree on a number of real and synthetic data sets.

In Chapter 7 we use the least squares viewpoint of (1.4) to combine smoothing ideas with other physical models that break the structure of the original Kalman smoother, by simultaneously solving for the state sequence of auxiliary unknown parameters. Specifically, we simultaneously solve for position and velocity (state parameters) as well as depth-related current (medium parameters) using hydrodynamic velocity, relative current, and two GPS measurements.

Chapter 2

FAST METHODS FOR ROBUST SINGULAR STATE-SPACE MODELS

Here we drop two crucial assumptions present in the classic model, Gaussian error and non-singular covariances. On their own either assumption can be removed without too much difficulty but without both of them a reformulation is needed. We propose a reformulation that leads to a convex constrained PLQ optimization problem and examine the use of a Douglas-Rachford (DRS) algorithm in this setting. This algorithm is known to have a locally linear rate of convergence for convex PLQ problems and we also show that this rate does not depend on the conditioning of the system. This allows DRS to compete with second order interior point methods in cases where both can be applied.

2.1 Introduction

The linear state space model is widely used in tracking and navigation [23], control [6], signal processing [7], and other time series [73, 150]. The model assumes linear relationships between latent states with noisy observations:

$$\begin{aligned} x_1 &= x_0 + w_1 \\ x_k &= G_k x_{k-1} + w_k, \quad k = 2, \ldots, N \\ y_k &= H_k x_k + v_k, \quad k = 1, \ldots, N, \end{aligned} \tag{2.1}$$

where x_0 is a given initial state estimate, $x_1, \ldots, x_N$ are unknown latent states with known linear process models G_k, and $y_1, \ldots, y_N$ are observations obtained using known linear models H_k.

The errors w_k and v_k are assumed to be mutually independent random variables with known covariances Q_k and R_k. In tracking and navigation, the end goal is the estimation of the

latent states $\{x_k\}$. In autocorrelated time series models (e.g. Holt-Winters c.f. [73], ARMA c.f. [150]), estimating the state is a necessary step to estimating additional parameters on which G_k, H_k, Q_k and R_k may depend. In both settings, estimating the state sequence $\{x_k\}$ efficiently is essential.

Singular Covariances. We are particularly interested in models where Q_k and R_k may be singular. These models arise in all settings where state-space formulations are used. The classic Kalman filter [82] and RTS smoother [124] assume that w_k, v_k are Gaussian, and find the minimum variance estimates of the state, conditioned on the observations [7]. More generally, the RTS smoother finds the linear minimum variance estimator. This procedure is well defined for singular covariances Q_k and R_k, and the smoother can be derived as a sequence of least squares projections [8]. However, when the noise is not Gaussian (e.g. in the presence of outliers), these estimates are not satisfactory; and far better estimates can be obtained through a maximum a posteriori (MAP) estimator [10]. Implementing a general MAP estimator for singular covariances requires a new approach.

General Kalman Smoothing. Classic Gaussian formulations fail when outliers are present in the data, are unable to track abrupt state changes, and cannot incorporate side information through constraints. To develop effective approaches in these cases, generalized Kalman smoothing formulations have been proposed in the last few years, see [10] and the references within. The conditional mean is no longer tractable to compute these estimates, and *maximum likelihood* (ML) formulations are much more natural. The general form of Kalman smoothing considered in [10] is given by

$$\min_{x \in X} \sum_{i=0}^{n} \rho_1(Q_k^{-1/2}(x_k - G_k x_{k-1})) + \rho_2(R_k^{-1/2}(y_k - H_k x_k)), \tag{2.2}$$

where ρ_1, ρ_2 are convex penalties, and $x \in X$ is a set of state-space constraints. The two approaches agree in the nonsingular Gaussian case, where (2.2) becomes a least squares (LS) problem that can be solved with classic RTS or Mayne-Fraser smoothing algorithms [10].

Contribution. We develop a new reformulation to extend (2.2) to singular covariance models Q_k and R_k, and implement a Douglas-Rachford splitting (DRS) algorithm to solve

this reformulation.

We analyze the DRS for the singular reformulation, and show that its convergence rate does not depend on the conditioning of the system. Even when the model is nonsingular, the new approach is potentially much faster than first-order and second-order methods for (2.2). The advantage increases as the models become more ill-conditioned; however the *local* linear rate means that initialization becomes very important.

The paper proceeds as follows. In Section 2.2 we discuss prior approaches to singular models. In Section 2.3, we develop a constrained reformulation of (2.2), building on early work of [117] for singular least squares. In Section 2.4, we show how to efficiently optimize a wide range of singular smoothing problems using DRS. The algorithm we use has a *local linear rate of convergence* for any piecewise linear-quadratic penalties ρ_1, ρ_2 in (2.2), and each iteration is efficiently and stably computed by exploiting dynamic problem structure. We compare the new algorithm to first-order methods, L-BFGS, and IPsolve, a toolbox specifically developed for PLQ Kalman smoothing (for nonsingular formulations).

2.2 Related Work

Several approaches in the literature deal with singular models. We give a brief description and references for each. To ground the discussion, consider tracking a particle moving along a smooth path in space, where state comprises velocity and position. Singular models arise naturally in this situation. We can model velocity as subject to error, and position as a deterministic integral:

$$\begin{aligned} x_{k+1} &= x_k + \Delta t \dot{x}_k \\ \dot{x}_{k+1} &= \dot{x}_k + \epsilon_k. \end{aligned} \tag{2.3}$$

Here, the process covariance matrix Q_k has rank one.

Using the original Kalman filter. In the linear Gaussian setting, the original Kalman filter does not require Q and R to be invertible. Applying the Kalman filter (and RTS smoother) will return the minimum variance estimate for singular innovation/measurement errors [7]. This result does not extend to the general context of (2.2), with robust losses,

sudden changes, and constraints.

Changing the model. A common approach is to modify the model to make Q_k, R_k nonsingular. Treating (2.3) as a discretization of a stochastic differential equation (SDE), many authors opt for a nonsingular error model [75, 115, 28, 22]

$$Q_k = \begin{bmatrix} \Delta t_k & \Delta t_k^2/2 \\ \Delta t_k^2/2 & \Delta t_k^3/3 \end{bmatrix},$$

derived by computing the variance of a discretized process noise term. The approach has limitations for navigation models with high-dimensional states driven by low-dimensional errors. The low-dimensional error structure should simplify estimation, but instead this approach introduces full-dimensional and ill-conditioned Q_k. In addition, making Q_k nonsingular is antithetical to state-space formulations for models such as ARMA, which use singularity to enforce auto-regressive constraints.

Pseudo-inverse with orthogonality constraints. The formulation that is closest to ours is that of [114], who replace the inverse of Q_k by a pseudo-inverse, and add orthogonality constraints. With potentially singular Q_k and R_k, the maximum likelihood estimate for the Gaussian/LS model can be formulated as

$$\begin{aligned} \min_x \sum_k \; & ||Q_k^{\dagger/2}(x_k - G_k x_{k-1})||^2 + ||R_k^{\dagger/2}(y_k - H_k x_k)||^2 \\ \text{s.t. } & Q_k^{\perp}(x_k - G_k x_{k-1}) = 0, \quad R_k^{\perp}(y_k - H_k x_k) = 0 \\ & \text{for all } k = 1, \ldots, N, \end{aligned} \tag{2.4}$$

see [10, Appendix A]. This requires computing both the pseudo-inverse and orthogonality constraints.

Constrained reformulation. The reformulation we choose was first used by Paige [117]. Given the singular least squares problem

$$\min_x \|Q^{\dagger/2}(Ax - b)\|^2 \quad \text{s.t.} \quad Q^{\perp}(Ax - b) = 0,$$

we can instead write it as

$$\min_{x,u} \|u\|^2 \quad \text{s.t.} \quad \mathrm{Q}^{1/2} u = Ax - b. \tag{2.5}$$

It is easy to see (2.4) and (2.5) are equivalent; the latter is more elegant, and only requires computing a root of Q, rather than using both Q and $Q^\dagger$. When Q is invertible, we can eliminate u from both formulations and reduce to a least squares problem in x. Splitting the affine constraint from the original penalty has theoretical and practical advantages for general Kalman smoothing, as shown in the next sections.

2.3 General Singular Kalman Smoothing

Following the ideas proposed by [114], we introduce variables u_k for the normalized process innovations, and t_k for the normalized residuals. We also introduce a penalty ρ_3 for the states. In the examples we consider, ρ_3 is an indicator function for the known feasible regions X_k:

$$\rho_3(x_k) = \begin{cases} 0 & x_k \in X_k \\ \infty & x_k \notin X_k \end{cases}.$$

The reformulated singular Kalman smoothing problem is given by

$$\begin{aligned} \min_{u,t,x} & \sum_{k=1}^{N} \rho_1(u_k) + \rho_2(t_k) + \rho_3(x_k) \\ \text{s.t.} \quad & Q_k^{1/2} u_k = G_k x_{k-1} - x_k \\ & R_k^{1/2} t_k = y_k - H_k x_k. \end{aligned} \tag{2.6}$$

This problem is equivalent to (2.2) when Q_k and R_k are nonsingular. For singular models, (2.6) requires only that roots $Q^{1/2}$ and $R^{1/2}$ are available.

Structure-preserving Reformulation. We now rewrite (2.6) into a more compact form. Define

$$\begin{aligned} D_i &= \begin{pmatrix} Q_i^{1/2} & 0 & I \\ 0 & R_i^{1/2} & H_i \end{pmatrix} \text{ for } i = 1, \dots N, \\ B_j &= \begin{pmatrix} 0 & 0 & -G_{j+1} \\ 0 & 0 & 0 \end{pmatrix}, \text{ for } j = 1, \dots, N-1, \end{aligned} \tag{2.7}$$

and let

$$A = \begin{pmatrix} D_1 & 0 & \dots & 0 \\ B_1 & D_2 & 0 & \vdots \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & B_{N-1} & D_N \end{pmatrix}. \tag{2.8}$$

Define also

$$\begin{aligned} z^T &= \begin{pmatrix} u_1^T & t_1^T & x_1^T & \dots u_N^T & t_N^T & x_N^T \end{pmatrix} \\ \hat{w}^T &= \begin{pmatrix} x_0^T & y_1^T & 0 & y_2^T & \dots & 0 & y_N^T \end{pmatrix}. \end{aligned} \tag{2.9}$$

Now we can write (2.6) compactly as

$$\begin{aligned} &\min_z \quad \rho(z) \quad \text{s.t. } Az = \hat{w}, \\ &\rho(z) = \sum_{k=1}^{N} \rho_1(u_k) + \rho_2(t_k) + \rho_3(x_k). \end{aligned} \tag{2.10}$$

The order of blocks in z is chosen to the constraint matrix A in (2.8) lower block bi-diagonal. The constraint $Az = \hat{w}$ raises a natural question: when is a singular Kalman smoothing model solvable? Clearly we want $\hat{w} \in \text{Ran}(A)$, but we want this condition to hold for any realization of the data $\hat{w}$, so we want to know when A is surjective. We can characterize this condition precisely in terms of a simple conditions on the individual blocks R_i, Q_i, H_i.

Theorem 2.3.1 (Surjectivity of A)**.** *The following are equivalent.*

1. *A is surjective.*

2. *Each block D_i is surjective.*

3. $\text{null}\left(\begin{bmatrix} Q_i^{1/2} & 0 \\ 0 & R_i^{1/2} \end{bmatrix}\right) \subset \text{Ran}\left(\begin{bmatrix} I \\ H_i \end{bmatrix}\right)$ *for all i.*

4. *$R_i + H_i \left(I - (Q_i + I)^{-1}\right) H_i^T$ is invertible for all i.*

Proof. Conditions $2, 3, 4$ are easily seen to be equivalent. To see that 2 and 3 are equivalent, note that the matrix

$$\begin{bmatrix} Q_i^{1/2} & 0 \\ 0 & R_i^{1/2} \end{bmatrix}$$

is symmetric, so its nullspace is perpendicular to its range. Therefore surjectivity of D_i is equivalent to the condition that the range of $\begin{bmatrix} I \\ H_i \end{bmatrix}$ covers this nullspace.

To see the equivalence of 2 and 4, recall that B is surjective if and only if BB^T is invertible, so D_i is surjective exactly when the matrix

$$\begin{bmatrix} Q_i + I & H_i^T \\ H_i & R_i + H_i H_i^T \end{bmatrix}$$

is invertible. $Q_i + I$ is always invertible, so invertibility of the block 2×2 matrix is equivalent to the invertibility of the Schur complement $R_i + H_i \left(I - (Q_i + I)^{-1}\right) H_i^T$.

It remains to show that conditions 1 and 2 are equivalent. We proceed by induction on N. The base case is trivial, since for $N = 1$, $A = D_1$. For the inductive case, consider that for $N = k$ the result holds, and write the $N = k + 1$ case as

$$\begin{bmatrix} A_k & 0 \\ [0 \quad B_k] & D_{k+1} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix},$$

and assume that A_k is surjective. We then know that there exists z_1 that satisfies $A_k z_1 = w_1$. The second row can now be written explicitly as

$$D_{k+1} z_2 = w_2 + G_{k+1} x_k,$$

where x_k is the last component of z_1. Thus A_{k+1} is surjective exactly when D_{k+1} is, as desired. □

Remark 2.3.2. *In practice it is not strictly necessarry for A to be surjective, in fact if $y_i \in \mathrm{Ran}(H_i)$ for all i, then $\hat{w} \in \mathrm{Ran}(A)$. It is also worth noting that if the data is not in the range of the measurment maps, a sparse penalty could be used instead and the same convergence rates would hold.*

2.4 Douglas-Rachford Splitting for General Singular Kalman Smoothing

Consider problem (2.10) as a sum of two functions, $\rho + g$, with ρ as in (2.10) and g the indicator function of the affine constraint $Az = \hat{w}$:

$$g(z) = \begin{cases} 0 & Az = \hat{w} \\ \infty & Az \neq \hat{w} \end{cases}. \tag{2.11}$$

Douglas-Rachford splitting (DRS) is a classic algorithm for this problem. For a convex function f, define the proximity operator (see e.g. [48]) as

$$\text{prox}_{\alpha f}(\zeta) = \arg\min_x \frac{1}{2\alpha}\|\zeta - x\|^2 + f(x).$$

The DRS algorithm for (6.4) detailed in Algorithm 1. For more on splitting methods and their convergence rates see the survey [49].

Algorithm 1 Douglas-Rachford Splitting (DRS)

Require: Initialize at any z^0, ζ^0.

1: **loop**

2: $\quad z^k = \text{prox}_{\tau g}(z^{k-1} - \tau\zeta^{k-1})$

3: $\quad \zeta^k = \text{prox}_{\sigma\rho^*}(\zeta^{k-1} + \sigma(2z^k - z^{k-1}))$

return z^k

Implementing DRS in our case requires computing two proximity operators at each iteration. One proximity operator is prox_{ρ^*}, where ρ^* denotes the *convex conjugate*:

$$\rho^*(y) = \sup_x \langle y, x\rangle - \rho(x)$$

The prox of of a function is related to the prox of its conjugate by Moreau's decomposition:

$$\text{prox}_\rho(x) + \text{prox}_{\rho^*}(x) = x.$$

Thus it suffices to compute prox_ρ. The function ρ captures all user-supplied models, including losses used process and measurement transitions, as well as penalties or constraints on

the state, ρ_1, ρ_2 and ρ_3. The proximity operators of these individual elements must be provided; then prox_ρ is a stack of these input functions. Proximity operators for many common functions are easily available [48], and we include a small library with our implementation[1]. The second proximity operator is prox_g, which is independent of user choice for process, measurement, and prior models:

$$\text{prox}_g(\eta) = \arg\min_{Az=\hat{w}} \frac{1}{2}\|\eta - z\|^2.$$

This is a simple quadratic with affine constraints, with optimality conditions given by

$$\begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} z \\ \nu \end{bmatrix} = \begin{bmatrix} \eta \\ \hat{w} \end{bmatrix}.$$

There are many ways to solve this system. We opt to reduce the problem to solving a block tridiagonal system:

$$\begin{bmatrix} I & A^T \\ 0 & AA^T \end{bmatrix} \begin{bmatrix} z \\ \nu \end{bmatrix} = \begin{bmatrix} \eta \\ A\eta - \hat{w} \end{bmatrix}$$

We solve $AA^T\nu = A\eta - \hat{w}$, then back-substitute to get the optimal z. The system AA^T does not change over iterations; only the right hand side changes. We can therefore compute a single factorization, then use it in each iteration. Since A is block bidiagonal (6.6), AA^T is block tridiagonal; when A is surjective, AA^T is nonsingular, and we can find a lower block diagonal Cholesky factorization L with $LL^T = AA^T$:

$$AA^T = \begin{bmatrix} a_1 & b_1^T & & \\ b_1 & a_2 & b_2^T & \\ & b_2 & a_3 & b_3^T \\ & & b_3 & a_4 \end{bmatrix}, \quad L = \begin{bmatrix} c_1 & & & \\ d_1 & c_2 & & \\ & d_2 & c_3 & \\ & & d_3 & c_4 \end{bmatrix} \tag{2.12}$$

The factorization is obtained by a simple variant of the block-forward Thomas algorithm, with $O(n^3N)$ arithmetic operations required. Once L has been pre-computed, we need only

[1] https://github.com/UW-AMO/KalmanJulia.

$O(n^2N)$ arithmetic operations to solve $LL^T\nu = Ac - \hat{w}$ for any right hand side. This is the same complexity as that of a matrix-vector multiply with A.

Local Linear Rate. When ρ is piecewise linear-quadratic [126, 15], the DRS algorithm converges locally linearly to a solution, see Figure 2.2. More precisely, there is a real number $R > 0$ such that if $||\eta^K - \eta^*|| < R$ then there is a constant κ with $0 < \kappa < 1$ such that for all $k > K$,

$$\|\eta^{k+1} - \eta^*\| < \kappa\|\eta^k - \eta^*\|,$$

where $\eta = \begin{bmatrix} z & \zeta \end{bmatrix}^T$, is the primal and dual pair. Before proceeding with the main theorem we need some notation. Define

$$D\eta \mapsto \begin{bmatrix} \partial g(z) \\ \partial \rho^*(\zeta) \end{bmatrix}, \quad M = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}, \quad H = \begin{bmatrix} \frac{1}{\tau}I & 0 \\ -2I & \frac{1}{\sigma}I \end{bmatrix}.$$

Define the nonlinear operator T by

$$T = (H + D)^{-1}(H - M). \tag{2.13}$$

T captures the iteration in Algorithm 1, which can be written as $\eta^k = T\eta^{k-1}$, for $\eta = \begin{bmatrix} z^T, \zeta^T \end{bmatrix}^T$. Then we have the following lemma.

Lemma 2.4.1. *Suppose that $\tau, \sigma < 1$. Then*

$$||T\eta - \eta||^2_{H-M} \leq \langle \eta^* - \eta, (H - M)(T\eta - \eta)\rangle$$

where η^* is such that $0 \in (D + M)\eta^*$.

Proof. This proof is similar to that of [92] but included in full detail here to emphasize independence from the conditioning of the system. As D is monotone we have

$$\langle \eta^* - T\eta, D\eta^* - DT\eta\rangle \geq 0.$$

As $0 \in (D + M)\eta^*$ this implies

$$\langle \eta^* - T\eta, -M\eta^* - DT\eta\rangle \geq 0.$$

Now $DT\eta = DT\eta + HT\eta - HT\eta = (H - M)\eta - HT\eta$. Thus

$$0 \leq \langle \eta^* - T\eta, -M\eta^* + HT\eta - (H - M)\eta \rangle$$

$$= \langle \eta^* - T\eta, -M(\eta^* - \eta) + H(T\eta - \eta) \rangle$$

$$= \langle \eta^* - \eta, -M(\eta^* - \eta) + H(T\eta - \eta) \rangle$$

$$+ \langle \eta - T\eta, -M(\eta^* - \eta) + H(T\eta - \eta) \rangle.$$

By definition of M we have

$$\langle M\eta, \eta \rangle = 0$$

for any η. Therefore

$$0 \leq \langle \eta^* - \eta, H(T\eta - \eta) \rangle + \langle \eta - T\eta, -M(\eta^* - \eta) \rangle$$

$$+ \langle \eta - T\eta, H(T\eta - \eta) \rangle - \langle \eta - T\eta, M(T\eta - \eta) \rangle$$

$$= \langle \eta^* - \eta, H(T\eta - \eta) \rangle + \langle \eta - T\eta, -M(\eta^* - \eta) \rangle - ||T\eta - \eta||^2_{H-M}$$

$$= \langle \eta^* - \eta, H(T\eta - \eta) \rangle + \langle M(\eta - T\eta), \eta^* - \eta \rangle - ||T\eta - \eta||^2_{H-M}$$

$$= \langle \eta^* - \eta, (H - M)(T\eta - \eta) \rangle - ||T\eta - \eta||^2_{H-M}.$$

□

Theorem 2.4.2. *If $\tau, \sigma < 1$, Algorithm 1 converges with a locally linear rate that does not depend on the conditioning of the matrix A.*

Proof. Suppose $\{\eta^j\}_{j=1}^{\infty}$ are defined by $\eta^{k+1} = T\eta^k$. Then $\lim_{j\to\infty} \eta^j = \eta^*$ and

$$||\eta^{k+1} - \eta^*||^2_{H_M} = ||(T\eta^k - \eta^k) + (\eta^k - \eta^*)||^2_{H-M}$$

$$= ||T\eta^k - \eta^k||^2_{H-M} + ||\eta^k - \eta^*||^2_{H-M} + 2\langle (H - M)T\eta^k - \eta^k, \eta^k - \eta^* \rangle$$

$$\leq ||T\eta^k - \eta^k||^2_{H-M} + ||\eta^k - \eta^*||^2_{H-M} - 2||T\eta^k - \eta^k||^2_{H-M}$$

by Lemma 2.4.1. Thus

$$||\eta^{k+1} - \eta^k||^2_{H-M} \leq ||\eta^k - \eta^*||^2_{H-M} - ||\eta^{k+1} - \eta^*||^2_{H-M}$$

Now by [92] $D+M$ is metrically subregular so there is a neighborhood $\mathcal{U}$ of η^* and a constant $c_1 > 0$ so that if $\eta \in \mathcal{U}$ then

$$||\eta - \eta^*||^2 \leq c_1||y||^2, \quad \text{for all } y \in (D+M)\eta.$$

Note that

$$\partial g(z) = \begin{cases} 0, & \text{if } Az = \hat{w} \\ \emptyset, & \text{else} \end{cases}$$

so c_1 depends only on the affine subset determinted by the equation $Az = \hat{w}$, not on the condition number of A. Define

$$v^k = -(H-M)(\eta^{k+1} - \eta^k)$$

so $v^k \in (D+M)\eta^{k+1}$ and thus for large enough k,

$$||\eta^{k+1} - \eta^*||^2 \leq c_1||\eta^{k+1} - \eta^k||^2.$$

Now by the inequality from above, for another constant $c_2 > 0$ that depends on $H - M$,

$$||\eta^{k+1} - \eta^*||^2_{H-M} \leq c_1c_2\left[||\eta^k - \eta^*||^2_{H-M} - ||\eta^{k+1} - \eta^*||^2_{H-M}\right]$$

Thus there is a $\kappa \in (0,1)$ ($\kappa = \frac{c_1c_2}{1+c_1c_2}$) that does not depend on the condition number of A such that

$$||\eta^{k+1} - \eta^*||^2_{H-M} \leq \kappa||\eta^k - \eta^*||^2_{H-M}$$

□

Comparison on Smooth Nonsingular Problems. If the covariances, Q, R are nonsingular and the penalties $\rho_{1,2}$ are $\mathcal{C}^1$-smooth, then the Kalman smoothing problem can be written as a smooth convex problem. In this case the same reformulation will work and

Algorithm 1 will still give a local linear rate. However more common algorithms such as gradient descent and L-BFGS can also be applied. We compare the performance of these three algorithms to track a particle moving along a smooth path with $N = 200$ and $n = 2$. We use non-singular Q_k, and Huber penalty functions.

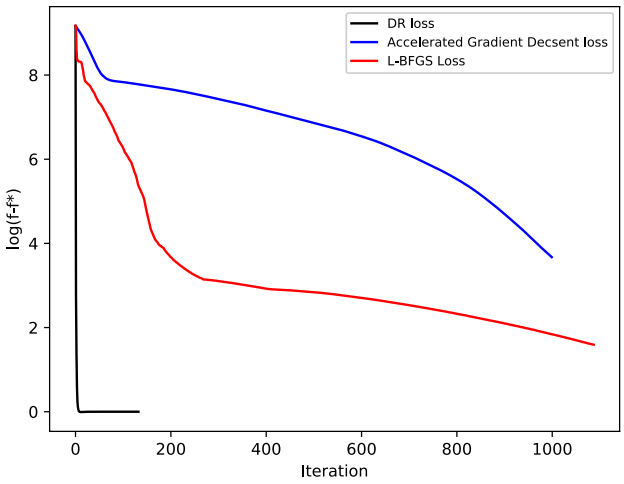


Figure 2.1: Objective vs. iteration counts of Algorithm 1 for (2.10) (black), vs. accelerated gradient descent (AGD) (blue) and L-BFGS (red) for (2.2). Both ρ_1 and ρ_2 are Huber losses, with $\rho_3 \equiv 0$, $N = 200, n = 2$ and Q, R nonsingular, so (2.10) and (2.2) are equivalent. All iterations require $O(n^2N)$ operations. DRS splitting is much faster than methods with linear convergence rates and similar iteration complexity.

As seen in Figure 2.1, Algorithm 1 for (2.10) converges far faster than either accelerated gradient descent or LBFGS method on the equivalent nonsingular smoothing formulation (2.2) . This is because its convergence rate does not depend on the condition number of the matrix A, so each iteration makes a lot of progress, and we can keep the complexity of each iteration at $O(n^2N)$, same as for a matrix-vector multiply needed for a gradient evaluation, if we factor the sparse block tridiagonal matrix AA^T once at the start of the

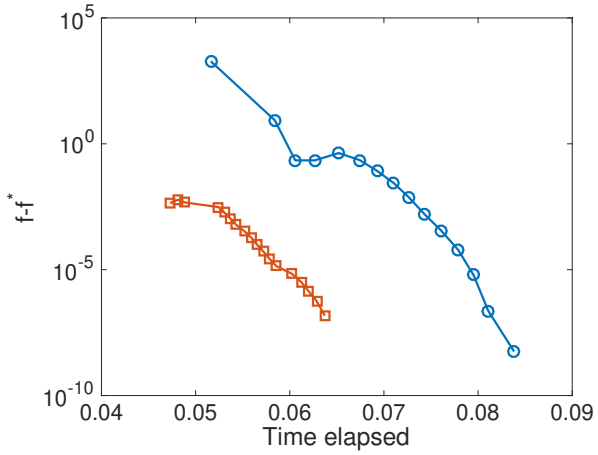


Figure 2.2: Timed run of Algorithm 1 vs. IPsolve for the same setup as presented in Figure 2.1. At this scale, we see the locally linear convergence rate of the DRS. Even though IPsolve has superlinear rate, DRS wins because the slope of the rate is very steep, and each iteration is fast. By the time DRS is done, IPsolve has had time for only taken a few iterations.

algorithm.

We also compare with the second-order interior point method, implemented in the IPsolve package[2]. Use-cases and performance of IPsolve for nonsingular Kalman smoothing is discussed in [10]. The results are shown in Figure 2.2, where IPsolve and DRS for the equivalent reformulation are compared for the nonsingular Huber model. Even though DRS has at best a linear rate, the constants are very good, as they do not depend on the conditioning of the Kalman smoothing problem. The other advantage is that DRS can use a pre-factorized matrix, while IPsolve has to solve a modified linear system every time; there is no simple strategy to pre-factor as with DRS.

The numerical experiments suggest that Algorithm 1 should be used regardless of whether

[2] https://github.com/UW-AMO/IPsolve.

Q and R are singular or not. In the next section, we focus on a rich class of singular noise models found in navigation.

2.5 Navigation Models

Autonomous navigation requires high-fidelity tracking using occasional GPS and frequent depth/height, gyrocompass, and linear acceleration data. Gyro, compass, and linear acceleration are readily available from inertial measurement units (IMUs).

In this section, we develop a simple kinematic model that is trivially applicable to any vehicle, and particularly useful for applications such as survey work where accelerations are heavily damped and autonomous vehicles often travel in long straight lines. When the attitude is known or changing slowly, the model can be linearized effectively and the situation simplifies considerably.

Linear Singular Navigation Model. For a vehicle that is well-instrumented in attitude, the uncertainty in position (and the x-y states in particular) is typically orders of magnitude larger than the uncertainty in attitude. We simplify the vehicle process model to track only position states (x, y, z), while assuming that the attitude states (r, p, h) are directly available from the most recent sensor measurements. To make the model linear, the position and its derivatives are referenced to the local-level frame.

To incorporate linear acceleration measurements from an inertial measurement unit (IMU), we must track both linear velocities and linear acceleration in the state vector. This leads to the augmented state

$$x_s = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}]^\top. \tag{2.14}$$

The linear kinematic process model is given by

$$\dot{x}_s = \underbrace{\begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix}}_{F_s} x_s + \underbrace{\begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix}}_{G_s} w_s, \tag{2.15}$$

where $w_s \sim \mathcal{N}(0, Q_s)$ is zero-mean Gaussian noise.

The linear process model (2.15) is usually discretized using a Taylor series:

$$x_{s_{k+1}} = F_{s_k} x_{s_k} + w_{s_k} \tag{2.16}$$

$$\begin{aligned} F_{s_k} = e^{F_s T} &= I + F_s T + \frac{1}{2!} F_s^2 T^2 + \cancelto{0}{\frac{1}{3!} F_s^3 T^3} + \cdots \\ &= \begin{bmatrix} I & IT & \frac{1}{2} IT^2 \\ 0 & I & IT \\ 0 & 0 & I \end{bmatrix} \end{aligned}$$

where the higher order terms are *identically zero* because of the structure of F_s, resulting in a simple closed-form solution for F_{s_k}. The discretized process noise

$$w_{s_k} = \int_0^T e^{F_s (T-\tau)} G_s w_s(\tau) d\tau, \tag{2.17}$$

is a zero-mean Gaussian, with covariance given by

$$Q_{s_k} = \begin{bmatrix} \frac{1}{3} T^3 & \frac{1}{2} T^2 & 0 \\ \frac{1}{2} T^2 & T & 0 \\ 0 & 0 & 0 \end{bmatrix} Q_s, \tag{2.18}$$

Model (2.18) forces the acceleration to be 0 across the entire model because the lower right corner is set to 0. As a result, the initialized track can be biased away from the data by a fixed velocity, obtained by finding the slope from the most recent position data.

Instead, we model the covariance as if the error were the next term in the Taylor series approximation, a technique suggested by [22]. More precisely we set covariance to be the outer product, $\Gamma^T \Gamma$ where

$$\Gamma = \begin{bmatrix} \frac{1}{3!} IT^3 & \frac{1}{2!} IT^2 & IT \end{bmatrix}$$

This leads to a rank 3 covariance for a 9×9 matrix for a model that comprises position, velocity, and acceleration in 3D space.

Measurement Models for the IMU. Any navigation system that relies on an IMU needs

occasional measurements that inform the position (e.g. GPS), otherwise the error in position estimates grows without bound. We are given these data from a separate source, sampled at a lower update rate than that of the IMU. For any s where such data is available, we have the measurement model

$$H_s = \begin{bmatrix} I_{3\times 3} & 0_{3\times 6} \\ 0_{3\times 6} & R(\varphi) \end{bmatrix}, \quad z_s = \begin{bmatrix} b^\top & \ddot{x}_{\text{meas}} & \ddot{y}_{\text{meas}} & \ddot{z}_{\text{meas}} \end{bmatrix}^\top.$$

with $R(\varphi)$ a known coordinate transform based on instrumented roll, pitch, and yaw. If there is no position data measured at time s then we use the model

$$H_s = \begin{bmatrix} 0_{3\times 3} & 0_{3\times 6} \\ 0_{3\times 6} & R(\varphi) \end{bmatrix}, \quad z_s = \begin{bmatrix} 0 & \ddot{x}_{\text{meas}} & \ddot{y}_{\text{meas}} & \ddot{z}_{\text{meas}} \end{bmatrix}^\top.$$

The covariance used for measurement data depends on whether there was position data available:

$$R_s = \begin{bmatrix} 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & r_s I_{3\times 3} \end{bmatrix}, \quad R_s = \begin{bmatrix} U_s & 0_{3\times 3} \\ 0_{3\times 3} & r_s I_{3\times 3} \end{bmatrix}$$

wher ethe top 3×3 block is either 0 (position not available) or U, a diagonal matrix reflecting position uncertainty (position is available). The scalar r_s models uncertainty in IMU measurements.

Synthetic Tracking Example. To keep the presentation brief, we use the models described above to track a particle moving along the path

$$y = \sin(t/2) + t/10$$

with $\Delta t = .1$. At every time point we observe acceleration and every 1 second observe position, both with small Gaussian errors ($\mathcal{N}(0, 1)$). With probability $p = .3$ the position observation is corrupted further by large errors (outliers) by drawing from a $\mathcal{N}(0, 4)$ distruibution. We test the model with both standard and robust penalties. The position estimates plotted against observed data are shown in Figure 2.3 and acceleration estimates along with data are shown in Figure 2.4.

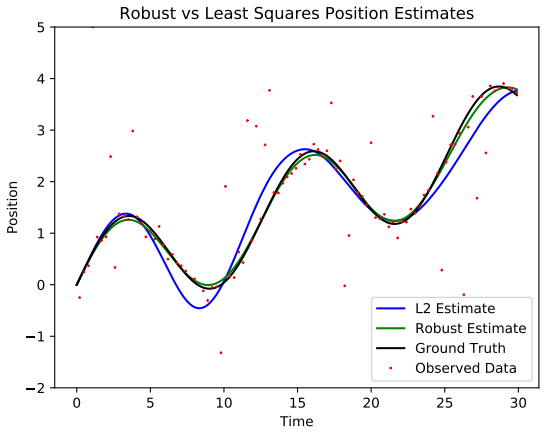


Figure 2.3: Position estimate for both robust and standard penalties

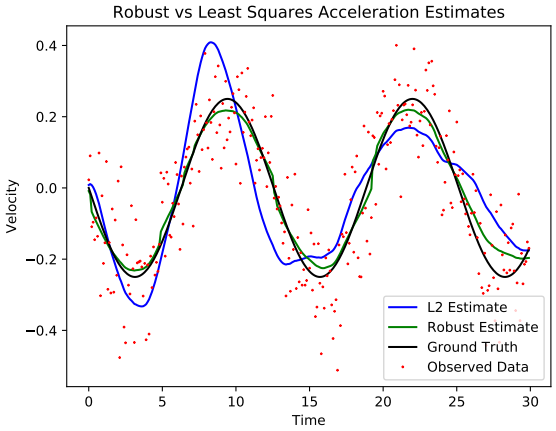


Figure 2.4: Acceleration estimate for both robust and standard penalties

When the data is corrupted, the results for the (singular) robust formulation (for both position and velocity) are much closer to the synthetic ground truth than the solutions of the singular least squares model.

Chapter 3

EFFICIENT ROBUST PARAMETER IDENTIFICATION IN GENERALIZED KALMAN SMOOTHING MODELS

In this work we consider simultaneously solving the smoothing problem and estimating parameters in the process and measurement maps, $G_k(\theta), H_k(\theta)$. Following the reformulation of Chapter 2 to allow for singular covariances and a robust estimate puts the parameter θ only in the constraints. To deal with this we restrict ourselves to the case where the penalty functions are smooth and use a variant of the implicit function theorem to explicitly compute second derivatives of a value function $\nu(\theta)$. We compare different first and second order methods using these derivatives including the Student's T case where the penalty functions are smooth but not convex

3.1 Introduction.

The linear state space model is widely used in tracking and navigation [23], control [6], signal processing [7], and other time series [73, 150]. The model assumes linear relationships between latent states with noisy observations:

$$\begin{aligned} x_k &= G_k x_{k-1} + \epsilon_k^p, \qquad k = 1, \ldots, N, \\ z_k &= H_k x_k + \epsilon_k^m, \qquad k = 1, \ldots, N, \end{aligned} \tag{3.1}$$

where x_0 is a given initial state estimate, $x_1, \ldots, x_N$ are unknown latent states with known linear process models G_k, and $z_1, \ldots, z_N$ are observations obtained using known linear models H_k. The errors ϵ_k^p and ϵ_k^m are assumed to be mutually independent random variables with covariances Q_k and R_k. These covariances may be singular to capture standard autoregressive structures.

In many applications the models G_k, H_k, Q_k, R_k are specified up to model parameters θ.

We restrict out attention to formulations where variances Q_k, R_k are known, while $G_k(\theta)$ and $H_k(\theta)$ are $\mathcal{C}^2$ mappings of θ. This captures smoothing parameters in Holt-Winters c.f. [73], autoregressive and moving average parameters in ARMA c.f. [150], and unknown dynamic parameters in navigation models. In most of these models, G and H are affine functions of unknown parameters θ.

In this chapter, we focus on MAP estimators, optimizing the likelihood in both the state and the parameters. The appropriateness of MAP estimators always depends on the problem and context, e.g. joint MAP estimation of states and parameters can be biased [74], and the bias can be addressed through corrections or alternative formulations, including marginal likelihood [26]. Nonetheless, MAP estimates are the standard for many of the motivating applications, and the techniques developed here can be extended to other settings (such as marginal likelihood) as we discuss in the conclusions.

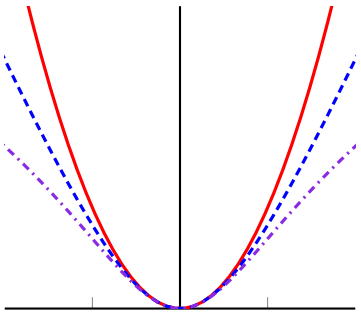


Figure 3.1: Common smooth loss functions: least squares (red solid), Hybrid (blue dashed), and Student's T (violet dash dot).

Standard models assume the errors ϵ_k^p and ϵ_k^m are Gaussian, which gives rise to the least squares penalty in the inference problem, see the red solid curve in Figure 3.1. Changing the observation model to the Hybrid (blue dash) or Student's T loss (violet dash dot) robustifies model estimates in the face of measurement outliers. Analogous changes to the innovations model allows the framework to track sudden changes.

Prior art for robust system identification includes adding a normalization term to the

parameter estimate in a classic least square solver [121] and explicitly modeling noise as a mixture of heavy tailed pdfs [32]. ARMA systems specifically have been reformulated using SVMs to compute robust parameter estimates [128]. Partial minimization techniques have been used for range of inverse problems [20, 19]. Here we consider state space models with potentially singular covariance terms, and obtain MAP estimates for parameters and state variables simultaneously using second order methods.

We can design efficient algorithms if we (1) minimize efficiently with respect to the state sequence $\{x_k\}$ and (2) obtain derivatives of the resulting value function, which depends on θ. Higher order derivatives enable second order methods to optimize this value function. We compute these derivatives for general models that cover motivating applications.

Example: Structural Unemployment Rate. We are interested in fitting parameters within structural unemployment rate models, see e.g. [109]. The state vector

$$x_k = \begin{bmatrix} u_{k-1} & u^c_{k-1} & u_k & u^c_k \end{bmatrix}^T \tag{3.2}$$

tracks total (u) and 'cyclic' (u^c) unemployment using the auto-regressive model

$$G_k = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -l_1 & 0 & 1+l_1 & 0 \\ 0 & 1/2 - l_2 & 0 & l_2 \end{bmatrix}, \quad \epsilon^p_k = \begin{bmatrix} 0 \\ 0 \\ \epsilon^1_k \\ \epsilon^2_k \end{bmatrix}, \tag{3.3}$$

$$H_k = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & \gamma/2 & 0 & \gamma/2 \end{bmatrix}, \quad \epsilon^m_k = \begin{bmatrix} \epsilon^3_k \\ \epsilon^4_k \end{bmatrix}. \tag{3.4}$$

Here, l_1 and l_2 are auto-regressive parameters while γ is an unknown measurement parameter. Unemployment rates can experience fast changes, so we need a heavy tailed model for innovations. To solve the full problem, we must

1. Estimate states $\{x_k\}$ as well as parameters l_1, l_2, γ.

2. Account for the singular process covariance Q.

3. Use non-Gaussian losses (e.g. Hybrid and Student's T) for ϵ_k^p to track fast rate changes.

The paper proceeds as follows. In Section 3.3, we review optimization formulations for state and model parameter inference using singular and nonsingular covariance models, and introduce the *value function* which depends only on the model parameters, as e.g. l_1, l_2, γ above. In Sections 3.4 and 3.5, we look in detail at nonsingular and singular Kalman smoothing models, and obtain existence results and formulas for first and second derivatives of the value function. Finally, in Section 4.6 we present use cases that show how to efficiently obtain structural parameters when using general losses and singular covariance structure.

3.2 Notations and Preliminaries

We first introduce notation and key definitions.

Superscript and subscript We use superscripts to distinguish process (p) and measurement (m) model variables and subscripts to represent partial derivatives ($\theta, y, r, \ldots$) and the index in the Kalman model (k).

Loss functions We use the following loss functions:

- Least squares: $\ell(r) = \frac{1}{2}\|r\|^2$.
- Hybrid: $\ell(r;\nu) = \sum_i \sqrt{r_i^2 + \nu^2} - \nu$.
- Student's T: $\ell(r;\nu) = \sum_i \ln(1 + r_i^2/\nu)$.

3.3 Differentiating Implicit Functions

In this section, we introduce a general theoretical result for calculating the derivatives for implicit functions in an optimization context. We then specialize this general theorem to nonsingular and singular state space models (SSM) in the next section.

Consider a $\mathcal{C}^2$-smooth function, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, where in SSM models we denote parameters by θ in $f(\theta, y)$ and the states together with any auxiliary variables (such as dual variables) by y. The appropriate stationarity condition is given by

$$\mathcal{H}(\theta, y) := f_y(\theta, y) = 0. \tag{3.5}$$

For any given θ, the optimal estimate $y(\theta)$ is obtained by solving the equation $\mathcal{H}(\theta, y) = 0$; in particular $y(\theta)$ depends on θ implicitly. When f is convex in y, (3.5) is equivalent to global optimality. More generally, $y(\theta)$ is any solution that satisfies the equation (3.5), and the standard implicit function theorem (below) provides weaker conditions under which we know $y(\theta)$ exists as a function locally. We introduce a variant of the implicit function theorem presented by [27] to characterize the structure of this implicit dependence. These developments do not require convexity of f.

Theorem 3.3.1 (Implicit Functions and Derivatives)**.** *Suppose that $\mathscr{U} \subset \mathbb{R}^n$ and $\mathscr{V} \subset \mathbb{R}^m$ are open, $\mathcal{H} : \mathscr{U} \times \mathscr{V} \to \mathbb{R}^m$ is continuously differentiable. If there exists $\overline{\theta} \in \mathscr{U}$ and $\overline{y} \in \mathscr{V}$, such that $\mathcal{H}(\overline{\theta}, \overline{y}) = 0$ and $\mathcal{H}_y(\overline{\theta}, \overline{y})$ is invertible. Then there exists (if necessary we choose $\mathscr{U}$ and $\mathscr{V}$ to be small neighborhood of $\overline{\theta}$ and $\overline{y}$ to guarantee the existence) a $\mathcal{C}^1$ mapping $Y : \mathscr{U} \to \mathscr{V}$ satisfying $Y(\overline{\theta}) = \overline{y}$, and $\mathcal{H}(\theta, Y(\theta)) = 0$ for all θ in $\mathscr{V}$.*

Moreover, we have the formula

$$Y_\theta(\theta) = -\mathcal{H}_y(\theta, Y(\theta))^{-1}\mathcal{H}_\theta(\theta, Y(\theta)).$$

This variant of the implicit function theorem gives an explicit formula for $Y(\theta)$, which we use in further development.

When the function $Y(\theta)$ as above exists, we can define the value function

$$v(\theta) = f(\theta, Y(\theta)), \text{ for } \theta \in \mathcal{U} \tag{3.6}$$

Our goal is to compute first and second derivatives of v, which are summarized in the following corollary.

Corollary 3.3.2 (Derivatives of the Value Function (3.6))**.** *Under the assumptions of Theorem 3.3.1, and using $\overline{y}$ to represent the y obtained by evaluating $Y(\overline{\theta})$, we have,*

$$\begin{aligned} v_\theta(\overline{\theta}) =& f_\theta(\overline{\theta}, \overline{y}) \\ v_{\theta\theta}(\overline{\theta}) =& f_{\theta\theta}(\overline{\theta}, \overline{y}) - \mathcal{H}_\theta(\overline{\theta}, \overline{y})^\top \mathcal{H}_y(\overline{\theta}, \overline{y})^{-1} \mathcal{H}_\theta(\overline{\theta}, \overline{y}). \end{aligned} \tag{3.7}$$

These derivations are along the lines of those presented by Bell and Burke [27] and are mainly given here for a self-contained exposition.

We now compute analytic expressions of derivatives with respect to model parameters for both nonsingular and singular Kalman smoothing systems.

3.4 Nonsingular SSM

Consider the case where the covariance matrices Q_k and R_k in SSM are nonsingular. Prewhitening ϵ_k^p and ϵ_k^m, the objective function of interest is given by

$$\begin{aligned} f(\theta, y) = \sum_{k=1}^{N} \Big\{ & \ell_k^p \left(Q_k^{-1/2} (x_k - G_k(\theta) x_{k-1}) \right) \\ & + \ell_k^m \left(R_k^{-1/2} (z_k - H_k(\theta) x_k) \right) \Big\}, \end{aligned} \tag{3.8}$$

where $y = x = [x_1; \ldots; x_N]$, ℓ_k^p and ℓ_k^m are the loss function corresponding to the distributions of ϵ_k^p and ϵ_k^m. Here we assume ℓ_k^p, ℓ_k^m are smooth; three key examples are least squares, Hybrid, and Student's T losses introduced in Section 3.2. Objective (3.8) can be written compactly as,

$$\begin{aligned} f(\theta, y) = & \ell^p \left(Q^{-1/2} (G(\theta) y - \zeta) \right) + \\ & \ell^m \left(R^{-1/2} (H(\theta) y - z) \right), \end{aligned} \tag{3.9}$$

where $G(\theta)$ and Q are given by

$$\begin{bmatrix} I & 0 & & \\ -G_2(\theta) & I & \ddots & \\ & \ddots & \ddots & 0 \\ & & -G_N(\theta) & I \end{bmatrix}, \qquad \begin{bmatrix} Q_1 & & \\ & \ddots & \\ & & Q_N \end{bmatrix},$$

while $H(\theta)$ and R are given by

$$\begin{bmatrix} H_1(\theta) & 0 & & \\ 0 & H_2(\theta) & \ddots & \\ & \ddots & \ddots & 0 \\ & & 0 & H_N(\theta) \end{bmatrix}, \qquad \begin{bmatrix} R_1 & & \\ & \ddots & \\ & & R_N \end{bmatrix}.$$

Here, $\zeta = [x_0; 0; \ldots; 0]$ and $z = [z_1; \ldots; z_N]$.

The stationary condition in this case is given by

$$\begin{aligned} \mathcal{H}(\theta, y) &= f_y(\theta, y) \\ &= G(\theta)^\top Q^{-\top/2} \ell_r^p(r^p) + H(\theta)^\top R^{-\top/2} \ell_r^m(r^m) \end{aligned} \tag{3.10}$$

where $r^p = Q^{-1/2}(G(\theta)y - \zeta)$ and $r^m = R^{-1/2}(H(\theta)y - z)$.

In the least squares case, (3.10) is a linear equation solved by inverting the block-tridiagonal system

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

for more general smooth penalties ℓ^p, ℓ^m a Newton method is needed to compute $y(\theta)$.

By Theorem 3.3.1, existence and differentiability of $Y(\theta)$ is guaranteed by the existence of the pair $(\overline{\theta}, \overline{y})$ such that, $\mathcal{H}(\overline{\theta}, \overline{y}) = 0$ and the partial Hessian below is invertible:

$$\begin{aligned} \mathcal{H}_y(\overline{\theta}, \overline{y}) =& G(\overline{\theta})^\top Q^{-\top/2} \ell_{rr}^p(r^p) Q^{-1/2} G(\overline{\theta}) + \\ & H(\overline{\theta})^\top R^{-\top/2} \ell_{rr}^m(r^m) R^{-1/2} H(\overline{\theta}). \end{aligned}$$

When ℓ^p, ℓ^m are least squares or Hybrid, $\mathcal{H}_y(\theta, y)$ is invertible for any (θ, y), and for every $\overline{\theta}$ there exist a $\overline{y}$ such that $\mathcal{H}(\overline{\theta}, \overline{y}) = 0$, since both penalties are strictly convex. In the case of the Student's T, the Hessian may fail to be positive definite at some pairs (θ, y) [18] and there is no absolute guarantee that the methodology will hold, as expected for a potentially nonconvex formulation. Practical behavior is another matter, and in numerical experience we see a positive definite Hessian at the minimizer, particularly when a descent method is

used to solve the state space problem (i.e. the objective function is required to decrease at each iteration). We see that the derivative formulas hold.

We now compute remaining terms in Corollary 3.3.2, assuming for simplicity that $G(\theta)$ and $H(\theta)$ are affine functions of θ.

$$\overline{r^p} := Q^{-\top/2}\ell_r^p(r^p), \quad \overline{r^m} := R^{-\top/2}\ell_r^m(r^m)$$

$$\begin{aligned}
f_\theta(\theta,\overline{y}) &= (G\overline{x})_\theta^\top Q^{-\top/2}\ell_r^p(r^p) + (H\overline{x})_\theta^\top R^{-\top/2}\ell_r^m(r^m) \\
f_{\theta\theta}(\theta,\overline{y}) &= (G\overline{x})_\theta^\top Q^{-\top/2}\ell_{rr}^p(r^p)Q^{-1/2}(G\overline{x})_\theta \\
&\quad + (H\overline{x})_\theta^\top R^{-\top/2}\ell_{rr}^m(r^m)R^{-1/2}(H\overline{x})_\theta \\
\mathcal{H}_\theta(\theta,\overline{y}) &= (G(\theta)^\top\overline{r^p})_\theta + G(\theta)^\top Q^{-\top/2}\ell_{rr}^p(r^p)Q^{-1/2}(G\overline{x})_\theta \\
&\quad + (H(\theta)^\top\overline{r^m})_\theta + H(\theta)^\top R^{-\top/2}\ell_{rr}^m(r^m)R^{-1/2}(H\overline{x})_\theta
\end{aligned}$$

We now have, fully and explicitly, first and second derivatives of the value function $v(\theta)$ in (3.7) for the nonsingular case. Though these results are straightforward, they do not appear in any smoothing literature we are aware of in this compact form, even for least squares losses.

3.5 Singular SSM

When covariances Q_k and R_k are singular, we rewrite (3.8) to include null-space constraints. A singular covariance matrix precludes any errors and innovations that are not in its range. We follow [76] in formulating this problem:

$$\begin{aligned}
\min_{\theta,x,r^p,r^m} \quad & \ell^p(r^p) + \ell^m(r^m) \\
\text{s.t.} \quad & Q^{1/2}r^p = G(\theta)x - \zeta, \\
& R^{1/2}r^m = H(\theta)x - z,
\end{aligned} \tag{3.11}$$

and introduce the *Lagrangian*

$$\begin{aligned} f(\theta, y) =& \mathcal{L}(\theta, x, r^p, r^m, \lambda^p, \lambda^m) \\ =& \ell^p(r^p) + \ell^m(r^m) \\ & - \left\langle \lambda^p, Q^{1/2} r^p - G(\theta)x + \zeta \right\rangle \\ & - \left\langle \lambda^m, R^{1/2} r^m - H(\theta)x + z \right\rangle. \end{aligned} \tag{3.12}$$

When Q_k and R_k are invertible, we can solve for r^p, r^m in (3.11) and reduce the problem to (3.8), so nonsingular systems are a special case of (3.11). Formulation (3.11) can be solved for a variety of loss functions ℓ^p and ℓ^m (see [76]).

Here we define the *value function* as a mini-max problem using the Lagrangian:

$$v(\theta) := \max_{\lambda^p, \lambda^m} \min_{x, r^p, r^m} \mathcal{L}(\theta, x, r^p, r^m, \lambda^p, \lambda^m). \tag{3.13}$$

Just as in the non-singular case, the difficulty with the Student's T case is that it is not convex. It therefore becomes difficulty to evaluate $v(\theta)$. However, the development below uses stationarity characterizations, allowing us to use systems of equations that characterize stationarity to proceed, just as in the non-singular case. The system of equations we are interested in is now

$$0 = \mathcal{H}(\theta, y) := f_y(\theta, y), \quad y := \{x, r^p, r^m, \lambda^p, \lambda^m\};$$

that is, the system of equations that defines a saddle point of the Lagrangian. Explicitly, $f_y(\theta, y) = 0$ is given by

$$f_y(\theta, y) = \begin{bmatrix} G(\theta)^\top \lambda^p + H(\theta)^\top \lambda^m \\ \ell_r^p(r^p) - Q^{\top/2} \lambda^p \\ \ell_r^m(r^m) - R^{\top/2} \lambda^m \\ G(\theta)x - \zeta - Q^{1/2} r^p \\ H(\theta)x - z - R^{1/2} r^m \end{bmatrix} = 0.$$

$Y(\theta)$ is differentiable when $\mathcal{H}_y(\theta, y) = f_{yy}(\theta, y)$ is invertible:

$$f_{yy}(\theta, y) = \begin{bmatrix} 0 & 0 & 0 & G(\theta)^\top & H(\theta)^\top \\ 0 & \ell^p_{rr}(r^p) & 0 & -Q^{\top/2} & 0 \\ 0 & 0 & \ell^m_{rr}(r^m) & 0 & -R^{\top/2} \\ G(\theta) & -Q^{1/2} & 0 & 0 & 0 \\ H(\theta) & 0 & -R^{1/2} & 0 & 0 \end{bmatrix} \tag{3.14}$$

We state the following theorem.

Theorem 3.5.1. *$f_{yy}(\theta, y)$ is invertible if and only if $\ell^p_{rr}(r^p), \ell^m_{rr}(r^m)$ are invertible and the so called Hessian of the Lagrangian*

$$\begin{aligned} &H(\theta)G(\theta)^{-1}Q^{1/2}(\ell^p_{rr}(r^p))^{-1}Q^{\top/2}G(\theta)^{-\top}H(\theta)^\top \\ &+ R^{1/2}(\ell^m_{rr}(r^m))^{-1}R^{\top/2}. \end{aligned} \tag{3.15}$$

is invertible. Furthermore (3.15) *is invertible if and only if*

$$\mathcal{N}(R) \cap \mathcal{N}(QG^{-T}H^\top) = \{0\}. \tag{3.16}$$

Proof. We reduce $\mathcal{H}_y(\theta, y)$ in (3.14) to block upper triangular form using invertible block row operations

$$\begin{aligned} \mathcal{R}_1 &= (\ell^p_{rr}(r^p))^{-1}\mathcal{R}_1 \\ \mathcal{R}_2 &= (\ell^m_{rr}(r^m))^{-1}\mathcal{R}_2 \\ \mathcal{R}_3 &= G(\theta)^{-T}\mathcal{R}_3 \\ \mathcal{R}_4 &= -G(\theta)^{-1}\left(\mathcal{R}_4 - Q^{1/2}\mathcal{R}_1 - Q^{1/2}(\ell^p_{rr}(r^p))^{-1}Q^{T/2}\mathcal{R}_3\right) \\ \mathcal{R}_5 &= \mathcal{R}_5 - R^{1/2}\mathcal{R}_2 - H\mathcal{R}_4 \end{aligned}$$

The resulting system is given by

$$\begin{bmatrix} I & 0 & -(\ell^p_{rr})^{-1}Q^{T/2} & 0 & 0 \\ 0 & I & 0 & 0 & -(\ell^m_{rr}(r^m))^{-1}R^{T/2} \\ 0 & 0 & I & 0 & -G^{-1}H^\top \\ 0 & 0 & 0 & I & G^{-1}Q^{1/2}(\ell^p_{rr}(r^p))^{-1}Q^{T/2}G^{-T}H^\top \\ 0 & 0 & 0 & 0 & \big(HG^{-1}Q^{1/2}(\ell^p_{rr}(r^p))^{-1}Q^{T/2}G^{-T}H^\top \\ & & & & \quad +R^{1/2}(\ell^m_{rr}(r^m))^{-1}R^{T/2}\big) \end{bmatrix}$$

The invertibility of $\mathcal{H}_y$ is thus equivalent to the invertibility of the symmetric positive semidefinite system (3.15). □

When Theorem 3.5.1 holds, we use Corollary (3.3.2) to get derivatives of $v(\theta)$ in (3.13):

$$\begin{aligned} v(\overline{\theta}) =& f(\overline{\theta},\overline{y}) \\ v_\theta(\overline{\theta}) =& f_\theta(\overline{\theta},\overline{y}) \\ v_{\theta\theta}(\overline{\theta}) =& f_{\theta\theta}(\overline{\theta},\overline{y}) - \mathcal{H}_\theta(\overline{\theta},\overline{y})^\top \mathcal{H}_y(\overline{\theta},\overline{y})^{-1}\mathcal{H}_\theta(\overline{\theta},\overline{y}). \end{aligned} \tag{3.17}$$

It remains only to compute f_θ, $f_{\theta\theta}$, $\mathcal{H}_\theta$, and $\mathcal{H}_y$.

$$f_\theta(\theta,\overline{y}) := (\langle \overline{\lambda^p}, G(\theta)\overline{x}\rangle)_\theta + (\langle \overline{\lambda^m}, H(\theta)\overline{x}\rangle)_\theta$$

When G, H are affine functions of θ, we have $f_{\theta\theta} = 0$. Finally,

$$\mathcal{H}_\theta(\theta,\overline{y}) = f_{y\theta}(\theta,\overline{y}) = \begin{bmatrix} (G(\theta)^\top\overline{\lambda^p})_\theta + (H(\theta)^\top\overline{\lambda^m})_\theta \\ 0 \\ 0 \\ (G(\theta)\overline{x})_\theta \\ (H(\theta)\overline{x})_\theta \end{bmatrix}.$$

3.5.1 Special case: Invertible R.

The structural unemployment model in the introduction has a singular Q but an invertible R. In such cases, the derivative formulas can be written using only primal quantities, which

significantly decreases the notational burdern. In particular, using the optimality conditions we have

$$\lambda^m = R^{-\top/2}\ell_r^m(r^m)\left(R^{-1/2}(H(\theta)y - z)\right)$$
$$\lambda^p = -G(\theta)^{-\top}H(\theta)^\top\lambda^m$$

and so we get the explicit primal-only formula for $v_\theta(\theta)$ by plugging these expressions into

$$v_\theta(\theta) = (\langle\overline{\lambda^p}, G(\theta)\overline{y}\rangle)_\theta + (\langle\overline{\lambda^m}, H(\theta)\overline{y}\rangle)_\theta.$$

3.5.2 Special case: Least squares.

If $\ell^p(\cdot)$ and $\ell^m(\cdot)$ are both given by $\frac{1}{2}\|\cdot\|^2$, the optimality conditions simplify substantially, and we have

$$r^p = Q^{T/2}\lambda^p, \quad r^m = R^{T/2}\lambda^m$$
$$Q^{1/2}r^p = Q\lambda_p = G(\theta)x - \zeta,$$
$$R^{1/2}r^m = R\lambda_m = H(\theta)x - z$$
$$0 = G(\theta)^\top\lambda^p + H(\theta)^\top\lambda^m.$$

Plugging these conditions back into the Lagrangian, we get the dual objective (i.e., the dual problem is to maximize the objective below with respect to (λ^p, λ^m)).

$$f^*(\lambda^p, \lambda^m) = -\frac{1}{2}(\lambda^p)^\top Q\lambda^p - \frac{1}{2}(\lambda^m)^\top R\lambda^m$$
$$- (\lambda^p)^\top\zeta - (\lambda^m)^\top z$$
$$\text{s.t.} \quad G(\theta)^\top\lambda^p + H(\theta)^\top\lambda^m = 0.$$

Using invertibility of G, we eliminate λ_p and the affine constraint, obtaining an objective function in λ_m alone:

$$f^*(\lambda_m) = -\frac{1}{2}(\lambda^m)^\top\left(HG^{-1}QG^{-T}H^\top + R\right)\lambda^m$$
$$- (\lambda^m)^\top(z - HG^{-1}\zeta).$$

In the least squares case, the dual solution λ^m that maximizes the above objective is unique exactly when the linear system

$$HG^{-1}QG^{-T}H^\top + R \tag{3.18}$$

is invertible, and then we have

$$\lambda_m = \left(HG^{-1}QG^{-T}H^\top + R\right)^{-1}(z - HG^{-1}\zeta),$$

a closed form solution. A simple sufficient condition for the invertibility of (3.18) is to have R itself invertible, as in special case A.

3.6 Numerical Examples

We now apply the results of the previous sections to analyze two simple singular models with unknown states and parameters[1]. In Section 3.6.1 we present a state-space formulation for the AR-1 model, show how to robustify it to outliers in the data, and present explicit derivatives for the value function. We use these derivatives to design an efficient solver for both standard and robust AR models using Newton, Gauss-Newton, and quasi-Newton algorithms. In Section 3.6.2, we apply the methods in this paper to fit a structural model for unemployment rates that can track fast changes. While the structural unemployment model is currently used in the EU, in this paper we only show results on simulated synthetic data where we know ground truth, leaving a data driven illustration to a future collaboration.

3.6.1 Robust AR Fitting

We examine the simplest case where a reformulated state space leads to naturally singularQ_k and R_k. This derivation illustrates how singular covariances can arise naturally in more complex scenarios including all ARMA models. An AR-1 model begins with equations

$$\begin{aligned} x_k &= c + \varphi x_{k-1} + \epsilon_k^p \\ y_k &= H_k x_k + \epsilon_k^m \end{aligned} \tag{3.19}$$

Where $\epsilon_k^p, \epsilon_k^m$ have covariances Q_k, R_k respectivly, c is an unknown constant, and φ is a parameter to be estimated. To make this take the form of (6.1) we create an augmented

[1] github.com/UW-AMO/SSM_ParamsEstimation

state

$$\hat{x}^k = \begin{bmatrix} x_k \\ c \end{bmatrix} \tag{3.20}$$

and use state equations

$$\begin{aligned} \hat{x}^k &= \begin{bmatrix} \varphi & 1 \\ 0 & 1 \end{bmatrix} \hat{x}_{k-1} + \hat{\epsilon}_k^p \\ y_k &= \begin{bmatrix} H_k & 0 \\ 0 & 0 \end{bmatrix} \hat{x}_k + \hat{\epsilon}_k^m \end{aligned} \tag{3.21}$$

Then

$$\hat{Q}_0 = \begin{bmatrix} Q_0 & 0 \\ 0 & 1 \end{bmatrix}, \quad \hat{Q}_k = \begin{bmatrix} Q_k & 0 \\ 0 & 0 \end{bmatrix} \quad k > 0$$

$$\hat{R}_k = \begin{bmatrix} R_k & 0 \\ 0 & 0 \end{bmatrix}$$

This choice of $\hat{Q}_0$ will allow us to fit the constant c as part of the state while $\hat{Q}_k, k > 0$ will act as equality constraints holding it constant through all time points. In order to compute the derivatives of the value functions we first note the following derivative formula in this case for any vector η.

$$\Big(G_i(\theta)\eta\Big)_\theta = \Big(\begin{bmatrix} \varphi & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \eta^1 \\ \eta^2 \end{bmatrix} \Big)_\theta = \begin{bmatrix} \eta^1 \\ 0 \end{bmatrix} = \tilde{D}\eta \tag{3.22}$$

Where $\tilde{D} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$. Define

$$D = \begin{bmatrix} 0 & 0 & & \\ \tilde{D} & 0 & \ddots & \\ & \ddots & \ddots & 0 \\ & & \tilde{D} & 0 \end{bmatrix}$$

Then using the above, for the AR-1 model

$$\Big(G(\theta)x\Big)_\theta = -Dx$$

And similarly

$$\Big(G(\theta)^T\lambda\Big)_\theta = -D^\top\lambda$$

We now have the expressions

$$f_\theta(\theta,\overline{y}) := -\left\langle\overline{\lambda^p}, D\overline{x}\right\rangle$$

$$f_{\theta\theta} = 0$$

$$\mathcal{H}_\theta(\theta,\overline{y}) = -\begin{bmatrix} D^\top\overline{\lambda^p} \\ 0 \\ 0 \\ D\overline{x} \\ 0 \end{bmatrix}$$

When combined with the general results of section 3.5, we get

$$\begin{aligned} v(\overline{\theta}) =& f(\overline{\theta},\overline{y}) \\ v_\theta(\overline{\theta}) =& f_\theta(\overline{\theta},\overline{y}) = -\left\langle\overline{\lambda^p}, D\overline{x}\right\rangle \\ v_{\theta\theta}(\overline{\theta}) =& -\mathcal{H}_\theta(\overline{\theta},\overline{y})^\top\mathcal{H}_y(\overline{\theta},\overline{y})^{-1}\mathcal{H}_\theta(\overline{\theta},\overline{y}). \end{aligned} \tag{3.23}$$

3.6.2 Fast Tracking of Unemployment Rates

In this section, we apply the proposed approach to estimate parameters (ℓ_1,ℓ_2,γ) for the structural unemployment model (3.2)—(3.4). To test the approach, we generate ground truth parameters and then create synthetic data in order to compare model performance and speed using different formulations and algorithms. The data is generated by fixing parameters to reasonable values similar to those observed in practice, namely at $\begin{bmatrix} l_1 & l_2 & \gamma \end{bmatrix} = \begin{bmatrix} 0.68 & 1.41 & -0.68 \end{bmatrix}$, and applying the unemployment rate state space model (3.2)—(3.4) to generate the state as well as noisy observations. We then consider three cases: nominal errors, outliers in the observations, and jumps in the unemployment process. In the nominal cases we use variance parameters known to the smoother. To generate outliers we randomly select 10% of measurements and add additional noise drawn from a $\mathcal{N}(0,1)$ Gaussian distribution.

		Nominal			Outliers			Large Jumps		
		ls/ls	T/ls	H/ls	ls/ls	ls/T	ls/H	ls/ls	T/ls	H/ls
Newton	$\|\|\hat{\theta}-\theta\|\|^2$	0.158	0.143	0.1884	1.72	0.115	0.142	0.296	0.032	0.099
	Inner Iter	19	250	206	14	71	102	21	329	373
	Outer Iter	9	5	10	6	15	13	13	9	14
	Time	11.5	36.5	29.7	6.3	20.2	19.3	13.3	48.1	49.5
L-BFGS	$\|\|\hat{\theta}-\theta\|\|^2$	0.158	0.137	0.184	1.72	0.115	0.141	0.296	0.034	0.099
	Inner Iter	19	228	207	19	86	88	32	375	519
	Outer Iter	10	18	10	9	17	12	16	14	15
	Time	6.2	33.9	26.5	6.3	19.4	14.0	10.6	51.4	67.5
LM-Newton	$\|\|\hat{\theta}-\theta\|\|^2$	0.158	0.069^2	0.184	1.04	0.12	0.142	0.296	0.028	0.099
	Inner Iter	16	385	180	32	70	114	24	302	357
	Outer Iter	12	15	20	25	28	19	18	15	21
	Time	7.2	51.6	26.4	14.4	20.0	21.0	11.3	43.2	48.1

Table 3.1: Table of results when run on generated data. The second row indicates the loss functions that were used where ls stands for least squares ($\frac{1}{2}||\cdot||^2$), T stands for Students T with $\nu = 10$, and H stands for Hybrid with $\epsilon = .7$.

To generate large jumps we add large deviations $.4, -.2$ at indices corresponding to 25 and 65. Two Examples of the generated data are in Figure 3.2. An example of the estimated using this data are in Figure 3.3.

All algorithms are initialized at $\begin{bmatrix}0 & 0 & 0\end{bmatrix}$, except for LM-Newton on T/ls in the nominal case, which is initialized at $\begin{bmatrix}0 & 0 & 0.5\end{bmatrix}$, as the standard zero initialization leads to bad results for this (nonconvex) case.

In the first iteration the state is initialized by propagating the initial x_0 through the dynamics for all time. In subsequent iterations the state is always initialized using the previous state solution. In all methods, the full state at each iteration is computed using

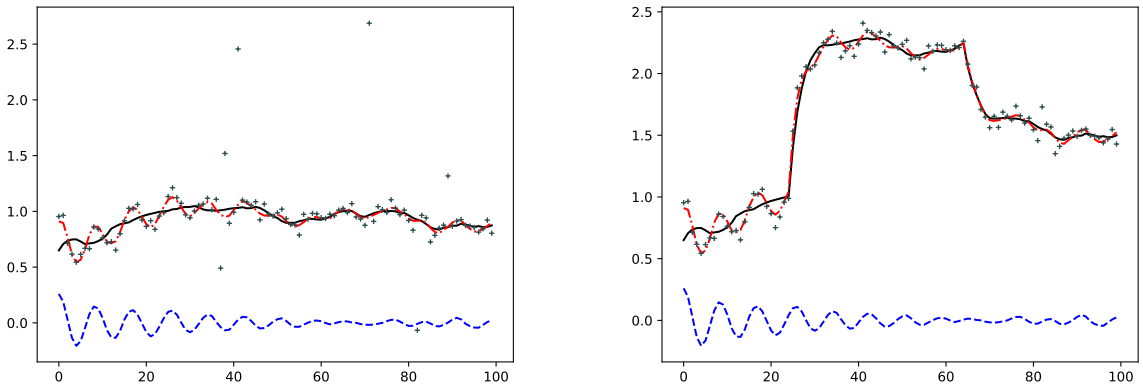


Figure 3.2: Example of generated data with outliers (left) and large jumps (right). Observations are shown in gray.

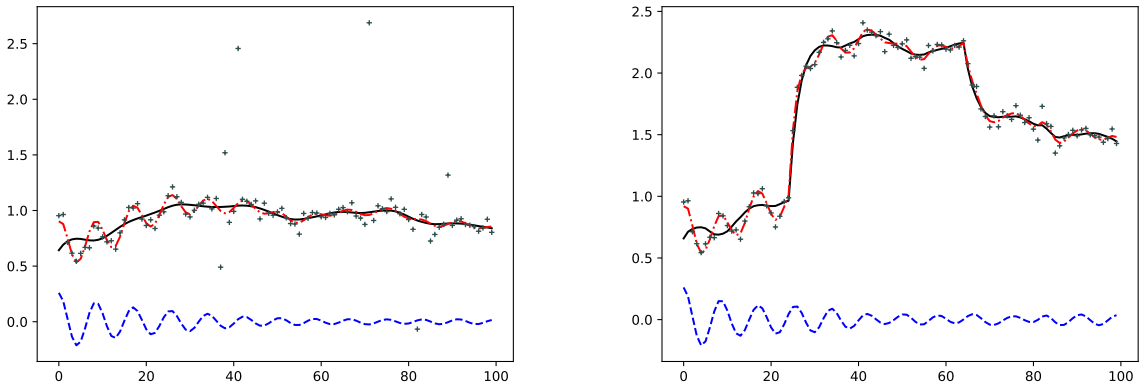


Figure 3.3: Example of estimated solution run on data in Figure 3.2. Left is ls/H run on data with outliers. Right panel shows T/ls run on data with large jumps added. Both are computed using Newton's method.

Newton's method to find a saddle point of the augmented Lagrangian

$$\begin{aligned} AL(\theta, y) =& \ell^p(r^p) + \ell^m(r^m) \\ & - \left\langle \lambda^p, Q^{1/2} r^p - G(\theta)x + \zeta \right\rangle \\ & - \left\langle \lambda^m, R^{1/2} r^m - H(\theta)x + z \right\rangle \\ & + \frac{1}{2} \| Q^{1/2} r^p - G(\theta)x + \zeta \|^2 \end{aligned} \tag{3.24}$$

using the Hessian of the Lagrangian:

$$AL_{yy}(\theta, y) = \begin{bmatrix} G(\theta)^T G(\theta) + H(\theta)^T H(\theta) & -G(\theta)^T Q^{1/2} & -H(\theta)^T R^{1/2} & G(\theta)^T & H(\theta)^T \\ -Q^{T/2} G(\theta) & \ell^p_{rr}(r^p) + Q^{T/2} Q^{1/2} & 0 & -Q^{T/2} & 0 \\ -R^{T/2} H(\theta) & 0 & \ell^m_{rr}(r^m) + R^{T/2} R^{1/2} & 0 & -R^{T/2} \\ G(\theta) & -Q^{1/2} & 0 & 0 & 0 \\ H(\theta) & 0 & -R^{1/2} & 0 & 0 \end{bmatrix} \tag{3.25}$$

The value at an optimal point of (3.24) is the same as at an optimal point of (3.12). (3.24) is better conditioned which leads to faster convergence in practice. For the outer iterations on the parameter space we compare a Newton method, L-BFGS, and a LM-Newton solver. The standard Newton and L-BFGS are from a standard python library. The LM-Newton solver is a quasi-Newton method where the Hessian is boosted by a parameter that is updated adaptively based on model performance. The results are summarized in Table 3.1.

All methods work well for convex models. In the nonconvex case, the algorithms become more sensitive. In particular when ℓ^p is Student's T, we have to boost ℓ^p_{rr} by a constant in order to make Newtons method converge. In practice this constant must be tuned depending on the parameter ν in the Student's T function. Convergence is therefore sensitive to the choice of ν and boosting constant but a good rule of thumb is to choose $1 \leq \nu \leq 20$ and boost just enough to make the Hessian positive semidefinite.

3.7 Discussion

We presented a general approach for parameter estimation in singular and non-singular Kalman smoothing models. In particular we showed how to compute first and second derivatives of the value function (optimizing over state) with respect to the hidden parameters for both singular and nonsingular cases, which captures a wide variety of models, including the motivating example. A simple numerical illustration shows how the computed quantities can be used by a variety of optimization methods. The examples also show that when working with structural parameters, it pays off to have convex subproblems within each iteration of the value function. While non-convex losses such as Student's T are always appealing from

a modeling perspective, when the problem is to find both the state and parameters, the resulting models are more fragile than those that use convex losses. This observation opens the way to future research in both theory and algorithm design.

Finally, MAP estimates are only one approach to estimate parameters. Multiple work addresses bias in MAP estimates [74, 26]. These approaches can benefit from the techniques developed here, since they require solving auxiliary MAP-like problems. For marginal likelihood, these problems arise when the Laplace approximation is used [138]. How to best use the methodology developed here in these broader contexts is left open to future work.

Acknowledgment. We are very grateful to Jon Nielsen (Economic Council of the Labour Movement (ECLM)) for pointing us to the use of Kalman smoothers in structural unemployment models, and for teaching us about these models.

Chapter 4

SECOND ORDER CONVEX-COMPOSITE PLQ OPTIMIZATION AND KALMAN SMOOTHING

In this Chapter we develop an interior point approach for convex-composite PLQ models. This work is motivated by Kalman smoothing models with nonlinear process and measurement models.

4.1 Introduction

We consider the full nonlinear state-space model:

$$\begin{aligned} x_1 &= x_0 + w_0 \\ x_k &= g_k(x_{k-1}) + w_k, \quad k = 2, \ldots, N \\ y_k &= h_k(x_k) + v_k, \quad k = 1, \ldots, N \end{aligned} \tag{4.1}$$

where $x_k \in \mathbb{R}^n$ are unknown latent states to be estimated, $g_k : \mathbb{R}^n \to \mathbb{R}^n$ is a known process model, $h_k : \mathbb{R}^n \to \mathbb{R}^m$ is a known measurement model, $y_k \in \mathbb{R}^m$ are known observations, and w_k, v_k are mutually independent mean zero random variables with known covariance matrices Q_k and R_k respectively.

The classic Kalman smoother assumes that g_k and h_k are linear maps and that the error terms w_k, v_k have Gaussian distributions [82]. In this case a maximum likelihood estimate can be obtained using the classic RTS algorithm [124]. The likelihood equation in this case is given by:

$$\min_{\{x_k\}} \sum_{k=1}^{N} ||Q_k^{-1/2}(G_k x_{k-1} - x_k)||^2 + ||R_k^{-1/2}(H_k x_k - y_k)||^2 \tag{4.2}$$

where G_k and H_k are matrices corresponding to the linear maps g_k, h_k. When g_k and h_k are

allowed to be any smooth map the likelihood equation becomes

$$\min_{\{x_k\}} \sum_{k=1}^{N} ||Q_k^{-1/2}(g_k(x_{k-1}) - x_k)||^2 + ||R_k^{-1/2}(h_k(x_k) - y_k)||^2 \tag{4.3}$$

for which techniques from convex-composite optimization must be used. In this case with nonlinear constraints added [29] suggest using a Gauss-Newton method of iteravely linearizing g_k, h_k to obtain local models which are then solved via an interior point scheme. As the Gaussian assumptions the least squares terms in (4.3) can be replaced by other loss functions corresponding to other distributions. The uses of ℓ_1, Huber, and Student's T have been explored [13, 18] using techniques similar to that described above. The main difficulty is that depending on the choice of loss function slight adjustments must be made, either a clever reformulation of the local subproblem to write it as a Quadratic Program (QP), or a careful choice of Hessian approximation.

In the case of linear g_k, h_k a general framework has been developed in [15] to allow for any convex piecewise linear quadratic (PLQ) loss function [126]. This involves solving the broad class of problems of the form

$$\min_x \rho_1(Gx - \eta) + \rho_2(Hx - y). \tag{4.4}$$

This approach takes advantage of the dual form of PLQ functions to implement an interior point scheme with a log-barrier term. With this as motivation we seek to find an efficient method to generalize this approach to that of nonlinear g_k, h_k.

A general second-order method for convex-composite PLQ functions has been developed by authors of [41], using a Newton-like method on generalized equations. Here for a given problem of the form

$$\min_z \rho(h(z))$$

first-order optimality conditions for the primal dual pair (z, ζ) are written as $0 \in g(z, \zeta) + G(z, \zeta)$, where $g(z, \zeta)$ is a $\mathcal{C}^1$-smooth function, and $G(z, \zeta)$ is a set-valued mapping. Emulating Newton's method, the functional piece of this is then linearlized and a Newton-like

method is obtained by the following iteration scheme:

$$\text{Find } (z^{k+1}, \zeta^{k+1}) \text{ so that } 0 \in g(z^k, \zeta^k) + \nabla g(z^k, \zeta^k) \begin{pmatrix} z^{k+1} - z^k \\ \zeta^{k+1} - \zeta^k \end{pmatrix} + G(z^{k+1}, \zeta^{k+1}). \quad (4.5)$$

The subproblem (4.5) is equivalent to finding a primal-dual pair that satisfies first-order optimality conditions of a related structured nonconvex problem. We modify this optimization problem with an implicit trust-region method using an iteratively updated penalty parameter. The resulting subproblem is solved using an interior point scheme that uses the dual form of PLQ functions designed to ensure that the first-order optimality conditions are satisfied. This gives an efficient second-order algorithm that solves the nonlinear Kalman problem for any PLQ loss functions.

In Section 4.2 we review necessary information about PLQ functions and convex interior point methods. In Section 4.3 the general theory of Newton's method for generalized equations is reviewed and an adaptive parameter scheme is introduced. This method leads to a structured but nonconvex subproblem that must be solved in each iteration. Section 4.4 focuses on an interior point method that can be used to solve these subproblems. In Section 4.5 we explicitly reformulate the nonlinear Kalman smoothing problem in a way to apply the method of the previous section and in Section 4.6 we give a simple numerical example that illustrates the proposed method.

4.2 PLQ Functions

Piecewise linear quadratic or PLQ functions [126] form a broad class of functions with strong properties for optimization. For our purposes we will take the following definition:

Definition 4.2.1. *A convex function $\rho(z) : \mathbb{R}^n \to \bar{\mathbb{R}}$ is a PLQ function if there is a symmetric psd matrix M, any matricies C, B and vectors $c \geq 0$ and b so that for any $z \in$ dom ρ*

$$\rho(z) = \sup_{Cu \leq c} \{\langle Bz + b, u\rangle - \frac{1}{2}\langle u, Mu\rangle\}.$$

Example 4.2.2. *PLQ loss functions are widely used in the literature across many applications. We quickly review some common scalar examples and how to write them in their dual form. Plots of all examples discussed are shown in Figure 6.2.*

- *least squares (quadratic): Take* $C = 0, c = 1, M = 1, b = 0, B = I$. *Then*

$$||z||^2 = \sup_{u \in \mathbb{R}} \{uz - \frac{1}{2}u^2\}$$

- *1-norm: Take* $C = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, c = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, M = 0, b = 0, B = I$. *Then*

$$|z|_1 = \sup_{u \in [-1,1]} \{uy\}$$

- *quantile: Take* $C = \begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{bmatrix}, c = \begin{bmatrix} 0 \\ \tau \\ \tau \\ 0 \end{bmatrix}, M = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, B = \begin{bmatrix} I \\ I \end{bmatrix}$. *Then*

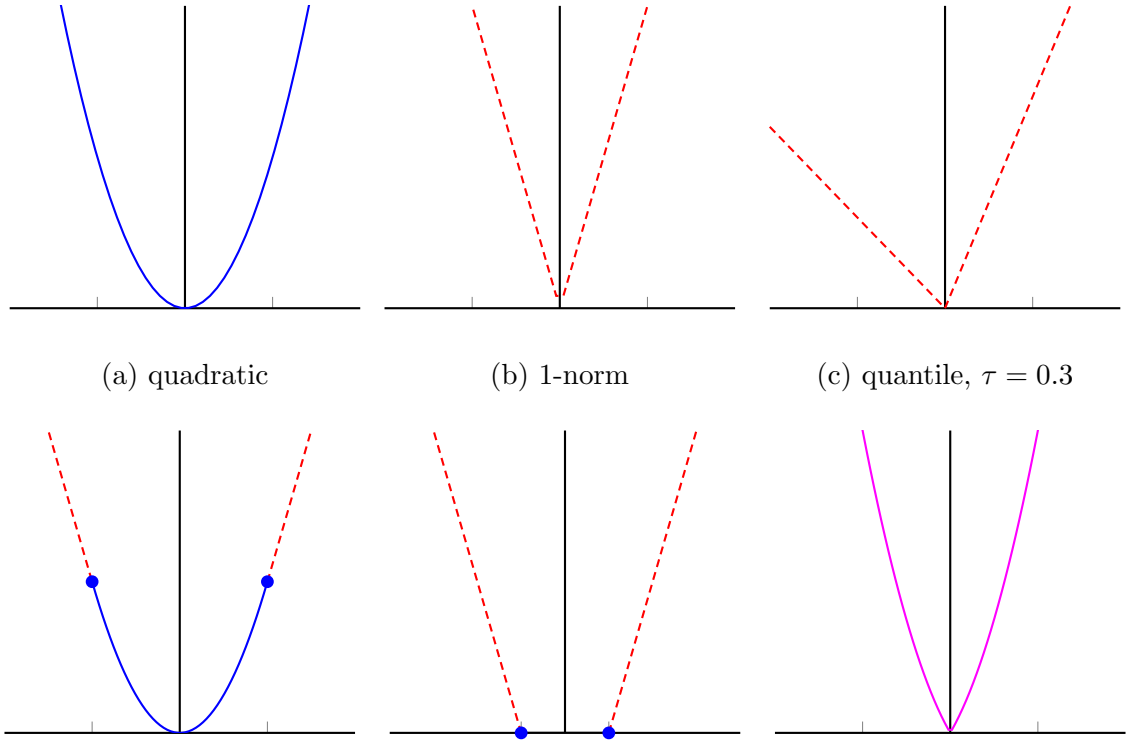
$$(-\tau z)_+ + ((1-\tau)z)_+ = \sup_{(u_1,u_2) \in [0,\tau] \times [-\tau, 0]} \{\langle u_1, z \rangle + \langle u_2, z \rangle\}$$

- *Huber: Take* $C = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, c = \begin{bmatrix} \kappa \\ \kappa \end{bmatrix}, M = 1, b = 0, B = I$. *Then*

$$\begin{cases} \kappa z - \frac{1}{2}\kappa^2, & \text{if } |z| \geq \kappa \\ \frac{1}{2}z^2, & \text{otherwise} \end{cases} = \sup_{u \in [-\kappa, \kappa]} \{\langle u, z \rangle - \frac{1}{2}u^2\}$$

- *Vapnik: Take* $C = \begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{bmatrix}, c = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, M = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, b = \begin{bmatrix} \epsilon \\ \epsilon \end{bmatrix}, B = \begin{bmatrix} I \\ -I \end{bmatrix}$. *Then*

$$(z-\epsilon)_+ + (-z-\epsilon)_+ = \sup_{(u_1,u_2) \in [0,1] \times [0,1]} \{\langle u_1, z - \epsilon \rangle + \langle u_2, -z - \epsilon \rangle\}$$



(a) quadratic (b) 1-norm (c) quantile, $\tau = 0.3$

(d) Huber, $\kappa = 1$ (e) Vapnik, $\epsilon = 0.5$ (f) elastic net

Figure 4.1: Common piecewise linear-quadratic (PLQ) losses. Quadratic parts are shown in solid blue. Linear parts are dashed red. Pink is sum of linear and quadratic.

In later sections optimization problems that are sums of PLQs will be considered. In fact this is equivalent to optimizing just one PLQ function due to the following remark.

Remark 4.2.3. *Suppose ρ_1, ρ_2 are PLQ with*

$$\rho_1(z) = \sup_{C_1 u \leq c_1} \{\langle u, B_1 z\rangle - \frac{1}{2}\langle u, M_1 u\rangle\}, \quad \rho_2(z) = \sup_{C_2 u \leq c_2} \{\langle u, B_2 z\rangle - \frac{1}{2}\langle u, M_2 u\rangle\}$$

Then $\rho = \rho_1 + \rho_2$ is also PLQ with

$$\rho(z) = \sup_{Cu \leq c} \{\langle u, Bz\rangle - \frac{1}{2}\langle u, Mu\rangle\}$$

where $C = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix}, c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}, M = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix}$, *and* $B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$.

Proof. This is clear to see by writing $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$. □

This dual form of PLQ functions is very useful because even in the nonsmooth case, the dual allows an explicit form of first-order KKT conditions.

4.3 Second order methods for convex-composite PLQ functions

We consider problems of the type:

$$\min_z \rho_1(Az - a) + \rho_2(F(z) - b) \tag{4.6}$$

where ρ_1, ρ_2 are convex PLQ functions, A is any matrix, and $F(z)$ is a smooth function. We will take the same perspective as [41] and consider a linearization of the generalized equation arising from optimality conditions. Explicitly, for the equivalent problem

$$\min_{z,y} \rho_1(Az - a) + \rho_2(y)$$
$$\text{s.t. } y = F(z) - b$$

consider the Lagrangian function obtained by dualizing the quality constraint:

$$L(z, y, \zeta) = \rho_1(Az - a) + \rho_2(y) + \zeta^T(F(z) - b - y).$$

From this we can derive KKT conditions for (4.6):

$$0 \in \begin{pmatrix} \nabla F(z)^T \zeta \\ F(z) - b \end{pmatrix} + \begin{pmatrix} A^T \partial \rho_1(Az - a) \\ \partial \rho_2^*(\zeta) \end{pmatrix} \tag{4.7}$$

where we have used that $\zeta \in \partial \rho_2(y) \iff y \in \partial \rho_2^*(\zeta)$. The same optimality conditions can be obtained by directly using the PLQ structure using the conjugate of ρ_2:

$$\rho_1(Az - a) + \rho_2(F(z) - b) = \sup_\zeta \rho_1(Ax - a) + \langle \zeta, F(z) - b \rangle - \rho_2^*(\zeta)$$

The convex-composite Lagrangian

$$\rho_1(Ax-a)+\langle\zeta,F(z)-b\rangle-\rho_2^*(\zeta)$$

appears through this process, and immediately gives rise to (4.7).

Next, for a fixed (z^k,ζ^k) we consider solving the following linearization of (4.7)

$$0\in\begin{pmatrix}\nabla F(z^k)^T\zeta\\ F(z^k)-b\end{pmatrix}+\begin{pmatrix}\sum_i\zeta_i^k\nabla^2F_i(z^k) & \nabla F(z^k)^T\\ \nabla F(z^k) & 0\end{pmatrix}\begin{pmatrix}z-z^k\\ \zeta-\zeta^k\end{pmatrix}+\begin{pmatrix}A^T\partial\rho_1(Az-a)\\ \partial\rho_2^*(\zeta)\end{pmatrix}. \quad (4.8)$$

In order to solve this linearized problem we employ a lemma

Lemma 4.3.1 ([41] lemma 3.2)**.** *The pair (d^{k+1},ζ^{k+1}) satisfy the optimality conditions for*

$$\min_d\rho_1(Az^k-a+Ad)+\rho_2(F(z^k)+\nabla F(z^k)d-b)+\frac{1}{2}d^T\sum_i\zeta_i^k\nabla^2F_i(z^k)d$$

if and only if (z^k+d^{k+1},z^{k+1}) solves (4.8)*.*

This establishes a way to implement a second order Newton like method on (4.6). However, like other Newton methods, this is not necessarily globally convergent. We therefore consider the quasi-Newton method that arises when we add an implicit trust-region to each subproblem. That is we consider solving the linearization

$$0\in\begin{pmatrix}\nabla F(z^k)^T\zeta\\ F(z^k)-b)\end{pmatrix}+\begin{pmatrix}\sum_i\zeta_i^k\nabla^2F_i(z^k)+\lambda I & \nabla F(z^k)^T\\ \nabla F(z^k) & 0\end{pmatrix}\begin{pmatrix}z-z^k\\ \zeta-\zeta^k\end{pmatrix}+\begin{pmatrix}A^T\partial\rho_1(Az-a)\\ \partial\rho_2^*(\zeta)\end{pmatrix} \quad (4.9)$$

which corresponds to the following optimization problem

$$\min_d\rho_1(Az^k-a+Ad)+\rho_2(F(z^k)+\nabla F(z^k)d-b)+\frac{1}{2}d^T\left(\sum_i\zeta_i^k\nabla^2F_i(z^k)+\lambda I\right)d, \quad (4.10)$$

where λ is used to implicitly control the size of the step d. Furthermore we will adaptivly update λ via a scheme motivated by [100]. Define

$$\text{pred}=\rho_1(Az^k-a)+\rho_2(F(z^k)-b)-\rho_1(A(z^k+d^{k+1})-a)-\rho_2\left(F(z^k)+\nabla F(z^k)d^{k+1}-b\right)$$

$$\text{actual} = \rho_1(Az^k - a) + \rho_2(F(z^k) - b) - \rho_1(A(z^k + d^{k+1}) - \rho_2(F(z^k + d^{k+1}) - b)$$

and consider the ratio, often referred to as the *gain ratio*, φ =actual/pred. If $\varphi < 0$ then we set $\lambda^k = 10\lambda^k$ and repeat the iteration. If $\varphi > 0$ then we set $\lambda^{k+1} = \max\{\frac{1}{3}, 1-(2\varphi-1)^3\}\lambda^k$. A plot of this function is shown in Figure 4.2. The purpose of this is that when λ is large, the local problem (4.10) will approximate a Gauss-Newton type subproblem which will have better performance when the iterate z^k is far from the optimal point. Similarly when λ is small, the local problem (4.10) will approximate a Newton step which performs better when the iterate z^k is close to the solution. This update scheme for λ gives a way to blend these two local models using on the gain ratio as a proxy for distance to the optimal point. This establishes the overall method used to solve (4.6). We now turn our attention to solving (4.10).

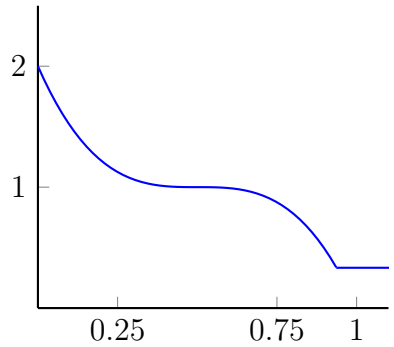


Figure 4.2: A plot of the update function for λ.

4.4 Non-Convex Interior Point

We now consider the class of problems of the form

$$\begin{aligned}\min_d \quad f(d) &= \rho_1(Ad + (Az^k - a)) + \rho_2(\nabla F(z^k)d + (F(z^k) - b)) + \frac{1}{2}d^T\left(H + \lambda I\right)d \\ &= \rho\left(\begin{bmatrix} A \\ \nabla F(z^k) \end{bmatrix} d + \begin{bmatrix} Az - a \\ F(z^k) - b \end{bmatrix}\right) + \frac{1}{2}d^T\left(H + \lambda I\right)d\end{aligned} \tag{4.11}$$

where $H = \sum_i \zeta_i^k \nabla^2 F_i(z^k)$.

Remark 4.4.1. *The linearization of F could lead to a feasibility problem for ρ_2. A full consideration of this is in [41] but we do note that sujectivity of $\nabla F(z^k)$ guarantees that* (4.11) *can be solved.*

In general (4.11) need not be convex as no assumptions are made on $\nabla^2 F_i$ (other than that they exist). In order to solve problems of this type we use an interior point method and exploit the structure of the PLQ penalties by writing them in their dual form. The details of an interior point scheme in a similar but convex setting are considered in [15]. As lemma 4.3.1 requires only that first-order optimality conditions are satisfied, *not* full optimality, the analysis is almost identical. We will instead focus only on the required implementation in this setting. From (4.11) we begin by dualizing:

$$\min_d \sup_{Cu \leq c} \left\langle \begin{bmatrix} A \\ \nabla F(z^k) \end{bmatrix} d + \begin{bmatrix} Az^k - a \\ F(z^k) - b \end{bmatrix}, u \right\rangle - \frac{1}{2} u^T M u + \frac{1}{2} d^T (H + \lambda I) d.$$

Using a log-barrier directly on the hyper-sparse constraints $Cu \leq c$ gives us

$$\min_d \sup_u \left\{ \mathcal{L}(d,u) := \left\langle \begin{bmatrix} A \\ \nabla F(z^k) \end{bmatrix} d + \begin{bmatrix} Az^k - a \\ F(z^k) - b \end{bmatrix}, u \right\rangle \right.$$

$$\left. -\frac{1}{2} u^T M u + \mu 1^T \log(c - Cu) + \frac{1}{2} d^T (H + \lambda I) d \right\}.$$

We introduce a dual variable $q = \mu 1/(c - Cu)$, where the division is coordinate-wise. With this substitution, we obtain the KKT system:

$$0 = F_\mu(q,u,d) := \begin{bmatrix} \operatorname{diag}(c - Cu)q - \mu 1 \\ \begin{bmatrix} A \\ \nabla F(z^k) \end{bmatrix} d + \begin{bmatrix} Az - a \\ F(z^k) - b \end{bmatrix} - Mu - C^T q \\ \begin{bmatrix} A^T & \nabla F(z^k)^T \end{bmatrix} u + (H + \lambda I) d \end{bmatrix}.$$

The IP iterations will use Newton's method on the system

$$\begin{bmatrix} D & -QC & 0 \\ -C^T & -M & \begin{bmatrix} A \\ \nabla F(z^k) \end{bmatrix} \\ 0 & \begin{bmatrix} A^T & \nabla F(z^k)^T \end{bmatrix} & H + \lambda I \end{bmatrix} \begin{bmatrix} \Delta q \\ \Delta u \\ \Delta z \end{bmatrix} = -F_\mu.$$

To solve this we apply the row operation

$$R_2 = R_2 + C^T D^{-1} R_1$$

to obtain

$$\begin{bmatrix} D & -QC & 0 \\ 0 & -T & \begin{bmatrix} A \\ \nabla F(z^k) \end{bmatrix} \\ 0 & \begin{bmatrix} A^T & \nabla F(z^k)^T \end{bmatrix} & H + \lambda I \end{bmatrix}$$

where $T = M + C^T D^{-1} QC$ has been carefully studied [15]; it is invertible when the combined PLQ is well-defined. Next we apply the operation

$$R_3 = R_3 + \begin{bmatrix} A^T & \nabla F(z^k)^T \end{bmatrix} T^{-1} R_2$$

to obtain

$$\begin{bmatrix} D & -QC & 0 \\ 0 & -T & \begin{bmatrix} A \\ \nabla F(z^k) \end{bmatrix} \\ 0 & 0 & H + \lambda I + \begin{bmatrix} A^T & \nabla F(z^k)^T \end{bmatrix} T^{-1} \begin{bmatrix} A \\ \nabla F(z^k) \end{bmatrix} \end{bmatrix}$$

The same row operations can be applied to $-F_\mu$ to obtain the modified right hand sides. All the work is now pushed forward into solving a linear system with the following matrix:

$$H + \lambda I + \begin{bmatrix} A^T & \nabla F(z^k)^T \end{bmatrix} T^{-1} \begin{bmatrix} A \\ \nabla F(z^k) \end{bmatrix}$$

$$= H + \lambda I + A^T T_1^{-1} A + \nabla F(z^k)^T T_2^{-1} \nabla F(z^k).$$

The parameter μ is to be modified according to standard rules, e.g. average value of the vector Dq divided by 10 [15]. We have the following corollary immediatly from work done in the convex case.

Corollary 4.4.2 ([15], Theorem 14)**.** *For large enough λ, interior point methods applied to* (4.11) *will converge to a first-order stationary point.*

4.5 Nonlinear Kalman Smoothing

Here we derive how extending the Kalman smoothing idea to a non-linear state space model can give rise to a equivalent problem of minimizing a PLQ over non-convex constraints. We consider the problem of inferring an unobserved state sequence $x_1, \ldots, x_N$ from noisy observations $y_1, \ldots, y_N$ using the model:

$$\begin{aligned} x_1 &= x_0 + w_1 \\ x_k &= g_k(x_{k-1}) + w_k, \quad k = 2, \ldots, N \\ y_k &= h_k(x_k) + v_k, \quad k = 1, \ldots, N \end{aligned} \tag{4.12}$$

where x_0 is a given initial state estimate, g_k, h_k are given smooth maps that govern the process and measurement models, and $y_1, \ldots, y_N$ are observations. We also assume that w_k and v_k are mutually independent random variables with known possibly singular covarianace matrices Q_k and R_k and that they both follow from log-concave distributions. To obtain a compact form of the MAP estimate for $x_1, \ldots, x_N$ given $y_1, \ldots, y_N$ we proceed as in chapter 2. Define new variables u_k, t_k by

$$\begin{aligned} Q_k^{1/2} u_k &= g_k(x_k) - x_{k-1}, \quad k = 1, \ldots, N \\ R_k^{1/2} t_k &= h_k(x_k) - y_k, \quad k = 1, \ldots, N \end{aligned}$$

and define

$$z = \begin{bmatrix} u_1^T & t_1^T & x_1^T & \cdots & u_N^T & t_N^T & x_N^T \end{bmatrix}^T.$$

Let $F : \mathbb{R}^{(2n+m)N} \to \mathbb{R}^{(n+m)N}$ be a smooth map given by:

$$F(z) = \begin{pmatrix} Q_1^{1/2}u_1 - x_1 + x_0 \\ R_1^{1/2}t_1 + h_1(x_1) - y_1 \\ Q_2^{1/2}u_2 - x_2 + g_2(x_1) \\ \vdots \\ R_N^{1/2}t_N + h_N(x_N) - y_N \end{pmatrix}. \tag{4.13}$$

Note that

$$\nabla F(z) = \begin{bmatrix} D_1 & 0 & \cdots & 0 \\ B_1 & D_2 & 0 & \vdots \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & B_{N-1} & D_N \end{bmatrix}$$

where

$$D_i = \begin{bmatrix} Q_i^{1/2} & 0 & -I \\ 0 & R_i^{1/2} & \nabla h_i(x_i) \end{bmatrix}, \quad B_j = \begin{bmatrix} 0 & 0 & \nabla g_{j+1}(x_j) \\ 0 & 0 & 0 \end{bmatrix}$$

and so ∇F maintains the same structure that was crucial in chapter 2. A maximum a-posteori (MAP) estimate for the states can be formulated as

$$\begin{aligned} &\min_z \rho(z) \\ &\text{s.t. } F(z) = 0 \end{aligned} \tag{4.14}$$

where $\rho(z) = \sum_{k=1}^{N} \rho_p(u_k) + \rho_m(t_k)$ and ρ_p, ρ_p are convex penalties corresponding to process and measurement noise respectively.

Remark 4.5.1. *Note that this fits into the framework of* (4.6) *by taking* $\rho_2 = \delta_{\{0\}}$*, the indicator function for* 0*. This can introduce feasibility problems particularly when solving* (4.11)*. As noted before, surjectivity of* ∇F *alleviates this problem so we state the following theorem from Chapter 2 for the* F *given above.*

Theorem 4.5.2 (Theorem 2.3.1)**.** *Let* F *be defined as in* (4.13)*. The following are equivalent.*

1. $\nabla F(z)$ *is surjective.*

2. *Each block D_i is surjective.*

3. $\text{Null}\left(\begin{bmatrix} Q_i^{1/2} & 0 \\ 0 & R_i^{1/2} \end{bmatrix}\right) \subset \text{Ran}\left(\begin{bmatrix} I \\ \nabla h_i(x_i) \end{bmatrix}\right)$ *for all i.*

4. $R_i + \nabla h_i(x_i)\left(I - (Q_i + I)^{-1}\right)\nabla h_i(x_i)^T$ *is invertible for all i.*

4.6 Numerical Example

In this section we consider the problem of estimate an unknown function in two variables from a set of noisy range measurements from two fixed points with outliers present[1]. To simulate this data we take

$$\begin{bmatrix} X(t) \\ \dot{X}(t) \\ Y(t) \\ \dot{Y}(t) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ e^{t/10} \\ \frac{1}{10}e^{t/10} \end{bmatrix}.$$

The model for the states is then given by $x_k = G_k x_{k-1} + w_k$, where w_k has covariance Q_k and

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

The inital state is given by $x_0 = \begin{bmatrix} 1 & 0 & 1 & .1 \end{bmatrix}^T$ with $Q_0 = I$. The measurement model consists of squared distance measurements from two fixed points, $(0, 0)$ and $(10, 0)$. Thus the nonlinear measurement maps $h_k : \mathbb{R}^4 \to \mathbb{R}^2$ are given by:

$$h_k(x_k) = \begin{bmatrix} x_{1,k}^2 + x_{3,k}^2 \\ (x_{1,k} - 10)^2 + x_{3,k}^2 \end{bmatrix}.$$

---

[1] github.com/UW-AMO/NC-PLQ

The measurement covariances are given by

$$R_k = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}.$$

For the experiment we consider the presence of outliers and so when the measurements are simulated we add noise to %10 of data points randomly distributed as $\mathcal{N}(0, 10^2)$. We then fix $N = 100, \Delta t = .1$ and $\sigma = .1$. We then consider the resulting Kalman smoothing problem using both a least squares and a robust ℓ_1. loss. Figure 4.3 shows a comparison of these two estimates on reconstructing distance measurements from the origin. The robust estimate drastically outperforms the nominal least squares solver.

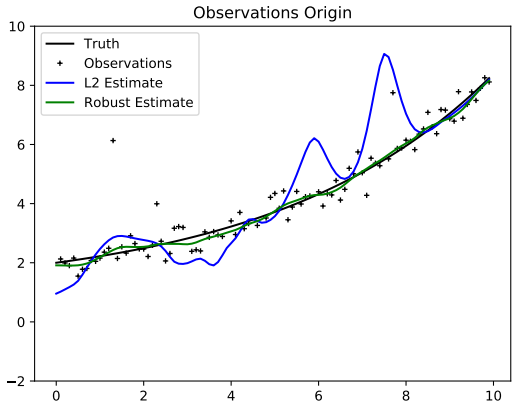


Figure 4.3: A comparison of robust and least squares estimates on range measurements from the origin. Some of the outleirs are not shown as they are too large.

Figure 4.4 shows the solvers recreation of the position and velocity in the Y dimension. Again the robust solver is much more accurate.

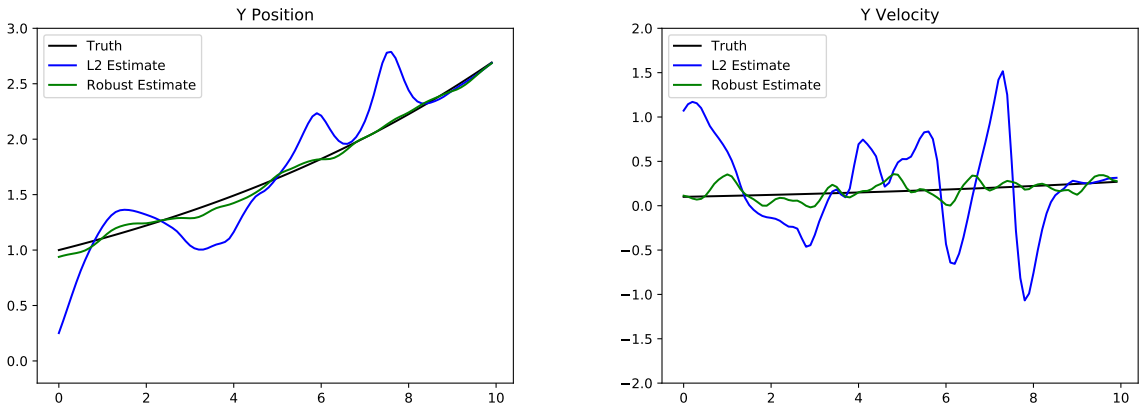


Figure 4.4: Robust and least squares solutions for the $Y(t)$ (left) and $\dot{Y}(t)$ (right). There are large improvements of the robust solver over least squares in both cases.

Chapter 5

PHYSICS-INFORMED MACHINE LEARNING FOR SENSOR FAULT DETECTION WITH FLIGHT TEST DATA

In this work we consider the problem of anomaly detection arising from sensor failure. These types of problems are well studied and solving them usually boils down to a feature selection problem. In general this can be difficult and depends quite a bit on the particular setting the problem is in. What we propose here is a fully data driven method that will work for a wide range of applications. First a model is created using a dynamic mode decomposition (DMD) method which then a Kalman filter is applied to. Then the difference between the Kalman filters prediction and incoming measurements is fed into a decision tree along with any other desired features. The main advantage of this is that it is data driven and a new method is not needed for every new setting, a common problem in feature selection.

5.1 Introduction

Sensor fault detection is an important problem in many fields of engineering, where monitoring and state estimation are required for a system's successful operation. Although catastrophic failures may be obvious, more insidious fault modes, such as decalibration, slow drift, and low-frequency oscillations, are more difficult to detect. Thus, there is a reliance in practice on engineering expertise and heuristics for anomaly identification. In the context of aircraft dynamics, redundant measurements are typically taken for critical quantities; a degree of robustness to faults is therefore achieved via a voting or weighted averaging sensor fusion scheme [3]. This is not always the case in flight test scenarios, where such large quantities of data are collected that it is impractical to either automatically detect faults

via redundancy or to manually monitor them. Fast, automatic anomaly detection can lead to reduced flight test time, resulting in significant cost savings for aircraft programs. Moreover, in routine operations, current designs require redundant sensing and flight controllers that are robust to faulty measurements. Robust and guaranteed fault detection may lead to designs with improved performance and reduced environmental impact [68].

Automated model-based fault detection has been well-studied for linear systems [107, 161, 153], but the general problem remains open for nonlinear dynamics [142]. When a nonlinear model is available, one approach to detect anomalous sensor behavior is to estimate the state with an extended Kalman filter [97, 158], and to then detect consistent discrepancies between the model and estimates by comparing observed deviations to an expected range of process noise [95, 155, 66]. This approach has been applied to flight dynamics models for identification [69] and automatic isolation [55] of anomalous sensors.

If a physics-based model is unavailable, data-driven methods offer an attractive alternative to detect faults, for example by training neural networks [111, 125, 110]. Although in recent years neural networks have had impressive successes in fields such as image and speech recognition [31], their behavior tends to be unreliable outside of training conditions: the precise regime that is arguably most important for investigating anomalous behavior. Indeed, deep learning models are well known to consistently fail in generalization and extrapolation tasks since they are interpolatory in nature [102]. An alternative is data-driven system identification [78, 79, 135, 134, 40, 151, 71, 39, 37, 38]. These methods can identify a dynamical model that is suitable for filter-based state estimation and fault detection [143, 142].

In this paper we build on previous work in data-driven model identification and anomaly detection, in particular [55] and [142], by combining optimal estimation and system identification with modern machine learning methods. We verify that the proposed approach is applicable to both strongly nonlinear dynamics and correlated, non-Gaussian process noise, and demonstrate fault detection on data from flight tests. A core component of the algorithm is a simple Kalman observer which is used to predict future sensor values based on current measurements. When the difference between this prediction and the observed values passes a

threshold over a period of time, a sensor fault is flagged. The Kalman filter requires a model of the underlying dynamics from which to estimate future states. This model is learned from data via the *dynamic mode decomposition* (DMD) [134, 91] with time-delays [35]. A decision tree [131] is then used to determine the right threshold for the gap between prediction and observation. The model is fully automatic; one need only specify a set of labeled training data and it will learn both a model for the dynamics present in the data and a set of rules for detecting sensor failures.

The key advantages of this approach are that it is fully data-driven, so that a model is not required for the dynamics underlying the system being monitored, only measurements. It is also automated and can readily support a large number of measurements/features. Its primary disadvantage is that it is a supervised method, meaning that one must supply labeled data to train the model. The techniques from which the model is built are all fairly general, granting it a large amount of flexibility while simultaneously restricting its accuracy for some specific applications. In domains in which underlying physical dynamics are well-understood, it may be advantageous to use a more specialized method.

The paper proceeds as follows. In Section 5.2 we give an overview of the mathematical background underlying our approach before describing the proposed method itself in Section 5.3. Section 5.4 discusses three example flight applications: one real-world flight test dataset and two simulated examples. We conclude with Section 6.6 which provides some final thoughts.

5.2 Background

In this section we provide a mathematical foundation for the proposed method. Sections 5.2.1 and 5.2.2 describe Kalman filters and their application to anomaly detection. A method for constructing a linear time invariant (LTI) model for use in a Kalman filter is discussed in Sections 5.2.3 and 5.2.4. Section 5.2.5 details a method of enriching the LTI model by introducing time-delays. Finally, Section 5.2.6 gives a brief discussion of decision trees, the last component of our method.

5.2.1 System identification and Kalman filtering

Since their development in the 1960's [83], various forms of Kalman filters have proven useful in fields ranging from robotics to weather prediction. The filter described in this section is a simple form of this powerful tool, but is nonetheless effective in many test problems. The method is essentially a simplification of those proposed in [66, 69, 55, 142], and references therein. It is thus potentially extensible to more complex detection and estimation problems, including those with underlying physical systems exhibiting strongly nonlinear dynamics.

Although application of Kalman filters to fault detection has been proposed since the 1970's, until recently an existing model was necessary for the method. Considering the scale of sensing in flight test applications, developing independent predictive models for the various sensors could be prohibitive. However, a recent development suggested in [142] was to identify a linear time-invariant (LTI) model that estimates the relationship between measurements using the DMD algorithm. DMD is a powerful method originally developed in the fluids community to study spatio-temporal coherence in high-dimensional numerical and experimental fluid flow data [134, 130, 151]. It has since found applications ranging from neuroscience to epidemiology [91]. The method is designed to efficiently extract dominant patterns from very large data sets and automatically uses correlations in the data for improved robustness.

A simple linear model identified with this algorithm may not be accurate enough to account for the complex interactions in the aircraft system. However, although Kalman filters are often used in full-state estimation, if the goal is not estimation but fault detection, the dynamic model does not need to be particularly accurate in order to capture anomalous behavior, as our results demonstrate. The first step of the procedure is therefore to identify a linear predictive model from a time series of *typical* measurements. Online, this DMD model is used to maintain a Kalman filtered estimate of the measurement of interest. The variance between the estimate and the actual measurement is monitored, and persistent deviations signal anomalous behavior.

5.2.2 Anomaly detection with Kalman filters

The following gives a simplified description of what will be used in our approach outlined in Section 5.3. More sophisticated methods can be found in [66, 69, 55, 142]. Let the vector of measurements we wish to monitor at discrete time step k be denoted by

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{y}_k, \tag{5.1}$$

where $\mathbf{x}_k$ is the observered measurement and $\mathbf{y_k}$ is the exogeneous observation. There does not need to be a direct causal relationship in the sense that actuation is usually taken in control theory; these measurements should just help to predict the next measurement of interest. Again, this model may be fairly inaccurate. The use of exogenous inputs is designed to stabilize the model and help to detect drift, high frequency noise, etc. This model may alternately be viewed as a linear regression predicting the next measurement. The simplest Kalman filter is a separate LTI "observer" system that maintains an estimate $\hat{\mathbf{x}}$ of the measurement of interest:

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\mathbf{y}_k + \mathbf{K}(\mathbf{x}_k - \hat{\mathbf{x}}_k). \tag{5.2}$$

The new term in this equation is the innovation $(\mathbf{x} - \hat{\mathbf{x}})$, the difference between the predicted and observered value of $\mathbf{x}_k$. which acts as feedback to stabilize the estimate. In general the Kalman gain $\mathbf{K}$ is a matrix, which can be chosen for optimal convergence of the estimate to the true state (by solving a Ricatti equation), given knowledge of the sensor and process noise covariances. In the example of Figure 5.1 (and for the numerical examples of Section 5.4), K is a scalar affecting the time sensitivity of the estimated state. We found that the method is relatively insensitive to the choice of Kalman gain, and values from 0.1 to 0.001 were tested with similar performance.

As suggested in [107, 69], a moving average of the innovation covariance can be used to identify anomalous behavior:

$$V_k = \frac{1}{N} \sum_{i=k-N}^{k} (\mathbf{x}_i - \hat{\mathbf{x}}_i)(\mathbf{x}_i - \hat{\mathbf{x}}_i)^\top. \tag{5.3}$$

Intuitively, this term will remain large when a measurement persistently yields anomalous behavior, provided the LTI model captures the important correlations between sensors. Even if the model is not particularly accurate, exogenous inputs can still produce an estimate that reflects anomalous behavior. It is worth emphasizing, however, that without significant additional validation there is no reason to believe that the Kalman filtered estimate is accurate.

For example, Figure 5.1 demonstrates the Kalman filter-based detection of a sensor failure in the real world data set (see Section 5.4.1). An anomaly can be flagged when the innovation covariance exceeds some threshold, which may itself be selected in an automated fashion. Although the filtered estimate is often inconsistent with both the faulty sensor and a redundant, working sensor, before and after the faulty sensor fails, the innovation covariance only grows significantly after the failure. In other words, the filtered estimate does not need to be accurate in order to be an accurate predictor of sensor faults.

5.2.3 Identifying a linear time-invariant model

The method described above requires a reasonably accurate model of the dynamics of the measurement of interest, possibly including the relationship between this measurement and the output of related sensors. In the simplest case (presented here) this could be a linear time-invariant (LTI) system, although application to a nonlinear model is possible using an Extended Kalman Filter (EKF) [66, 69, 55] or optimization-based nonlinear Kalman smoothing approaches [11, 17].

Although obtaining a model in general can be a labor-intensive and problem-specific task, recent developments in system identification have enabled a range of straightforward, efficient model estimation tools. The method presented here uses one such system identification algorithm: DMD. DMD was originally developed in the fluid dynamics community as a method to extract coherent spatio-temporal structures from complex, high-dimensional data [134, 130, 151]. As such, it is designed to take advantage of correlations in the data and reduce the underlying dimensionality of the model. Although there have been many theoretical and numerical refinements of DMD proposed (see e.g. [77, 122, 159, 160, 35, 112, 21]),

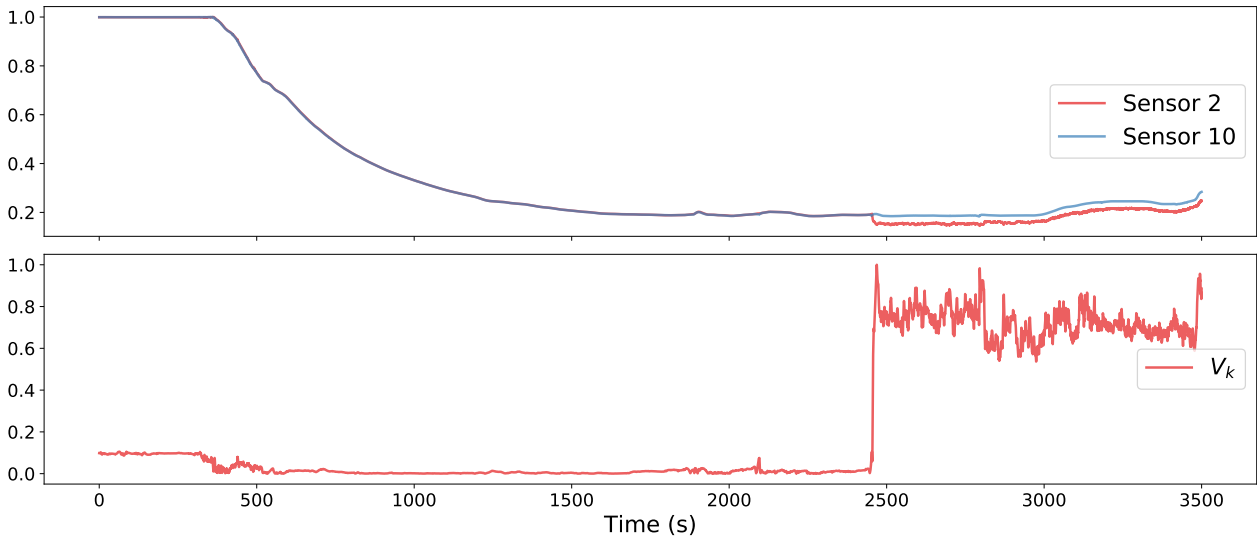


Figure 5.1: Anomaly detection for the flight test dataset of Section 5.4.1 with the Kalman filter. Top: measurements from two redundant sensors. Just before 2500 seconds, sensor 2 breaks and begins giving erratic readings. Bottom: a moving average, V_k, of the covariance term (see 5.3). Note that V_k remains negligible until the sensor failure event leads to persistent anomalous measurements relative to the Kalman filter.

we present a simple formulation of the method in the paragraphs that follow.

Suppose we have a series of measurements $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{m+1}\}$ and we assume that these are related by approximately linear dynamics, i.e.

$$\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k. \tag{5.4}$$

If we arrange the measurements in a time-shifted pair of matrices $\mathbf{X}$ and $\mathbf{X}'$ so that

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}, \qquad \mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_{m+1} \\ | & | & & | \end{bmatrix}, \tag{5.5}$$

these matrices are related by

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X}. \tag{5.6}$$

Denoting the pseudoinverse of $\mathbf{X}$ by $\mathbf{X}^\dagger$, the least-squares estimate of $\mathbf{A}$ is given by

$$\mathbf{A} \approx \mathbf{X}'\mathbf{X}^\dagger. \tag{5.7}$$

The spectral properties of the system are then estimated as usual by the eigendecomposition of $\mathbf{A}$.

For the case of high-dimensional systems, significant computational gains can be realized by reducing the dimensionality of the problem. The rank of $\mathbf{A}$ is limited by the minimum dimension of $\mathbf{X}$ and $\mathbf{X}'$. Instead of studying the spectral properties of the full-state system, we can project the high-dimensional state onto the leading principal components of $\mathbf{X}$ and approximate the spectrum of $\mathbf{A}$ by the spectrum of the matrix that steps this low-dimensional approximation forward in time. That is, if the singular value decomposition of $\mathbf{X}$ is given by

$$\mathbf{X} = \mathbf{\Psi}\mathbf{\Sigma}\mathbf{V}^*, \tag{5.8}$$

then the projection of an arbitrary snapshot $\mathbf{x}_k$ onto the leading r principal components is

$$\alpha_k = \mathbf{\Psi}_r^*\mathbf{x}_k, \tag{5.9}$$

where $\mathbf{\Psi}_r$ consists of the first r columns of $\mathbf{\Psi}$. The spectrum of $\mathbf{A}$ can be approximated by the spectrum of $\tilde{\mathbf{A}}$, where

$$\alpha_{k+1} \approx \tilde{\mathbf{A}}\alpha_k. \tag{5.10}$$

After some manipulation, we find that a least-squares estimate for $\tilde{\mathbf{A}}$ is given by

$$\tilde{\mathbf{A}} = \mathbf{\Psi}_r^* \mathbf{X}' \mathbf{V}_r \mathbf{\Sigma}_r^{-1} \mathbf{\Psi}_r. \tag{5.11}$$

For this anomaly detection application, the system dimensionality n will typically be much lower than the number of available time steps m, so the computation of $\mathbf{A}$ by equation (5.7) is tractable. However, we still use a dimensionality reduction approach, partly to take advantage of correlations in the time series and partly to make the method scalable to larger problems. To estimate the full system matrix $\mathbf{A}$ would require only a slight modification to equation (5.11):

$$\mathbf{A} = \mathbf{X}' \mathbf{V}_r \mathbf{\Sigma}_r^{-1} \mathbf{\Psi}_r. \tag{5.12}$$

Thus DMD provides an automated means of constructing an LTI model evolving the measurements of interest $\mathbf{x}$ in time, but we would also like this model to take into account readings from other sensors. Put another way, DMD finds the matrix $\mathbf{A}$ of equation (5.2) and must be extended to produce $\mathbf{B}$.

5.2.4 Dynamic Mode Decomposition with Control (DMDc)

The full LTI system $(\mathbf{A}, \mathbf{B})$ can be estimated via a slight modification to the DMD procedure developed by Proctor, *et al.* called *DMD with control* (DMDc) [122]. We split the full state vector into measurements of interest, $\mathbf{x}$, and exogenous "inputs" $\mathbf{y}$. These inputs are treated as *actuation* in the Kalman filter model, but are more accurately taken to be simply exogenous predictors of the sensor measurements. As with the state vectors, the input vectors can be compiled into a single matrix $\mathbf{\Upsilon}$. The estimation then proceeds similarly (as explained in

detail in [122]). First data matrices are constructed

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}, \quad \mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_{m+1} \\ | & | & & | \end{bmatrix}, \quad \boldsymbol{\Upsilon} = \begin{bmatrix} | & | & & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_m \\ | & | & & | \end{bmatrix}.$$

The dynamics with control may be written in terms of these data matrices as

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X} + \mathbf{B}\boldsymbol{\Upsilon}.$$

Finally, the system matrices may be obtained by regression

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} = \mathbf{X}' \begin{bmatrix} \mathbf{X} \\ \boldsymbol{\Upsilon} \end{bmatrix}^{\dagger}.$$

With the singular value decomposition

$$\begin{bmatrix} \mathbf{X} \\ \boldsymbol{\Upsilon} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Psi}_1 \\ \boldsymbol{\Psi}_2 \end{bmatrix} \boldsymbol{\Sigma} \boldsymbol{V}^*, \tag{5.13}$$

the matrices $\boldsymbol{\Psi}_1$ and $\boldsymbol{\Psi}_2$ now give the principal components of the state and input subspaces, respectively. Dimensionality reduction based on the singular values $\boldsymbol{\Sigma}$ is also possible at this stage.

The system $(\mathbf{A}, \mathbf{B})$ is finally estimated by

$$\mathbf{A} = \mathbf{X}' \boldsymbol{V} \boldsymbol{\Sigma}^{-1} \boldsymbol{\Psi}_1^* \tag{5.14}$$

$$\mathbf{B} = \mathbf{X}' \boldsymbol{V} \boldsymbol{\Sigma}^{-1} \boldsymbol{\Psi}_2^*. \tag{5.15}$$

5.2.5 Time delays: HAVOK and Delay-DMD

For complex dynamics, a standard linear system may not have enough descriptive ability to serve as a model for the Kalman filter fault detection method. One approach is to enrich the library with nonlinear functions of the state, leading to *Extended DMD* (EDMD) or *Koopman Mode Decomposition* (KMD) [159]. EDMD/KMD has been demonstrated to yield models

that are predictive enough for accurate full-state estimation with Kalman filters [143], and for sensor fault detection in a power grid model [142]. In the latter work, anomalies were detected using hypothesis testing for the distribution of normalized innovation squared in the Kalman filter under the assumption of Gaussian white noise for process disturbances. However, accurate EDMD/KMD models rely on a judicious choice of observable functions [36, 112], which can be challenging in practice; the data matrices quickly become ill-conditioned as the number of observables is increased.

For systems that cannot be accurately represented with standard DMD, we instead augment the library with time-delayed measurements. The use of time-delays has a long history in system identification, including the widely used Eigensystem Realization Algorithm and Observer Kalman Identification [78, 79] methods, and deep connections to dynamical systems theory [146, 35]. Augmenting the state vector with time-delays allows the model to capture some of the effects of latent variables. For example, consider a simple harmonic oscillator in periodic sinusoidal motion. A first-order one-dimensional linear model is only capable of expressing exponential growth and decay, not oscillatory dynamics. However, if the state is augmented by a time delay of 1/4 period, a linear model can effectively capture the second-order dynamics (or the latent, imaginary component of motion). Applying DMD to a time-delay-augmented vector can therefore give highly accurate representations of quasiperiodic dynamics [43].

The modeling and estimation procedure is effectively the same, except that a scalar measurement x_k at time t_k is replaced by a vector $\mathbf{x}_k = \begin{bmatrix} x_k & x_{k-d} & x_{k-2d} & \cdots & x_{k-n_d d} \end{bmatrix}^\top$, where d is the length of each delay and n_d is the number of delays. For anomaly detection applications, only the innovation corresponding to the current time step is tracked.

5.2.6 Decision trees

Decision trees are a popular machine learning method for both classification and regression problems [131, 129]. In this work we are interested in classification. Given a vector of real-valued features (e.g. sensor measurements or the innovation covariance), a decision

tree uses a set of threshold-based rules to determine whether a failure has occurred[1]. The rules are learned during a training phase in which labeled examples (the labels tell the tree which examples correspond to sensor failures) are shown to the tree. Once trained the rules are, in essence, a series of if-else statements with conditional expressions checking whether features are above or below different threshold values. We use the Scikit-learn decision tree implementation [120], trained with the CART algorithm [33].

Decision trees are a natural choice in this context because they automate and generalize the threshold-selection process. Rather than choosing, through trial and error, a cutoff in the innovation covariance above which a sensor failure is deemed to have occurred, a decision tree automates the choice. This approach also allows the model to take into account other sensor-to-sensor interactions not captured by the Kalman filter. In particular, the decision tree is able to set different innovation covariance thresholds for different regimes in the dynamics (e.g. perhaps one threshold is appropriate for low angle-of-attack maneuvers and another is better suited to high angle-of-attack flight patterns). This is especially important for the dataset of Section 5.4.1.

Decision trees have additional benefits relevant to our use-case: their decision mechanism is interpretable, they scale extremely well to large datasets and are fast to evaluate in an online setting, and they are able to identify which features are most useful. Without regularization they tend to overfit training data, so we limit the depth of the trees used in our experiments.

5.3 The proposed method

The following gives a high-level overview of the algorithmic method used for sensor failure detection. We break the process into an offline phase where the model is calibrated using training data, and an online phase where the model is deployed to detect sensor fault events

[1]In this work the only features passed to the decision tree are raw sensor values and the average innovation covariance. However, it is common practice to engineer other features tailored to the specific problem domain.

in real time.

1. **Offline phase:** Given a training set $\mathcal{D}_{train}$ consisting of sensor measurements at various time points and corresponding labels,

   (a) Compute the DMDc system $(\mathbf{A}, \mathbf{B})$ as in Section 5.2.4 using time-delayed measurements described in Section 5.2.5;

   (b) Derive any desired features from $\mathcal{D}_{train}$ to be used with the decision tree. This includes the average innovation covariance (5.3), which can be computed using $\mathbf{A}$ and $\mathbf{B}$;

   (c) Train the decision tree using the derived and raw features.

2. **Online phase:** Given a series of measurements $\mathbf{x}_1, \mathbf{x}_2, \ldots$

   (a) Compute any features expected by the decision tree, using the previously constructed $\mathbf{A}$ and $\mathbf{B}$ to compute the average innovation covariance;

   (b) Pass the features into the decision tree to obtain a class prediction (either that the sensor has failed or continues to function properly).

The Kalman filter-based anomaly detection method described in this section can be applied to all available sensors simultaneously. The innovation covariance then becomes a matrix whose diagonal entries can be monitored to identify faults in the corresponding sensors. Note that in this work we restrict our attention to models for a single scalar state x. As such the decision tree is trained to output a binary result for a single sensor, although extension to parallel detection for multiple sensors is straightforward.

5.4 Three example applications

5.4.1 Real-world dataset

We first consider detecting sensor failure using anonymized (scaled to lie in $[-1, 1]$) measurements from aircraft sensors collected during flight tests. We focus on detecting faults in

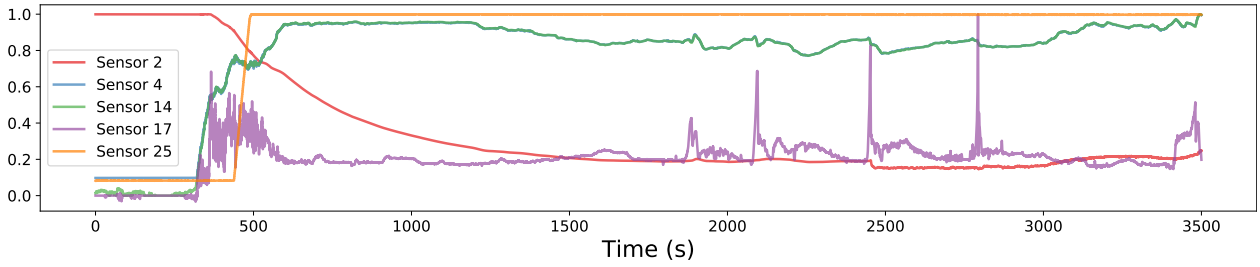


Figure 5.2: A plot of data from a subset of the sensors for test flight 1. Measurements from the faulty sensor (Sensor 2) are shown in red. The sensor fails just before second 2500, where there is a short drop followed by erratic, noisy measurements. We plotted maximally important features as returned by a decision tree trained on raw sensor data to predict failure events.

a given sensor that has a high failure rate (sensor 2) using the data from 25 other on-board sensors. Several sensors capture similar or redundant information which the DMD model can exploit. In total there are 21 flights, each roughly seven hours in length, with measurements recorded at a frequency of 20 Hz. Sensor 2 fails in 14 of the test flights. There is one time series in which the sensor failure was detected, sensor 2 was fixed, and then broke again. This case was split into two separate time series. When sensor 2 fails there is typically a small constant shift in the data followed by increased noise for the duration of the flight. Such a failure is shown along with anonymized data from some of the more relevant sensors in Figure 5.2.

In this real data setting (data are from actual flights), the time of sensor failure time must be inferred from looking at the data. We have hand-labeled estimated time of failure in each case; reported time to detection is based on these estimates. Systematic error in the absolute detection time is therefore possible, but comparisons between flight tests should be reliable.

In order to train the decision tree the flights are divided into a training and testing set (no

separate validation step is necessary as all parameters are determined with cross-validation[2] using the training data). The training set consists of data from six test flights, flights 0 through 5, four of which contain a sensor failure.

The DMD model is calibrated using flight 3, which contains no failure events. Notably, the model learns to predict future values of Sensor 2 by averaging together the current measurement from Sensor 2 with those of three other sensors with redundant signals (Sensors 0, 1, and 10). Each is given roughly equal weight. This is reminiscent of a weighted-mean method for consolidating redundant measurements into one fault tolerant estimate [3]. From Figure 5.1 it is easy to see that something like the difference between sensors 2 and 10 would be a good proxy for when sensor 2 is behaving properly, but we stress that the model figures this out *automatically*.

A decision tree is then trained to predict sensor faults with its hyperparameters being selected using five-fold cross validation. Examples are reweighted before being fed to the tree in order to mitigate the effects of class imbalance. The top ten signals along with their feature importances (with respect to the decision tree) are given in Table 5.1.

As seen in Table 5.1 most of the importance is concentrated in just two of the features, with the majority of the weight going to the moving average of the innovation covariance. The second-ranked feature, Sensor 25, turns out to be an almost binary signal indicating when measurements should be collected from the sensor of interest.

The results of the detection of sensor failures on the flights in the test data set is summarized in Table 5.2. We show performance metrics for each individual flight so that poor results can be more easily investigated. As there are so many data points for each flight, standard metrics such as precision (the proportion of flagged examples that were actually true positives) and recall (the proportion of true positives we were able to detect) are not particularly helpful here[3]. Instead we focus on the number of false positives and false nega-

[2]Cross-validation done using sklearn library for decision trees (tree) along with cross-validation (GridSearchCV).

[3]True positives and true negatives are positive (failed sensor) and negative (working sensor) examples correctly classified by the model. False positives are negative examples the model classified as belonging

Feature	Importance
V_k	0.7514
Sensor 25	0.1985
Sensor 4	0.0261
Sensor 5	0.0095
Sensor 9	0.0057
Sensor 14	0.0056
Sensor 17	0.0012
Sensor 22	0.0008
Sensor 13	0.0005
Sensor 7	0.0002

Table 5.1: Feature importance for the decision tree trained using the flight test dataset.

tives along with lag time (time from actual sensor failure to detection). Overall accuracy is also included to give more context to the number of false positives and negatives reported.

Assuming an end goal of using our model to alert a human of potential sensor failure during a flight test within a few seconds of occurence, the model performance is promising, with the exception of two flights: 16 and 19. There is some fluctuation in lag time, most likely due to circumstances that affect flight dynamics. The false negatives are almost entirely due to lag time. The false positives tend to persist briefly (about two seconds) before automatically correcting. For example for flight 8 we observe four different short periods when the model thinks a sensor failure has occurred. These correspond to abrupt changes in flight conditions as the pilot(s) carry out different test maneuvers.

---

to the positive class, i.e. instances where the sensors are working properly, but the model erroneously predicts a sensor has failed. Similarly, false negatives are positive examples the model thought were negative examples.

Flight	Total examples	False positives	False negatives	Accuracy	Lag Time (s)
6	175,162	708	0	0.995958	N/A
7	213,964	0	22	0.999897	1.1
8	143,140	0	4,663	0.967424	10.1
9	121,413	0	412	0.996607	1.35
10	368,140	0	0	1.000000	N/A
11	152,146	0	130	0.999146	6.5
12	278,465	50	0	0.999820	N/A
13	372,950	3	0	0.999992	N/A
14	124,570	2,617	0	0.978992	N/A
15	166,880	0	0	1.000000	0
16	302,550	0	11,690	0.961362	∞
17	295,680	0	63	0.999787	2.9
18	64,700	0	9	0.999861	0.45
19	472,090	59,809	157	0.872978	7.85
20	117,650	0	229	0.998054	11.45

Table 5.2: Prediction results for commercial test flights.

Now we turn our attention to the two flights with poor model performance. In flight 16 there are a large number of false negatives (and a lag time of ∞) as the model completely misses an actual sensor failure. This is because sensor 25, which is normally active during the tests, was completely inactive throughout the entirety of this flight. If one wished to learn to predict regardless of whether or not this sensor was in use, one may need to resort to training a separate model using only data where the sensor was inactive, because it is used in all the other flights.

In flight 19 we see a large number of false positives. This is caused by persistent discrepancies between sensor 2 and its redundant sibling, sensor 10. We suspect that sensor

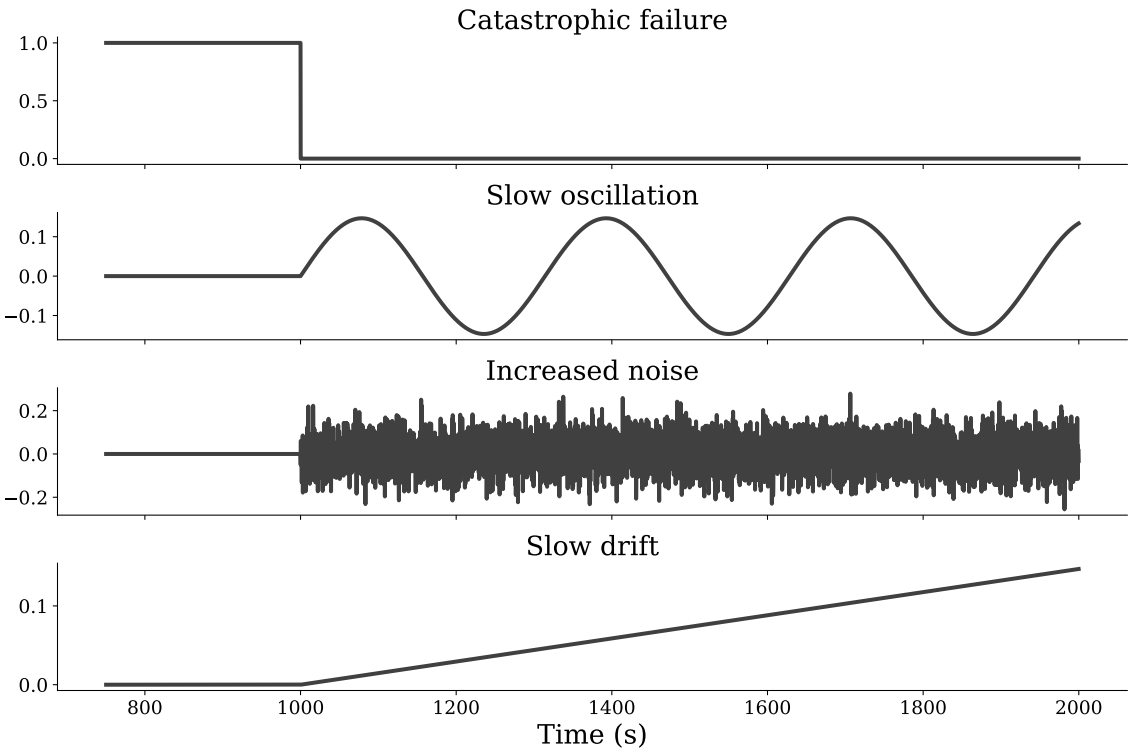


Figure 5.3: The sensor faults applied to the simulated datasets. In particular, the faults used for the dataset of Section 5.4.2 are shown here. Those used for the dataset of Section 5.4.2 are the same up to rescaling.

10 temporarily malfunctioned. Whether or not it is desirable for the algorithm to flag these anomalous measurements depends on the application. After the abnormal behavior of sensor 10 the model detects the failure of sensor 2 almost perfectly.

The final model is able to reliably detect true sensor failure events within 12 seconds (with most occurring in under five). There are infrequent false positives detected with a brief persistence of about two seconds. In some cases the measurements taken by the faulty sensor drift close to the values of the reliable redundant sensors, leading to false negatives.

5.4.2 Synthetic datasets

In order to better understand the capabilities of our proposed approach we apply it to two sensor failure tasks derived from simulated datasets. In both instances we use a dynamical model to generate realistic sensor data, which we then augment in various ways to mimic common sensor failure modes. Specifically, we simulate the faults shown in Figure 5.3, similar to those studied by Eykeren and Chu [55]: catastrophic failure (multiplicative), slow oscillation (additive), increased noise (additive), and slow drift (additive). In every case we add a small amount of white noise (mean 0, standard deviation 5×10^{-3}) to the underlying sensor measurement of interest *after* incorporating the fault.

Goman-Khrabrov model

A long-standing challenge in aerodynamic modeling was capturing the effect of a separated flow on aerodynamic moments. This arises for example in high angle-of-attack maneuvers, when an airfoil wake can detach, leading to much more complex physical behavior such as dynamic stall [94]. The aerodynamic moments in this case depend not only on the airfoil configuration, as in standard linearized approaches based on stability and control derivatives, but also on the state of the separated flow. Because of their importance in a range of unsteady fluid dynamic contexts, dynamic stall, and separated flows more generally, have been widely studied [137, 4, 101, 145, 116, 47, 51, 5, 144, 54].

Goman and Khrabrov proposed a mathematical model of dynamic stall that treats the flow state as a dynamic internal system variable [67]. For the case of a high angle-of-attack airfoil this is a scalar variable $x \in (0, 1)$ representing the separation point normalized by the chord length, so that fully attached flow corresponds to $x = 1$. The internal flow field dynamics are modeled with a simple time-delay model:

$$\tau_2 \dot{x} + x = x_0 \left(\alpha - \tau_1 \dot{\alpha}\right). \tag{5.16}$$

The function $x_0(\alpha)$ defines the empirical steady separation point as a function of angle of attack α. Quasisteady effects are expressed through the time-delay shift $\tau_1 \dot{\alpha}$. The overall

model dynamics are a relaxation towards the quasisteady separation point on a timescale τ_2. The moments are then algebraic functions of the aerodynamic state, for example $C_L = C_L(\alpha, x)$. For a high angle-of-attack airfoil, the model

$$C_L(\alpha, x) = \frac{\pi}{2} \sin\left[\alpha(1 + \sqrt{x})^2\right] \tag{5.17}$$

was shown to accurately describe experimental data for a NACA 0015 airfoil [67].

For our synthetic data, we use a simple model for the steady separation point:

$$x_0(\alpha) = \frac{1 - \tanh\left[20(\alpha - 0.25)\right]}{2}. \tag{5.18}$$

This produces the expected hysteretic behavior, although it is not expected to accurately represent the steady separation point of any particular airfoil. The time constants τ_1 and τ_2 in the model can be obtained in general by fitting to experimental data. We use the reported values $\tau_1 = 0.5$ and $\tau_2 = 4.5$, nondimensionalized by chord length and free stream velocity [67].

The separation point and lift coefficient for a sinusoidal pitching motion at nondimensional frequency $\omega = 0.05$ is shown in figure 5.4, along with the steady values as a function of angle of attack. The effect of the model is to capture observed hysteresis in the curves; the flow remains attached to higher angle-of-attack on pitch-up motions and stall is delayed. Conversely, when the flow is separated during a pitch-down maneuver it remains so for longer, resulting in reduced lift relative to the steady value.

Our experimental dataset consists of measurements of C_L, α, and $\dot{\alpha}$ taken five times per second for 2000 seconds. The DMDc model is trained using the full time-series. We then simulate failure of a hypothetical C_L sensor at $t = 1000$ using each of the fault modes shown in Figure 5.3, which results in the time series given in Figure 5.5a. We then precompute the innovation covariance, V_k, for each sensor failure type, with the Kalman filter (5.2) constructed using DMDc. The scale of V_k differs between the modes so only training on one type of sensor failure may lead to poor results when trying to detect different failure types. We plot the moving average of the covariance for the different failure modes in Figure 5.5b.

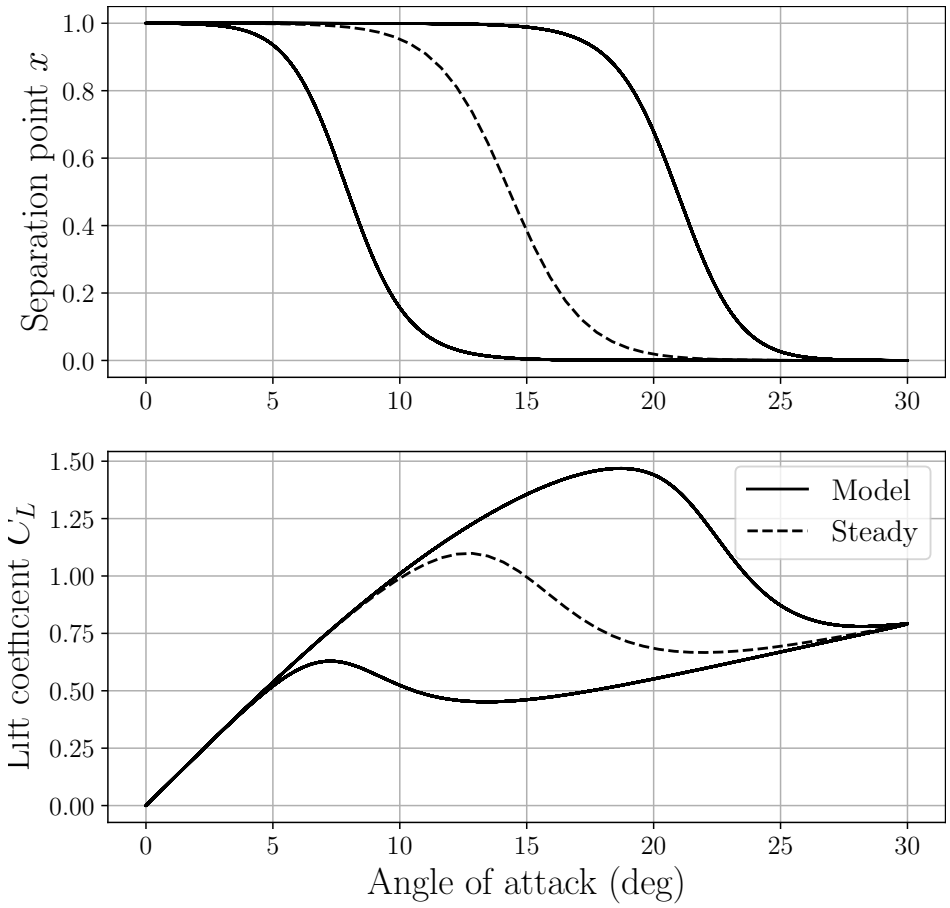
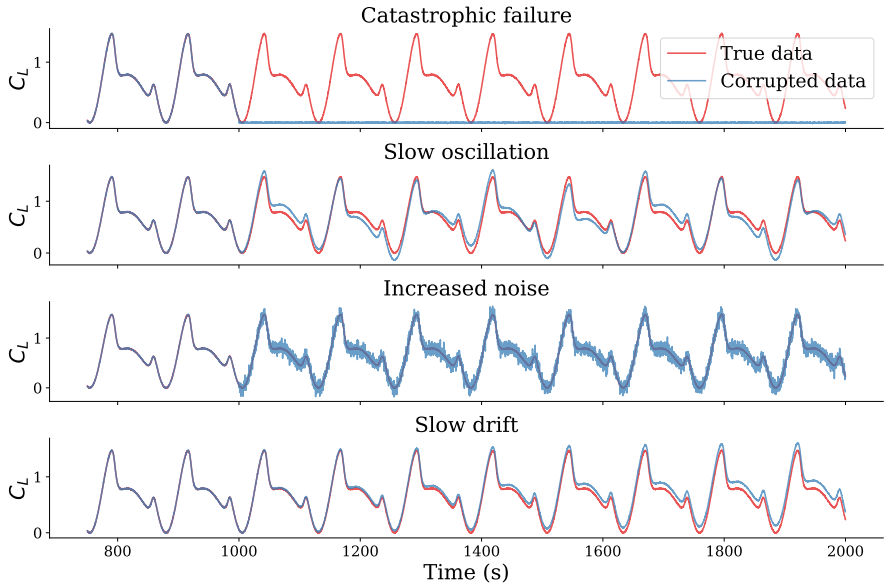
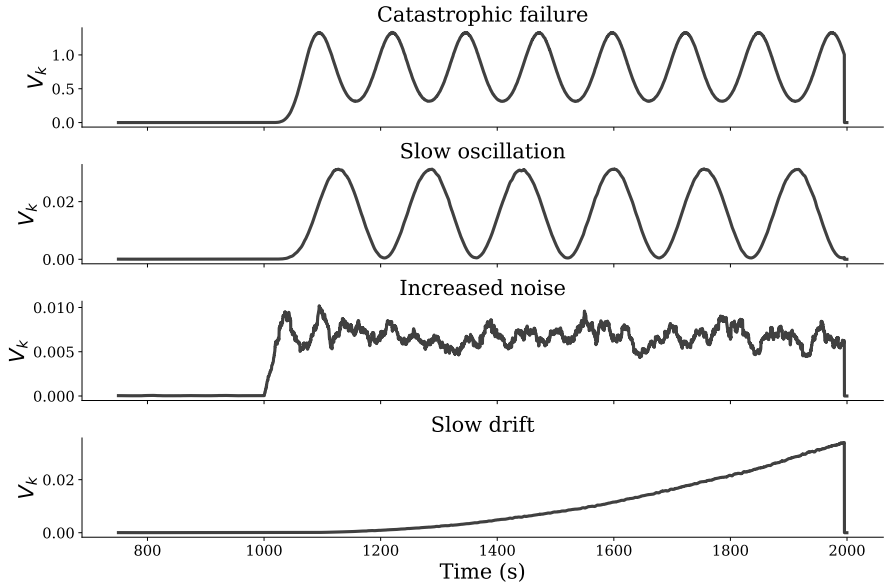


Figure 5.4: Hysteresis in flow separation (top) and lift coefficient (bottom) in the Goman-Khrabrov model for an airfoil undergoing sinusoidal pitching motion at nondimensional frequency $\omega = 0.05$. Stall is delayed relative to the steady value for pitch-up motions with attached flow (upper curves on both plots).



(a) Simulated data



(b) Innovation covariance

Figure 5.5: (a) Visualizations of the different lift coefficient (C_L) sensor failure modes for the Goman-Khrabrov model. (b) Moving average of the innovation covariance for each sensor fault type using data generated with the Goman-Khrabrov model. Sensor failure occurs at $t = 1000$. Note that we omit from this plot measurements taken with $t < 750$.

Note the differences in scale between the modes. A threshold for V_k, above which a sensor failure is deemed to have occurred, which is chosen based on only one type of fault, may be inappropriate for the others.

Finally, we train a decision tree to predict when a sensor failure has occurred. The tree is given access to α, $\dot{\alpha}$, the corrupted C_L measurements, and V_k as features. At each time point the model must attempt to predict whether the given C_L measurement has been corrupted or not (whether the sensor has failed). We use five-fold cross-validation to select model parameters. We construct two types of training and testing data; the first involves incorporating random examples from *all four* failure types into the training set and the second sources its training data from just one sensor fault and attempts to predict when the others have occurred. In each case the training set contains 30,000 examples and the testing set contains 10,000 examples.

Model performance is summarized in Table 5.3. "Train accuracy" refers to the accuracy of the model on a holdout set during cross-validation. For models 1-4 this number gives an estimate of the models' accuracies on the fault type on which they were trained. The best accuracy and precision scores are both achieved by the model with training data from all four fault types. However, model 4 has the best recall, meaning that it misses the fewest sensor failure events. This is likely due to the fact that model 4 must choose a very small threshold for V_k at which to separate negative and positive class instances. All five models list the average innovation covariance as their top feature.

The proposed method reliably detects sensor faults for data generated from the Goman-Khrabrov model, even on unseen fault types. However, if either the amplitude or frequency of the forcing term changes after the DMD model has already been trained, the DMD model becomes too inaccurate to be useful and overall predictive performance suffers considerably. This is a general drawback of data-driven models: when training and testing sets are different enough in distribution, models learned on one set have a hard time generalizing to the other.

Model	Fault types seen	Accuracy	Precision	Recall	Train accuracy	Tree depth
1	Catastrophic failure	0.9732	0.9980	0.9483	0.9343	5
2	Slow oscillation	0.9672	0.9629	0.9718	0.9351	**2**
3	Increased noise	0.9730	0.9993	0.9467	0.9482	**2**
4	Slow drift	0.9784	0.9707	**0.9867**	0.8678	3
5	All	**0.9818**	**0.9996**	0.9642	**0.9823**	5

Table 5.3: Performance metrics for models trained on different subsets of sensor fault types with data generated with the Golman-Khrabrov model. The best values for each column are bolded.

Flight dynamics model

Motivated by anomaly detection in a flight test setting, we also consider a longitudinal flight dynamics model for a business jet in atmospheric turbulence [141]. The equations of motion for the longitudinal model capture motions in the forward and vertical directions, including pitch for a total of three degrees of freedom. Flight controls are included for elevator, thrust, flaps, and stabilator; these are trimmed for steady, level flight. The aerodynamic model includes a realistic geometric configuration, stability and control derivatives, US standard atmospheric conditions interpolation, and a Mach number correction. The dynamics are forced by atmospheric turbulence generated to approximate the von Kàrmàn spectrum by filtering band-limited white noise [108, 157].

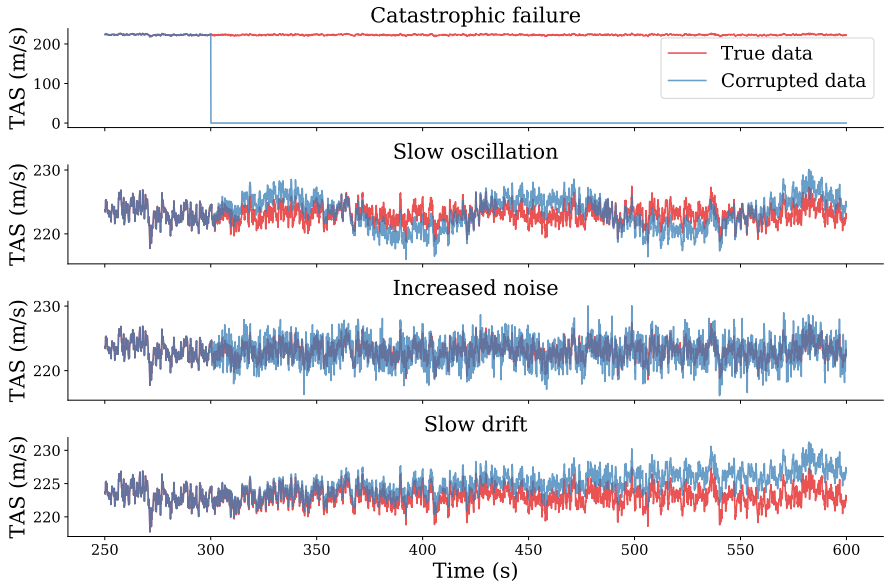
From this system we can measure not only the dynamic variables for inertial velocity and pitch, but also lift, drag, pitching moment, true airspeed, angle of attack, and Mach number. We consider measurements of the true airspeed (TAS), informed also by angle of attack, inertial airspeed, pitch, lift, and thrust. Note that the flight controls are constant, but the model includes an altitude correction for effective thrust. We generate samples for each variable over $t \in [0, 600]$ at a rate of 10 samples per second. Next we build a Kalman filter from a DMDc model trained on the TAS data. Various faults are then introduced

starting at $t = 300$ in the TAS sensor to obtain the time series shown in Figure 5.6a. These measurements are much noisier than those from the previous section due to the turbulence-based forcing.

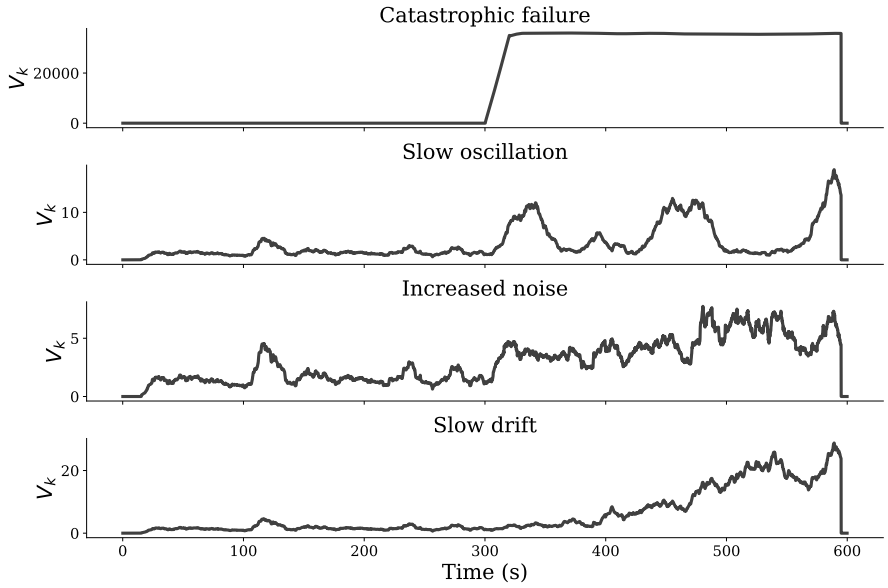
Again we precompute the innovation covariance, V_k, for each fault type. The results are shown in Figure 5.6b. It is evident from this plot that all faults, except the catastrophic failure, will be difficult to detect. None of the other covariance time series admit a single threshold that distinguishes bad readings from good ones. As before we study the behavior of decision trees trained on a data from all four failure modes or from just one mode. The trees are given measurements of TAS, angle of attack, inertial air speed, pitch, lift, thrust, and V_k. Training and testing sets are of equal size, with each training set constituting 75% of the data. Since the dataset is more complex we allow our cross-validation procedure to select trees of depth up to seven.

Table 5.4 details the results of these experiments. Model 5 outperforms all the others by a considerable margin, though it is also the most complicated model. Since the decision tree sees examples of each failure type, it is able to construct a more complex set of thresholding rules for different situations. In contrast, the other models consist of trees which appear to *overfit* to the failure modes they are shown, as evidenced by their diminished accuracy scores relative to model 5. Notably, model 1 adopts a classification rule allowing it to predict catastrophic sensor failures with perfect accuracy (its in-group accuracy is 1), however for all other failure types it is no better than a coin toss. The threshold it selects for V_k is much too large for the other fault types and so it almost always predicts that no sensor failure has occurred. In this case model 2 has the highest recall, but also a low precision, for the same reason as model 4 did previously: its threshold for V_k is set lower than the others, reducing the number of false negatives at the cost of more false positives. The average innovation covariance has the highest feature importance for each of the five models.

The proposed method is effective at identifying sensor faults in simulated flight test data, but there is a noticeable degradation in performance relative to data simulated with the Goman-Khrabrov model. This is due in part to the complexity of the simulated dynamics as



(a) Simulated data



(b) Average innovation covariance

Figure 5.6: Flight dynamics model data. (a) Visualizations of the different true airspeed (TAS) sensor failure modes. (b) Moving average of the innovation covariance for each sensor fault type. Sensor failure occurs at $t = 300$. Note that we omit from (a) measurements taken with $t < 250$.

Model	Fault types seen	Accuracy	Precision	Recall	Train accuracy	Tree depth
1	Catastrophic failure	0.5002	0	0	1.0	**2**
2	Slow oscillation	0.9549	0.9230	**0.9926**	0.8412	3
3	Increased noise	0.8974	0.9915	0.8016	0.8878	**2**
4	Slow drift	0.9572	0.9659	0.9478	0.8190	4
5	All	**0.9867**	**0.9961**	0.9767	**0.9859**	7

Table 5.4: Performance metrics for models trained on different subsets of sensor fault types for the flight dynamics model. The best values for each column are shown in bold.

well as the erratic atmospheric forcing. We observe that for this more challenging dataset, it is increasingly important that the model be trained using examples from multiple sensor failure modes, allowing it to establish different decision thresholds for different types of dynamics.

If data for some failure types is unobtainable, then the model should be trained using fault types that are most difficult to detect. Failure modes that can be detected trivially, such as catastrophic sensor failure, may prove insufficient to train a robust detector.

5.5 Conclusion

We have developed a fully automatic approach to detect sensor failures in systems with multiple types of sensor failures. The method first uses the dynamic mode decomposition for control with time-delay measurements to learn a simple linear time-invariant model for the evolution of a sensor of interest in time. This model is embedded in a Kalman observer which is then used to predict future measurements. A potential sensor fault is detected when the predicted and measured sensor values disagree by a large margin, with the margin size selected using a decision tree. All components are trained automatically. The performance of the proposed method was demonstrated on three test datasets: real measurements from a series of flight tests and two simulated datasets from the Goman-Khrabrov and a realistic

flight dynamics model. In each case the difference between the true and Kalman-observer-predicted values of the sensor of interest provided an accurate proxy for when sensor failure had occurred.

There are numerous extensions that could be explored for improving upon the results obtained here. Any of the components of the algorithm could be replaced with more sophisticated variants. For example, advances in Koopman theory could be leveraged to enrich the linear time invariant physics model. A nonlinear model such as an extended Kalman filter or a more general estimator [10] or a model learned via some other model discovery framework [135, 39, 80] could be used in place of the Kalman filter. Such generalizations would allow for the application of the proposed method to systems exhibiting strongly nonlinear dynamics. It would also be interesting to incorporate this analysis within the context of robust statistics [42], which has recently been shown to improve flow measurements [132]. The performance of the decision tree could be enhanced by employing an ensemble [96], a cost-sensitive training algorithm [99], or by better utilizing class probabilities output by the tree.

Acknowledgments

The authors acknowledge support from the Boeing Company. JNK also acknowledges support from the Air Force Office of Scientific Research (FA9550-19-1-0011). SLB also acknowledges support from the Air Force Office of Scientific Research (FA9550-18-1-0200).

Chapter 6

ROBUST SINGULAR SMOOTHERS FOR TRACKING USING LOW-FIDELITY DATA

This work can be thought of as a fully fleshed out example of the model in chapter 2 applied to a real (and quite difficult) data set. One of the benefits of the reformulated model is its flexibility and we demonstrate this using a a dataset that is quantized, noisy and biased. The bias is removed by adding a term to the state variable which leads to a singular covariance. In order to account for the quantization and noise we create a custom penalty with linear tails and a 'dead zone' around the origin. These all can be done within the framework of chapter 2. We finish with experimentation on adding different amounts of additional data as a proof of concept of this model in an autonomous navigation setting.

6.1 Introduction

State-space models are ubiquitous in signal processing, and allow integration of disparate measurements to inform estimation, decisions, and control. Classic filtering [81, 84] and smoothing [123, 63] are core tools used to estimate these models. Their dependence on high-fidelity data, driven by Gaussian assumptions on errors and innovations, has demanded unequivocal attention from researchers and practitioners, and inspired robust dynamic inference methods.

While early robust approaches [85, 133] sought to modify iterations of the Kalman filter (KF) and Rauch-Tung-Striebel (RTS) smoother, over the last 25 years researchers have used *robust formulations* to weave assumptions on errors and innovations directly into the estimation problems themselves [57, 24, 86, 14, 60, 16, 18]. Constraints, when available, are also readily incorporated into the problem formulation [29]. Specifying the formulation

leaves one free to choose from a range of optimization algorithms; the survey [10] describes a general class of models as well as first- and second-order methods to solve them.

Our focus is on models with singular variances for process and measurement residuals. These models are excluded by the assumptions of generalized smoothing [10] and all of the various special cases cited in the survey. In this paper, we build on the recently proposed framework of [76] for singular models, and systematically develop complementary modeling elements: **robust penalties, informative constraints, and singular models**. The resulting approach exploits the structure of singular covariances head on rather than using workarounds such as pseudo-inverses or variance boosting that either do not work in the general setting or introduce unnecessary changes to the fundamental model (see discussion in [76]). A simple synthetic tracking example with a singular process model shows how

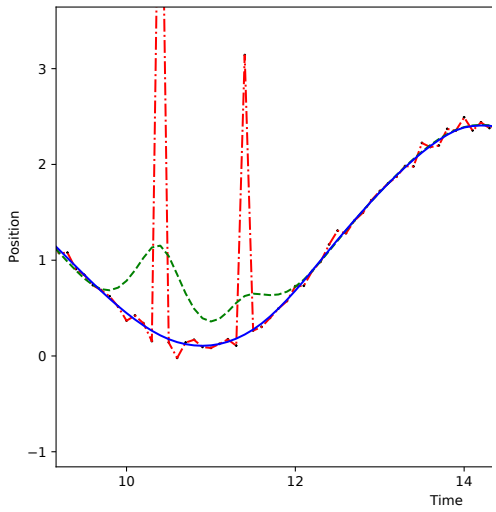


Figure 6.1: We track a simple trajectory in the presence of outliers. The red dash-dot shows a'robust' Huberized approach implemented using a pseudo-inverse; green dash shows the proposed singular ℓ^2 estimate; blue solid shows the proposed singular Huber estimate, which clearly tracks the true state.

the common tack of replacing the inverse by the pseudo-inverse fails dramatically in the

presence of outliers (Figure 6.1). Any robust approach requires control of the null spaces associated to process and observations. We develop a direct practical formulation and method, and test it on a batch smoothing analysis of real field data, with a view towards real-time implementation in future work.

Background. Our main goal is to infer an unobserved state sequence $x_1, \ldots, x_N$ from noisy observations $y_1, \ldots, y_N$ using the model:

$$\begin{aligned} x_1 &= x_0 + w_1 \\ x_k &= G_k x_{k-1} + w_k, \quad k = 2, \ldots, N \\ y_k &= H_k x_k + v_k, \quad k = 1, \ldots, N, \\ x_k &\in X_k, \quad \text{each } X_k \text{ polyhedral,} \end{aligned} \tag{6.1}$$

where x_0 is a given initial state estimate, G_k and H_k are linear process and measurement models, $y_1, \ldots, y_N$ are observations, and X_k specify additional information through constraints. The framework of [10] assumes that w_k and v_k are mutually independent random variables with known *nonsingular* covariances Q_k and R_k, and that they follow from log-concave distributions; in particular they may be non-Gaussian.

Synthesizing all of this information gives the problem

$$\begin{aligned} \min_{x_1 \in X_1, \ldots, x_n \in X_N} \sum_k \rho_p(Q_k^{-1/2}(G_k x_k - x_{k-1})) \\ + \rho_m(R_k^{-1/2}(H_k x_k - y_k)), \end{aligned} \tag{6.2}$$

with ρ_p and ρ_m convex penalties. Using (6.2) provides estimates that are robust to outliers and can follow sudden changes in the state. Most of the inference- or optimization-based work in the convex dynamic setting is a special case of (6.2). Many examples, including robust penalties and constraints, are collected in [10].

We now extend to *singular covariances* R_k (for errors v_k) and Q_k (for innovations w_k). These models specify key use cases, particularly for innovations (process) modeling, briefly summaried below (see [76] for a more detailed discussion).

Deterministic integrals. Most models in robotics, particularly in navigation, use integration to model process relationships between state variables (e.g. when position, velocity,

and acceleration are part of the state). Any deterministic integral yields a singular process model. The simplest example (with position a direct integral of stochastic velocity) is used to create Figure 6.1.

Nuisance parameters. Unknown constants that need calibration (such as fixed instrument biases) require special modeling in the nonsingular paradigm (6.2). With singular models, we can augment the state in order to infer these parameters.

Auto-regressive models and correlated errors. State-space models are broadly used in auto-regressive, moving average, and time series models [72]. These elements also appear in general smoothing models, particularly to deal with correlated measurement errors [46].

All three examples are accessible in the classic linear Gaussian setting. The KF need not invert Q or R, and provides the minimum variance estimate for both singular and nonsingular models [9]. Some algorithms rely on precise knowledge of the error structure or explicit equality constraints [90, 87, 2]. Correlated errors are dealt with by augmenting the state and using a singular model [46]. **None of these techniques generalize to singular models in the setting of** (6.2), and some naive generalizations fail dramatically (Figure 6.1).

This paper builds on the reformulation of [76] for singular models. We develop a systematic approach and test it using a navigation model with a real-world mooring dataset. We show how robust statistics, singular models, and constraints can be systematically used to overcome a range of challenges simultaneously present in the dataset: (1) outliers, (2) deterministic relationships between states, (3) measurement biases, and (4) coarsely discretized observations.

The paper proceeds as follows. Section 6.2 summarizes the singular formulation of [76] and relevant optimization algorithms. Section 6.3 develops the key modeling elements to address common data challenges. Section 6.4 presents the navigation models. Section 6.5 shows how the model elements come together to analyze the target mooring dataset, and obtain a high fidelity track from low-fidelity observations. Section 6.6 concludes with discussion and future work.

6.2 Robust Singular Formulation and Algorithm

We reformulate the robust smoothing problem to seamlessly allow both singular and nonsingular covarinace models for errors and innovations. The resulting problem can be solved with any primal-dual algorithm. We show how to apply the classic Douglas-Rachford splitting (DRS) algorithm (see e.g. [52, 49]) to the reformulated problem.

Problem (6.2) can be reformulated by introducing auxiliary variables u_k and t_k to represent pre-whitened innovations and measurement residuals:

$$\begin{aligned} \min_{x,u,t} \quad & \sum_k \rho_p(u_k) + \rho_m(t_k) + \rho_s(x_k) \\ \text{s.t.} \quad & Q_k^{1/2} u_k = G_k x_k - x_{k-1} \\ & R_k^{1/2} t_k = H_k x_k - y_k \end{aligned} \tag{6.3}$$

where $\rho_s(x_k)$ may be taken as the convex indicator function to recover the constraints in (6.2):

$$\rho_s(x_k) = \begin{cases} 0 & x_k \in X_k \\ \infty & x_k \notin X_k. \end{cases}$$

When Q_k and R_k are invertible, we can solve for u_k, t_k and recover (6.2). Otherwise, problem (6.3) is well-posed while (6.2) is not. We can write (6.3) in compact form

$$\begin{aligned} &\min_z \quad \rho(z) \quad \text{s.t. } Az = \hat{w}, \\ &\rho(z) = \sum_{k=1}^{N} \rho_p(u_k) + \rho_m(t_k) + \rho_s(x_k). \end{aligned} \tag{6.4}$$

where

$$\begin{aligned} z^T &= \begin{pmatrix} u_1^T & t_1^T & x_1^T & \dots u_N^T & t_N^T & x_N^T \end{pmatrix} \\ \hat{w}^T &= \begin{pmatrix} x_0^T & y_1^T & 0 & y_2^T & \dots & 0 & y_N^T \end{pmatrix}, \end{aligned} \tag{6.5}$$

$$A = \begin{pmatrix} D_1 & 0 & \dots & 0 \\ B_1 & D_2 & 0 & \vdots \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & B_{N-1} & D_N \end{pmatrix}, \tag{6.6}$$

and

$$D_i = \begin{pmatrix} Q_i^{1/2} & 0 & I \\ 0 & R_i^{1/2} & H_i \end{pmatrix}, B_j = \begin{pmatrix} 0 & 0 & -G_{j+1} \\ 0 & 0 & 0 \end{pmatrix}.$$

The variables are ordered in such a way that A is block bi-diagonal. If all observations z_i lie in the range of H_i, the constraint $Az = \hat{w}$ will be feasible [76].

The problem (6.4) is a convex optimization problem and can be solved using a variety of techniques. We show that the DRS algorithm is straightforward to implement, and preserves the computational complexity of the classic KF/RTS algorithms because of the way A is structured in (6.6).

Given a convex function f, its convex conjugate f^* is given by

$$f^*(y) = \sup_x \langle x, y \rangle - f(x),$$

and its proximal operator with step α, denoted by $\text{prox}_{\alpha f}$ (see e.g. [48]) is given by:

$$\text{prox}_{\alpha f}(\zeta) = \arg\min_x \frac{1}{2\alpha}\|\zeta - x\|^2 + f(x). \tag{6.7}$$

For a long list of objectives, prox operators are available in closed form or are efficiently computable. In particular this is the case when ρ_p, ρ_m, ρ_s form any subset of the numerous elements briefly surveyed in Section 6.3. It is actually the prox of ρ^* that appears in the DRS iteration (Algorithm 2) rather than the prox of ρ, but these are linked by the simple formula

$$\text{prox}_{\rho}(z) + \text{prox}_{\rho^*}(z) = z.$$

To specify the algorithm, we let $g(z)$ be the indicator of the affine feasible region $Az = \hat{w}$:

$$g(z) = \begin{cases} 0 & Az = \hat{w} \\ \infty & Az \neq \hat{w} \end{cases}$$

Problem (6.4) can now be written simply as

$$\min_z \rho(z) + g(z)$$

which is a natural template for DRS, detailed in Algorithm 2.

Algorithm 2 Douglas-Rachford Splitting (DRS)

Require: Initialize at any z^0, ζ^0.

1: **loop**

2: $z^k = \text{prox}_{\tau g}(z^{k-1} - \tau\zeta^{k-1})$

3: $\zeta^k = \text{prox}_{\sigma\rho^*}(\zeta^{k-1} + \sigma(2z^k - z^{k-1}))$

return z^k

To implement Algorithm 2 we need proximal operators of ρ_p, ρ_m, and ρ_s. Eight common piecewise linear-quadratic (PLQ) penalties are shown in Figure 6.2, and their proximal operators are summarized in Table 6.1.

The proximal operator for g is given by

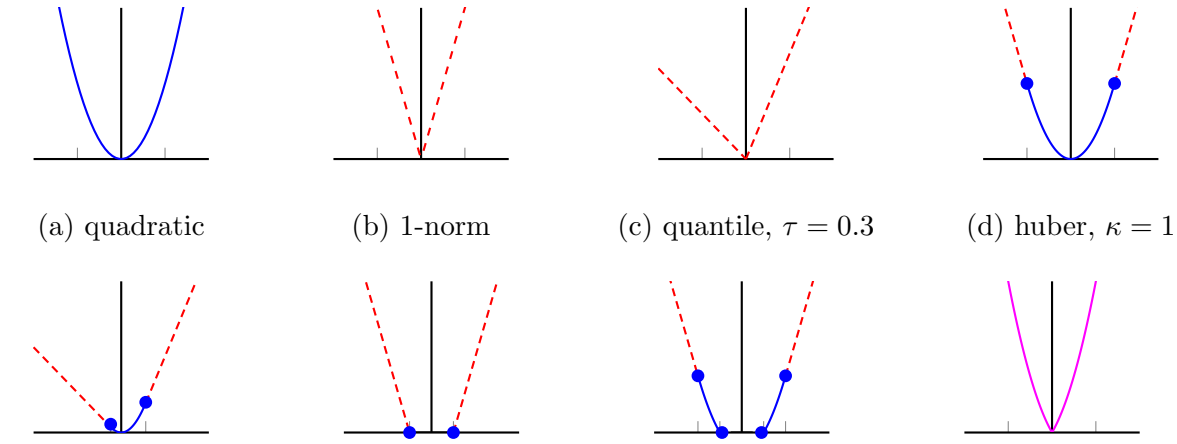
$$\text{prox}_g(\eta) = \underset{Az=\hat{w}}{\arg\min} \frac{1}{2}||\eta - z||^2$$

which is a least squares problem with affine constraints. Solving it efficiently leverages the structure of (6.6). In particular we need to solve a single structured linear system

$$\begin{bmatrix} I & A^T \\ 0 & AA^T \end{bmatrix} \begin{bmatrix} z \\ \nu \end{bmatrix} = \begin{bmatrix} \eta \\ A\eta - \hat{w} \end{bmatrix} \tag{6.8}$$

where AA^T is block tridiagonal and does not change between iterations. In our implementation, we need only compute a single block bidiagonal factorization once, which can then be used to solve (6.8) in $O(n^2N)$ operations in each iteration, no more expensive than a single matrix-vector multiply.

For piecewise-linear quadratic ρ [127, 10], DRS converges to an optimal solution at a local linear rate [76], which does not depend on the condition number of A. A good initialization makes DRS competitive with the fastest available solvers, even second order methods with quadratic local rates [10].



(a) quadratic (b) 1-norm (c) quantile, $\tau = 0.3$ (d) huber, $\kappa = 1$

(e) quantile huber (f) vapnik, $\epsilon = 0.5$ (g) hubnick (h) elastic net

Figure 6.2: Common piecewise linear-quadratic (PLQ) losses.

6.3 Modeling Elements

The proposed framework has three complementary modeling elements: singular covariance matrices Q and R; process/measurement penalties ρ_p, ρ_m; and constraints ρ_s on the state. In this section, we show a range of choices for each element, and compute the operators required for Algorithm 2.

Singular covariances can be used to capture affine constraints, auto-regressive structure, integrated errors, and bias.

- *Affine constraints using singular R.* the ith element of the state at time k is known exactly, add row

$$\begin{bmatrix} 0 & \dots & 0 & \underbrace{1}_{i} & 0 & \dots & 0 \end{bmatrix}$$

to the measurement model H_k, a row and column of zeros to R_k, and the known value as the last element of z_k.

- *Bias with singular Q.* A common model for bias is to include it as a non-varying

component across the state:

$$\widetilde{x}_k = \begin{bmatrix} x_k \\ b \end{bmatrix}, \quad \widetilde{Q}_k = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}.$$

- *Correlated noise using singular Q.* Correlated noise w_k is typically modeled by [45]

$$w_k = Mw_{k-1} + \beta_k, \quad \beta_k \sim N(0, Q).$$

Here too we can augment the state and use a singular process variance:

$$\widetilde{x}_k = \begin{bmatrix} x_k \\ w_k \end{bmatrix}, \widetilde{G}_k = \begin{bmatrix} G_k & I \\ 0 & M \end{bmatrix}, \quad \widetilde{Q}_k = \begin{bmatrix} 0 & 0 \\ 0 & Q_k \end{bmatrix}.$$

Piecewise linear-quadratic (PLQ) Penalties. The proposed framework allows process innovations, measurement residuals, and state regularization to come from any convex prox-friendly penalty. To keep the exposition simple, we collect eight commonly used convex piecewise linear-quadratic penalties in Figure 6.2, and compute their prox operators in Table 6.1. The penalties can be thought of in terms of three features:

- Behavior at origin: nonsmooth features encourage exact fitting of the quantity being measured, while deadzones are appropriate for discretized observations.

- Tail growth: asymptotically linear penalties are more tolerant of large inputs. Applied to measurements, this gives robustness to outliers; applied to innovations, it gives an ability to quickly track evolving trends.

- Asymmetry: allows handling of special cases where under-estimating is qualitatively different from over-estimating.

Constraints. It is very convenient to enforce simple constraints on the state estimates x_k. If we take $\rho_s(x) = \delta_X(x)$ then the prox operator prox_{ρ_s} is simply the projection onto the set X. Box constraints are a very common type of constraints that enforce known bounds on the state, and have a trivial projection. The proposed framework allows us to use any convex region that has a computationally efficient projection.

Table 6.1: Prox operators of common PLQ penalties.

Penalty f	$\mathbf{prox}_{\alpha f}(z)$	**Ref.**
$\frac{1}{2}\|x\|^2$, Fig. 6.2a	$\frac{1}{1+\alpha}z$	[64, 136]
$\|x\|_1$, Fig. 6.2b	$\text{sign}(z) \odot (\|z\| - \alpha)_+$	[70, 53]
q_τ, Fig. 6.2c	$\begin{cases} z_i - \alpha(1-\tau) & z_i > \alpha(1-\tau) \\ z_i + \alpha\tau & z_i < -\alpha\tau \\ 0 & \text{else} \end{cases}$	[88, 89]
h_κ, Fig. 6.2d	$\frac{\alpha}{\alpha+\kappa}z + \frac{\kappa}{\alpha+\kappa}\text{prox}_{(\alpha+\kappa)\|\cdot\|_1}(z)$	[103]
$q_{\tau,\kappa}$, Fig. 6.2e	$\frac{\alpha}{\alpha+\kappa}z + \frac{\kappa}{\alpha+\kappa}\text{prox}_{(\alpha+\kappa)q_\tau}(z)$	[1]
v_ϵ, Fig. 6.2f	$\begin{cases} z_i - \alpha & z_i > \epsilon + \alpha \\ \epsilon & \epsilon < z_i \leq \alpha + \epsilon \\ z_i & -\epsilon \leq z_i \leq \epsilon \\ -\epsilon & -\epsilon - \alpha < z_i \leq -\epsilon \\ z_i + \alpha\tau & z_i < -\alpha - \epsilon. \end{cases}$	[152]
hubnik-κ, Fig. 6.2g	$\frac{\alpha}{\alpha+\kappa}z + \frac{\kappa}{\alpha+\kappa}\text{prox}_{(\alpha+\kappa)v_\epsilon}(z)$	[44, 93]
e-net, Fig. 6.2h	$\text{prox}_{\frac{\alpha}{1+2\alpha}\|\cdot\|_1}\left(\frac{1}{1+2\alpha}z\right)$	[164]

6.4 Navigation Model

We use a constant-velocity kinematic model that is appropriate for many underwater vehicle applications, where accelerations are heavily damped and trajectories are often long straight lines (e.g. for transit or survey work). When the attitude is known or changing slowly, the model can be linearized effectively. For a vehicle that is well-instrumented in attitude, the uncertainty in position (and the x-y states in particular) is typically orders of magnitude larger than the uncertainty in attitude. Thus, in practice, we often simplify the full nonlinear vehicle process model to track only position states (x, y, z), while assuming that the attitude states (r, p, h) are directly available from the most recent sensor measurements. To make the model linear, the position and its derivatives are referenced to the local-level frame.

An effective model must counteract biases, outliers, and data discretization in the IMU data. We develop this model using the elements of the proposed framework.

Process model. To incorporate linear acceleration measurements from an IMU, we track linear velocities and linear acceleration in the state vector:

$$x_s = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}]^\top. \tag{6.9}$$

The linear kinematic process model is given by

$$\dot{x}_s = \underbrace{\begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix}}_{F_s} x_s + \underbrace{\begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix}}_{G_s} w_s, \tag{6.10}$$

where $w_s \sim \mathcal{N}(0, Q_s)$ is zero-mean Gaussian noise. The linear process model (6.10) is discretized using a Taylor series:

$$x_{s_{k+1}} = F_{s_k} x_{s_k} + w_{s_k} \tag{6.11}$$

$$F_{s_k} = e^{F_s T} \approx\approx \begin{bmatrix} I & IT & \frac{1}{2}IT^2 \\ 0 & I & IT \\ 0 & 0 & I \end{bmatrix},$$

where I in (6.11) denotes the 3×3 identity matrix, and the higher order terms are identically zero because of the structure of F_s. We model the process covariance as if the error were the next term in the Taylor series approximation, a technique suggested by [22]. More precisely we set covariance to be the outer product, $\Gamma^T \Gamma$ where

$$\Gamma = \begin{bmatrix} \frac{1}{3!} IT^3 & \frac{1}{2!} IT^2 & IT \end{bmatrix}$$

This leads to a rank 3 covariance for a 9×9 matrix for a model that comprises position, velocity, and acceleration.

Given this covariance structure, the process model will penalize changes in acceleration. As the vehicle travels in a relatively straight line with small corrections, we expect to see acceleration mostly constant with a few small jumps. We use the ℓ_1 norm for innovations, as it encourages exact fits while simultaneously allowing occasional sudden changes.

Measurement model. The inertial measurement unit (IMU) measures linear and angular accelerations relative to the physical frame of the vehicle on which it is mounted, while the state tracks linear acceleration relative to the navigation frame. We obtain the coordinate transformation between these frames using heading, pitch, and roll of the vehicle:

$$R(\varphi) = R_h^\top R_p^\top R_r^\top, \tag{6.12}$$

where R_h, R_p, and R_r are given by

$$\begin{bmatrix} ch & sh & 0 \\ -sh & ch & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} cp & 0 & -sp \\ 0 & 1 & 0 \\ sp & 0 & cp \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & cr & sr \\ 0 & -sr & cr \end{bmatrix} \tag{6.13}$$

with $c\cdot$ and $s\cdot$ shorthand for $\cos(\cdot)$ and $\sin(\cdot)$.

Position data from the USBL is sampled at a lower update rate than the IMU. For any s_k where such position data is available, we have the measurement model

$$H_{s_k} = \begin{bmatrix} I_{3\times 3} & 0_{3\times 6} \\ 0_{3\times 6} & R(\varphi) \end{bmatrix}, \quad z_{s_k} = \begin{bmatrix} \text{usbl}^\top & \ddot{x}_{\text{meas}} & \ddot{y}_{\text{meas}} & \ddot{z}_{\text{meas}} \end{bmatrix}^\top$$

If there is no position data measured at time s then we use the model

$$H_{s_k} = \begin{bmatrix} 0_{3\times3} & 0_{3\times6} \\ 0_{3\times6} & R(\varphi) \end{bmatrix}, \quad z_{s_k} = \begin{bmatrix} 0 & \ddot{x}_{\text{meas}} & \ddot{y}_{\text{meas}} & \ddot{z}_{\text{meas}} \end{bmatrix}^\top.$$

The covariance used for measurement data similarly depends on whether there is position data available:

$$R_{s_k} = \begin{bmatrix} U_s & 0_{3\times3} \\ 0_{3\times3} & r_s I_{3\times3} \end{bmatrix}, \quad R_{s_k} = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & r_s I_{3\times3} \end{bmatrix}$$

where U is a diagonal matrix reflecting position uncertainty, while r_s captures uncertainty in IMU measurements.

Bias model. To compensate for the bias in acceleration data we augment the state vector to include bias variables:

$$\bar{x}_s = [x_s^T, b_1, b_2, b_3]^\top$$

where b_1, b_2, b_3 are bias terms for acceleration in the x, y, z directions in the local frame. To pass the bias estimates forward in time the process matrix is augmented with an identity block.

$$\bar{x}_{s_{k+1}} = \bar{F}_{s_k} x_{s_k} + \bar{w}_{s_k} \tag{6.14}$$

$$\bar{F}_{s_k} = \begin{bmatrix} F_{s_k} & 0 \\ 0 & I \end{bmatrix} \tag{6.15}$$

At the first time point ($s_k = 1$) we augment the covariance matrix with an identity block, and at all other time points we augment with a zero block. This adds equality constraints for the bias terms over all time points. The approach generalized easily to model piecewise-constant biases over longer periods.

$$\bar{Q}_1 = \begin{bmatrix} \Gamma^T\Gamma & 0 \\ 0 & I \end{bmatrix}, \quad \bar{Q}_{s_k} = \begin{bmatrix} \Gamma^T\Gamma & 0 \\ 0 & 0 \end{bmatrix} \quad s_k > 1.$$

The measurement matrices are augmented with an identity block that shifts the acceleration measurements using the bias.

Discretization model. The measurement loss function is chosen to account for the level of discretization present in the data. The Vapnik loss (Figure 6.2f) has a 'deadzone' around the origin where small discrepancies are not penalized. This region is set using the level of discretization in the data, in this case 0.05. The sharp corners of the Vapnik loss encourage errors to lie exactly in them, an unnecessary artifact. Thus we use the 'Huberized' version dubbed 'hubnik' (Figure 6.2 g). The prox operators for both losses are computed in Table 6.1.

6.5 Analysis of Mooring Data

We are interested in the ability to maintain an accurate position estimate on-board an autonomous underwater vehicle in real-time using acceleration measurements from a low-cost IMU, given periodic position fixes.

To test this, we use the singular robust Kalman framework to analyze data collected from a surface mooring equipped with an IMU on the subsea node. The mooring node, which is drifting with the current, is used as a proxy for a slowly moving underwater vehicle subject to unknown disturbances. We look at the position uncertainty and error accrued over time between the periodic, world-referenced position fixes provided by the USBL system.

Data description. Position fixes are available from a ship-based Sonardyne Ranger 2 USBL every 2 seconds, which we subsample to varying degrees for the analysis. Linear acceleration data from an LSM303D 3-axis accelerometer was collected at ~ 0.075 m/s^2 precision using a Raspberry Pi Zero.

A small snippet of vertical acceleration data, Figure 6.3, shows the relatively coarse discretization of the measurements along with a mean that is shifted away from zero indicating bias. Outliers also appear likely. Because the IMU was generally upright with small perturbations in pitch and roll, the discretizations in the z-axis of the instrument frame are still clearly visible when the data are rotated into the vertical world frame. The small perturbations in pitch and roll cause the slight variations visible within each discretization level.

To illustrate our model's usefulness on a data set such as in Figure 6.3, we isolate a 25-

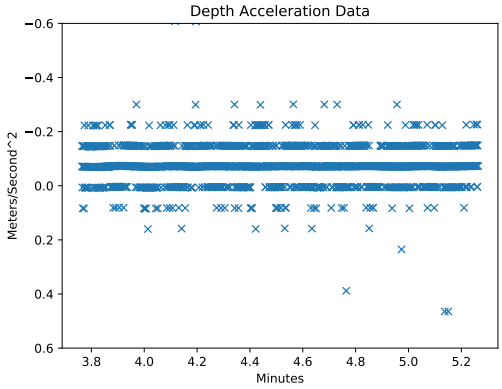


Figure 6.3: A snippet of the depth acceleration data, rotated into the world frame, shows the discretization and bias of the acceleration data.

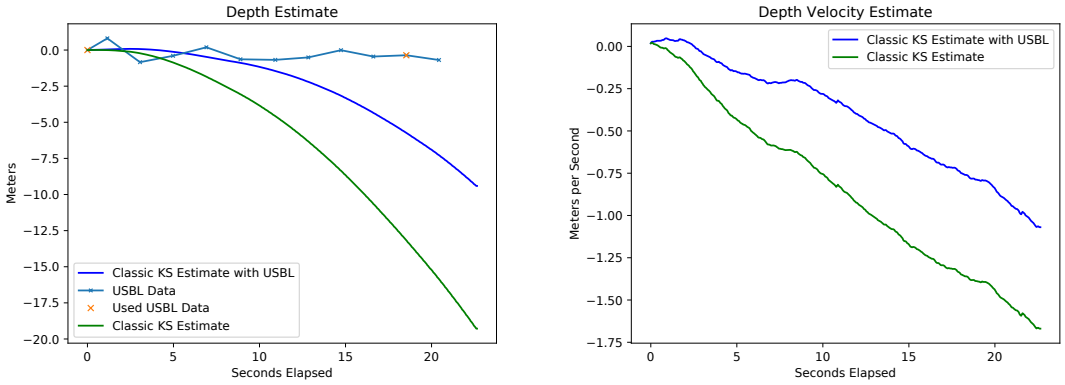


Figure 6.4: Comparison of classic Kalman smoothing applied to depth data with and without USBL position fixes.

second window of the data and add modeling elements one at a time, noting the improvements they provide. We finish by running the full model with varying amounts of infrequent position data on a 10-minute section of data as a more practical experiment.

Model elements applied to data. We consider 25 seconds of IMU data and apply a

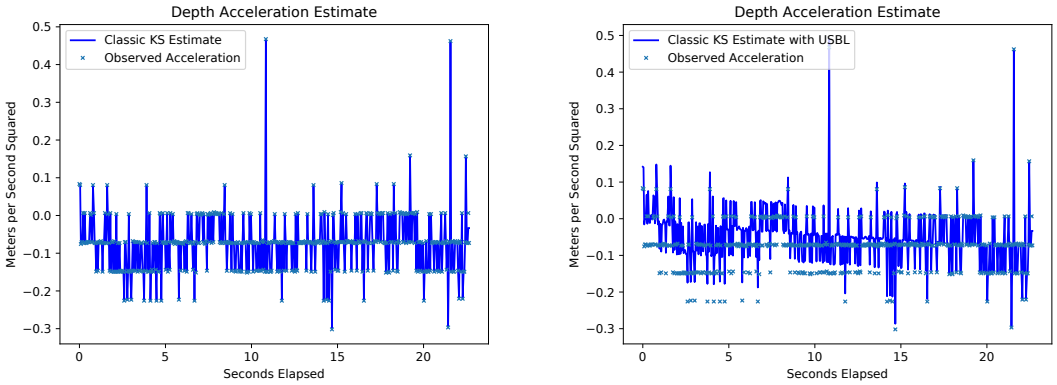


Figure 6.5: Acceleration estimates of classic Kalman smoother without USBL data (left) and with USBL data (right).

classic Kalman filter (using least squares measurement and process loss) with the singular navigation model detailed in section 6.4. In practice, one would expect an underwater vehicle to be well instrumented in depth, but for illustration purposes we focus on depth and the vertical acceleration measurements. These measurements are the most biased and therefore improvements made by adding modeling elements are most clearly shown.

The 25 seconds of IMU data starts at a USBL measurement. Position is initialized to this starting USBL measurement, but for ease of comparison that position is treated as (0,0,0) and all subsequent USBL measurements are treated as relative offsets. DRS works better when its reasonably initialized. The initial state vector over the smoothing window is populated by setting acceleration to 0, and then propagating forward the most recent position fix with a significantly damped measurement of the most recent velocity to prevent divergence of the initial vector.

We start by adding bias estimation (for the acceleration measurements). This requires a second USBL position fix, 19 seconds into the 25-second data series. Figure 6.4 shows position and velocity estimates from a classic Kalman smoother applied to depth acceleration data with and without additional USBL position data. As expected due to the quality of the data

we see poor performance in both cases although the small amount of additional position data offers slight improvement.

Figure 6.5 shows acceleration estimates of the classic Kalman smoother with and without additional USBL data. The two USBL measurements affect acceleration estimates locally but are generally overpowered by the vastly more complete set of (biased) acceleration data.

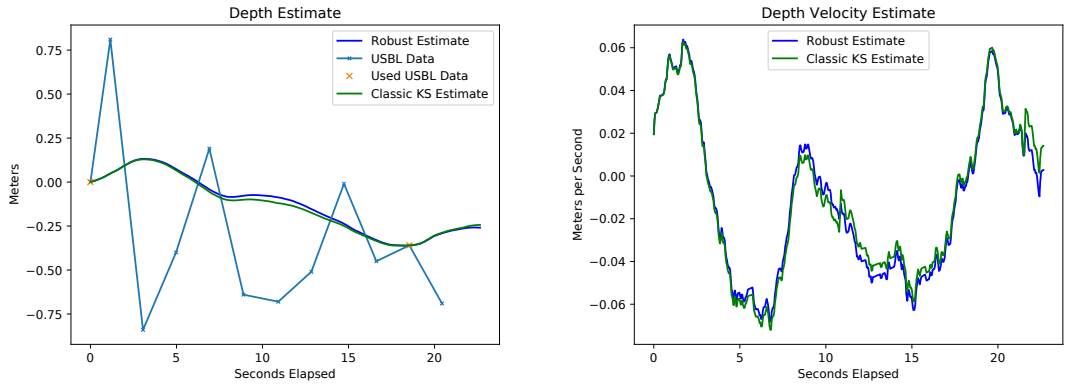


Figure 6.6: Comparison (position and velocity estimates) of debiasing least squares Kalman smoother vs. a robust debiasing smoother equipped with the hubnik loss (Figure 6.2g).

The picture improves significantly when we add in bias estimation (via singular process models), using the same two observations (compare Figures 6.4 and 6.6). With the bias removed, we can now focus removing the noticeable outliers form the acceleration data. For this purpose, we use the robust hubnik (Figure 6.2g) as our measurement loss. The 'deadzone' is designed to work with the coarse discretization of the acceleration data.

Figure 6.6 compares the results of (debiased) fitting between the least squares Kalman smoother and the robust version using the hubnik loss. Figure 6.7 shows the acceleration estimates with bias removed in both the classic and robust setting. There are noticeable differences in the acceleration estimates at around 11 and 22 seconds. When using the robust smoother, the effect of the outliers on the model's estimate is greatly reduced (see the

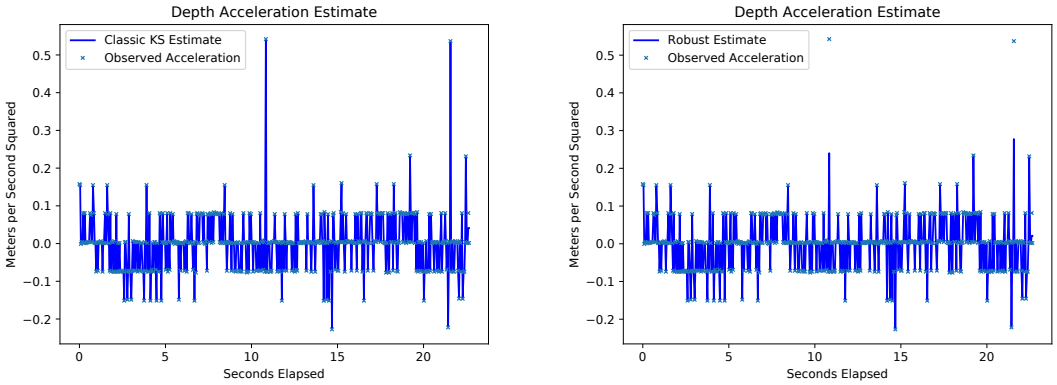


Figure 6.7: Acceleration estimates for least squares debiasing Kalman smoother (left) vs. robust debiasing smoother equipped with the smoothed Vapnik loss (Figure 6.2g) (right).

velocity estimates at 11 and 22 seconds). In the context of real-time tracking or forecasting, these sudden jumps will yield inaccurate predictions.

10 minutes of data. We apply the robust smoother with with bias, outliers, and discretization modeling elements to 10 minutes of IMU data. USBL data is available approximately every 2 seconds, but we test performance of the smoothing algorithm at larger gaps, with USBL data supplied at $30, 60$, and 120 seconds.

Figure 6.8 has the fitted position plots for all three frequencies. We see that without any USBL fix data, the estimate suffers; but even infrequent fixes give significant improvements when using the full capability of the smoother. In fact there are diminishing returns in increasing the USBL frequency; this is a promising result towards the future goal of a practical online implementation, particularly in settings where high-quality USBL observations are unavailable (e.g. during dives).

Figure 6.9 shows the fitted velocity. Here the effect of additional USBL data are more apparent, as velocity is completely inferred from position and acceleration. However, the smoothing estimates of velocity at infrequent USBL fixes are still very good compared to those informed by frequent USBL fixes.

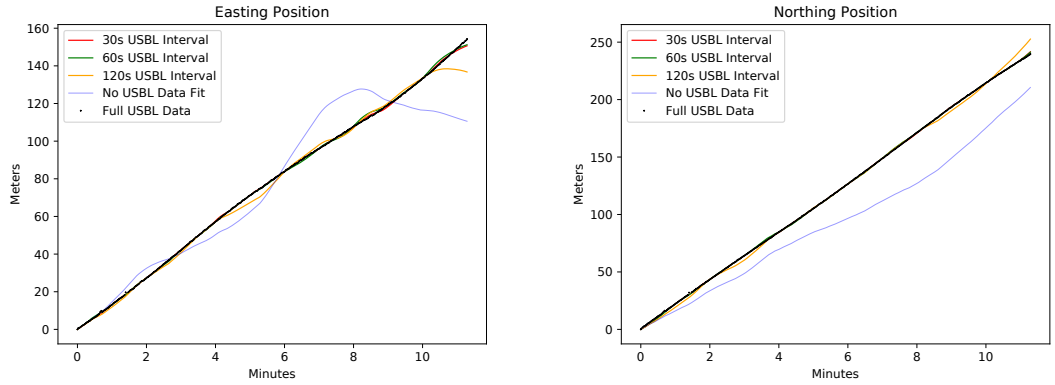


Figure 6.8: Position estimates obtained with robust debiasing smoother for three frequencies of USBL fixes. Robust smoothing allows reasonable tracking from infrequent USBL observations.

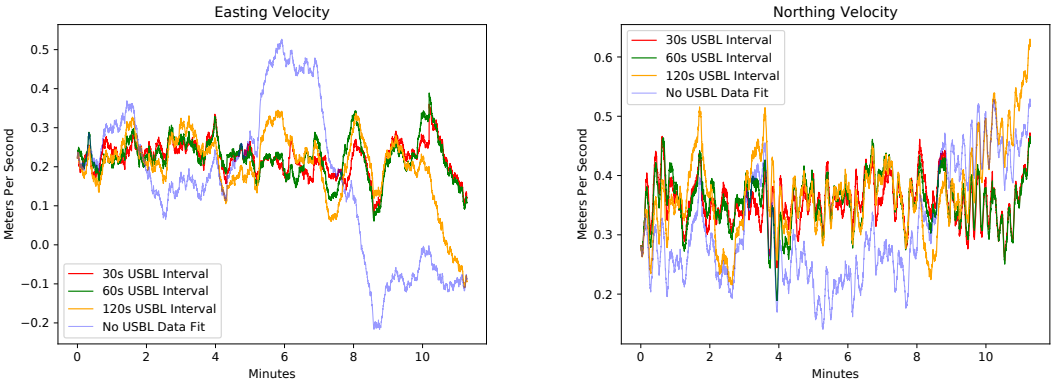


Figure 6.9: Velocity estimates obtained with robust debiasing smoother for different frequencies of position data. Errors from acceleration measurements build up without USBL fixes, but infrequent USBL measurements still allow velocity estimation.

6.6 Discussion and Future Work

We have proposed a singular Kalman smoothing framework that can use singular covariance models for process and measurements, convex robust losses, and state-space constraints. The modeler can use any convex loss that has an implementable prox, a class that includes the most common choices used for inference in tracking and navigation. The framework offers a wide range of flexibility that can be used to either counteract undesirable characteristics present in data or be used to increase model performance based off of relevant field knowledge. Future work will consider real-time implementation, as well as extension to nonlinear models.

Numerical experiments show that these modeling elements yield significant improvement on a noisy, challenging dataset. We also see that having a robust model makes the smoother less reliant on frequent high-quality position updates, which is a very promising development for underwater navigation.

This paper develops several tools required to move to robust singular tracking in real-time. A promising aspect of singular noise models is that they make it possible to do simple robust windowed smoothing, where estimates are constrained between windows as the tracking proceeds. Constraints on the state may play a bigger role in real-time estimation, since they can help detect outliers faster. Finally, robust penalties that provide better estimates may further improve performance of the DRS algorithm, by providing an effective initialization for each new window. We will focus on these developments in future work.

Acknowledgements

This material is based upon work supported by the Defense Advanced Research Agency (DARPA) and Naval Information Warfare Center (NIWC) Pacific under Contract No. N66001-16-C-4001. It is approved for public release, distribution unlimited. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Chapter 7

PRELIMINARY RESULTS IN CURRENT PROFILE ESTIMATION AND DOPPLER-AIDED NAVIGATION FOR AUTONOMOUS UNDERWATER GLIDERS

Here we consider estimating additional unknown variables some of which are coupled with the states. Using the MAP approach these variables can simply be added to the optimization problem with the downside being that the traditional Kalman smoothing structure is broken.

7.1 Introduction

The goal of this work is to simultaneously estimate a) absolute (Earth-referenced) ocean velocity profile, and b) the absolute autonomous underwater glider (AUG) path over the bottom, based on on-board acoustic Doppler current profiler (ADCP) observations of relative current velocity profiles. Figure 7.1 shows two Seagliders with upward-facing ADCPs mounted in the aft fairing.

The main challenge of AUG-based ADCP velocity profiling is that each velocity profile is measured relative to the through-the-water (TTW) motion of the glider, as opposed to the earth referenced glider velocity. The AUG's TTW velocity can be inferred from the AUG dynamic model, but is subject to uncertainty because the model is based on steady state flight and doesn't take roll into account. This lack of a georeferenced platform velocity requires a simultaneous estimation of current and glider TTW velocity using all available data—ADCP velocity profiles, surrounding water density, glider buoyancy engine state, glider attitude and angle of attack, glider depth, and GPS positions at the start and end of the dive.

Here we describe two different frameworks for performing this estimation—a linear global inverse, and a flexible state space approach. In this initial work, we use a pre-computed es-

Figure 7.1: Ready for launch, Seagliders SG196 and SG198 are loaded on the R/V Ukpik in Prudhoe Bay, AK, with the upward-facing ADCPs visible where they are installed in the aft fairing.

timate of the glider TTW velocity from the hydrodynamic model for both methods. The overarching future goal is to use the flexibility of the state-space model to include the nonlinear hydrodynamic model and nonlinear range measurements as part of the approach.

7.2 Background

Historically, ADCP measurements were made from ships with relatively low frequency systems (70 kHz) or from moorings with low or mid-frequency instruments (300 kHz). The 300 kHz ADCPs are also common on large underwater vehicles, in particular instruments that can be used as a Doppler velocity log to provide "bottom lock"—i.e., vehicle velocity relative to the seafloor [163] or, in some specialized cases, the underside of ice [105].

To provide better resolution of deep currents from shipboard measurements, lowered ADCP methods were developed using overlapping shear traces from a higher frequency (typ-

ically 150 or 300 kHz) ADCP lowered from a (relatively) stationary ship. The shear traces are then used to reconstruct the full current profile. The two main lowered ADCP processing frameworks are the shear method, originally described by [61, 62], and the velocity inversion method [154].

In the last 10 years, there has been increased interest in using ADCPs from autonomous underwater vehicles to characterize the currents between the surface and the seafloor and to close the gap of uncertainty in vehicle drift between when GPS is available at the surface and bottom lock is obtained within range of the seafloor [139, 140, 106].

Aided by the development of very high frequency ADCPs (1 MHz) specifically for autonomous underwater gliders, the two main lowered ADCP processing frameworks—the shear method and the inversion method—have been modified to estimate depth-varying current profiles from gliders by [147] and [148, 149] respectively. A related, but distinctly different method based on non-linear objective mapping of ADCP shear has been also developed by [50]. In all implementations ([147, 149, 50]), the authors needed to contend with the limitation of Slocum and Spray glider ADCPs that only collect data during one half of the dive (either ascending or descending).

Our contribution builds on and extends the velocity inversion method with several key innovations. We use current data from the entire dive (as opposed to just during descent), which is crucial for multi-hour dives where, as we show, the current profiles on descent versus ascent differ significantly. We then explicitly separate the over-the-ground glider velocity into the drift velocity (as a result of advection), and the glider's horizontal through-the-water velocity. This enables us to directly incorporate hydrodynamic model velocity estimates and independently control the smoothness regularization of the different velocity components.

7.3 Finding the Current Profile by Inversion

In this section we develop an approach to use (1) on-board ADCP observations of relative current velocity profiles, (2) dive start and end GPS coordinates and (3) a measure of through the water (TTW) velocity obtained from the glider to simultaneously estimate (a) absolute

(Earth-referenced) ocean velocity profile, and (b) the glider path through the water. We present the derivation for a scalar velocity v, which can be understood to be either one of the velocity components (u, v), or a complex-number representation of velocity, $u + iv$.

7.3.1 Formulation of the inverse problem

Similar to [154, 148], glider-borne ADCP observations of horizontal currents are considered to be a sum of three unknown components:

$$u^a(z,t) = u^o(z) - u^g(t) + \epsilon(z,t),$$

where

- u^o is the absolute ocean velocity
- u^g is the over-the-ground (OTG) velocity of the glider platform
- ϵ is the ADCP measurement noise.

Here, we separate the OTG glider velocity into the drift velocity $u^d = u^o(z^g(t))$, equal to the ocean velocity at the glider depth z^g, and the glider horizontal TTW "propulsion" speed u^p, which is determined by the glider control and its flight dynamics:

$$u^g(t) = u^o(z^g(t)) + u^p(t).$$

Thus the inverse problem is to obtain $u^o(z)$ and $u^g(t)$ such that the discrepancy with the observed values of u^a are minimized,

$$\min \|u^a(z,t) - (u^o(z) - u^o(z^g(t)) - u^p(t))\|_2^2. \tag{7.1}$$

This formulation uses the least squares loss, corresponding to a Gaussian assumption on the noise ϵ. More robust losses are of interest and will be studied in future work.

This formulation is more complex than that of [148], as it requires interpolation of ocean velocity onto the glider location $(u^o(z^g(t)))$, but it allows explicit treatment of the glider state. Additionally, applying smoothness regularization to u^p as discussed below is more appropriate than to u^g.

7.3.2 Discrete formulation

We consider a set of individual ADCP observations $u^a_{ij} = u^a(z_{ij}, t_j)$ at depth cells z_{ij} and times t_j, with the $i \in [1, I]$ being the ADCP range cell index, and $j \in [1, J]$ being the sample time (ping) index. All the valid observations form an observation column vector

$$u^a = [u^a_1 ... u^a_k]^\top,$$

where the index $k \in [1, K], K \leq IJ$ enumerates valid observations. The vectors $[z_k]$ and $[t_k]$ represent the corresponding depth and time coordinates of the k-th valid observation.

The unknown state vector

$$x = \begin{bmatrix} u^g \\ u^o \end{bmatrix}$$

consists of a vector of the unknown glider OTG velocities

$$u^g = [u^g_1 ... u^g_M]^\top$$

defined on a temporal grid $\hat{t}_m, m \in [1, M]$, and a vector of the unknown ocean velocities

$$u^o = [u^o_1 ... u^o_L]^\top$$

defined on a regular vertical grid $\hat{z}_l = (l-1)\Delta\hat{z}, l \in [1, L]$. For convenience, the temporal grid $\hat{t}_m$ is taken to be a superset of the sample times t_j, with the extra intervals added during the gaps in the ADCP record (at the beginning and the end of the dive, as well as during a brief intermission at the bottom of the dive); the spacing of the extra intervals is set to the average ADCP sampling interval. Discrete formulation of the ADCP sampling problem (7.1) then becomes

$$u^a = -H^t u^g + H^z u^0 + \epsilon,$$

where ϵ is the noise vector. The matrix operator H^t represents temporal interpolation from the time grid $\{\hat{t}_m\}$ onto the sampling times $\{t_k\}$, and H^z represents spatial interpolation from the vertical grid $\{\hat{z}_l\}$ onto $\{z_k\}$. Since $\{t_k\} \subset \{\hat{t}_m\}$ by construction, H^t is simply a

subsampling matrix,

$$H^t_{km} = \begin{cases} 1, & \text{if } t_k = \hat{t}_m \\ 0, & \text{otherwise} \end{cases}.$$

The formulation of vertical interpolation matrices depends on the chosen interpolation method. Visbeck [154] method is equivalent to the nearest-neighbor interpolation. Here, we employ linear interpolation, corresponding to a matrix operator

$$H^z_{kl} = \begin{cases} 1 - (\hat{z}_l - z_k)/\Delta\hat{z}, & \hat{z}_{l-1} \le z_k < \hat{z}_l \\ 1 - (z_k - \hat{z}_l)/\Delta\hat{z}, & \hat{z}_l \le z_k < \hat{z}_{l+1} \\ 0, & \text{otherwise.} \end{cases}$$

7.3.3 Additional Regularization

The inversion framework balances physical knowledge of the glider's motion (such as start- and end-of-dive GPS fixes) against assumptions about the physical environment, including smoothness of current profiles in time and space.

Start- and End-of-Dive GPS: GPS fixes before and after the dive provide tie-points on the integral $\int u^g dt$ over the duration of the dive. The discrete formulation of these relationships is given by

$$Sx \approx s,$$

where s is the scalar (complex) horizontal displacement between the two GPS fixes, $S = \begin{bmatrix} w & 0 \end{bmatrix}$, and w represents the weights corresponding to trapezoidal rule integration,

$$w_m = \begin{cases} 0.5(\hat{t}_2 - \hat{t}_1), & m = 1 \\ 0.5(\hat{t}_{m+1} - \hat{t}_{m-1}), & 1 < m < M \\ 0.5(\hat{t}_M - \hat{t}_{M-1}), & m = M \end{cases}$$

Glider Velocity: A measure of the glider's TTW velocity, such as that provided by the glider hydrodynamic model, can be used to provide a constraint on u^p(t) in (7.1). Discrete

formulation of this constraint is

$$u^p = -H^t u^g + H^{z0} u^o + \epsilon,$$

where u^p is the TTW velocity predicted by the hydrodynamic model, and H^{z0} represents spatial interpolation of ocean velocity profile onto the glider depth (so that $H^{z0}x$ is the glider drift speed); this matrix is constructed in the same way as H^z. The relative confidence, within the model, of these measurements compared to the ADCP measurements can be controlled by adjusting the relative magnitude of the noise vectors ϵ for the two measurements. As is demonstrated and discussed in the Results section, however, understanding the subtleties of how to weight the constraints is still a work in progress.

Smoothness Regularization: Additionally, we require both the ocean and the glider velocities to be smooth, which is equivalent to minimization of $D_2 u^g$, and $D_2 u^o$ where D_2 are the second derivative operators of the appropriate sizes,

$$D_2 = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & & \\ \vdots & & & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 & -2 & 1 \end{bmatrix}.$$

Subtleties of Regularization: Ideally, we want to solve an inverse problem that is aware of both the vehicle dynamics and control and the ADCP observations, because this is the only way to make a self-consistent estimate. The challenge, as mentioned above with the glider velocity estimates in particular, is that measurements are inherently noisy, often biased, and optimally choosing the relative weighting between measurements is still a work in progress.

In the extreme case, if we had a perfect hydrodynamic model, we would not need the inverse at all, as we would know the TTW velocity at any moment (based solely on the buoyancy and pitch control), and therefore would have absolute ocean velocity estimates from every trace. At the other extreme, if the ADCP measurements were perfect and ex-

tended close to the vehicle, we would have a direct measure of the TTW velocity from that information alone and would not need the dynamic model.

In reality, the ADCP is noisy and has gaps (notably, the blanking distance). And without the hydrodynamic model, we can't distinguish between the legitimate accelerations due to AUG control (which should be kept) and spurious accelerations arising from ADCP noise (which should be eliminated)—and we would be forced to either keep both, or eliminate both, which is suboptimal.

7.3.4 Least-Squares Formulation

The final least-squares problem formulation is given by

$$\min_x \|Gx - d\|_2^2,$$

where

$$G = \begin{bmatrix} -H^t & H^z \\ w & 0 \\ -H^t & H^{z0} \\ r_o D_2 & 0 \\ 0 & r_g D_2 \end{bmatrix}, \qquad d = \begin{bmatrix} u^a \\ s \\ u^p \\ 0 \\ 0 \end{bmatrix},$$

and r_o and r_g are regularization parameters. The solution is given by

$$x = (G^\top G)^{-1} G^\top d,$$

and can be computed in a more efficient manner (e.g., using a QR decomposition aware of the block structure of G).

7.3.5 Modification: two-profile solution

The ocean velocity profile is expected to change over the duration of the dive. Therefore, it may be reasonable to seek *two* ocean velocity profiles u^d and u^u, corresponding to the

descent and asent. The state vector and the observation matrix then become

$$x = \begin{bmatrix} u^g \\ u^d \\ u^u \end{bmatrix},$$

$$H = \begin{bmatrix} \; -H^t \; & | & \; H^{zd} \; & | & \; H^{zu} \; \end{bmatrix},$$

where the interpolation matrices H^{zd} and H^{zu} are constructed as H^z before, except that only those rows of H^{zd} that correspond to the downcast z_k are non-zero, and vise versa.

An additional constraint is necessary, requiring the ocean velocity profiles to match at the bottom end, i.e.

$$\begin{bmatrix} 0 & e_n^\top & -e_n^\top \end{bmatrix} x = 0.$$

where e_n is the elementary vector with 1 in the last entry.

The modified system is given by

$$\min_x \|Gx - d\|_2^2,$$

where

$$G = \begin{bmatrix} -H^t & H^{zd} & H^{zu} \\ w & 0 & 0 \\ -H^t & H^{z0d} & H^{z0u} \\ r_o D_2 & 0 & 0 \\ 0 & r_g D_2 & r_g D_2 \end{bmatrix}, \qquad d = \begin{bmatrix} u^a \\ s \\ u^p \\ 0 \\ 0 \end{bmatrix},$$

For shorter dives, assuming identical current profiles on ascent and descent may be a reasonable assumptions. For the data collected over multi-hour dives, however, as we show, there can be significant deviation in the current profile during descent and ascent, and we use the two-profile solution.

7.4 Deconvolving Glider State and Current Profile: State-Space Approach

In this section, we propose a state-space model that fuses available information (ADCP current profiles, GPS coordinates, and measure of smoother velocities) to simultaneously infer the states of the glider together with current profile estimation. The formulation uses identical information to that used in Section 7.3. However, the main innovation here is to use a more detailed navigation model, with a view toward simultaneously solving the current mapping and glider localization problems. A direct consequence of this is an updated estimate of the state variables of the glider, along with the estimates of the current profile.

7.4.1 Glider State-Space Model

We start by constructing a linear four-dimensional state-space model for the glider positions and their derivatives (OTG velocity):

$$\begin{aligned} x_k &:= \begin{bmatrix} e_k^g & n_k^g & \dot{e}_k^g & \dot{n}_k^g \end{bmatrix}^T, \\ x_{k+1} &= G_k x_k + \epsilon_k, \quad \epsilon_k \sim \mathcal{N}(0, Q_k) \\ z_k &= H_k x_k + \nu_k, \quad \nu_k \sim \mathcal{N}(0, R_k), \quad \text{for } k = 1, N \\ G_k &= \begin{bmatrix} 1 & 0 & \Delta t_k & 0 \\ 0 & 1 & 0 & \Delta t_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{aligned} \tag{7.2}$$

The four states are east/north over-the-ground (OTG) velocities and their approximate integrals. The only measurements used here are the GPS position fixes at the beginning and end of the dive. A separate term is added later to compare OTG velocities with (measured) TTW velocities.

The state-space model (7.2) does not force smoothness of the velocities, unlike the previous section. The full-state block bi-diagonal process-discrepancy matrix G, and block

diagonal matrices H, Q, R are

$$G = \begin{bmatrix} I & 0 & & \\ -G_2 & I & \ddots & \\ & \ddots & \ddots & 0 \\ & & -G_N & I \end{bmatrix}, \quad H = \begin{bmatrix} H_1 & 0 & & \\ 0 & \ddots & \ddots & \\ & \ddots & \ddots & 0 \\ & & 0 & H_N \end{bmatrix}$$

$$Q = \begin{bmatrix} Q_1 & 0 & & \\ 0 & \ddots & \ddots & \\ & \ddots & \ddots & 0 \\ & & 0 & Q_N \end{bmatrix}, \quad R = \begin{bmatrix} R_1 & 0 & & \\ 0 & \ddots & \ddots & \\ & \ddots & \ddots & 0 \\ & & 0 & R_N \end{bmatrix}$$

We also introduce notation for the full state X, measurements z, and initial state w:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \quad z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}, \quad w = \begin{bmatrix} x_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

The Kalman smoother estimate for the velocities (and their derivatives) from direct observations would be obtained by solving the single least squares problem

$$\min_x ||Gx - w||^2_{Q^{-1}} + ||Hx - z||^2_{R^{-1}}$$

7.4.2 ADCP Measurements

We now consider unknown current profiles c_u and c_v, which connect to the glider states in the previous sections via the simple equations

$$c_u = \dot{e}^g + z_u^o + \epsilon_{cu}$$

$$c_v = \dot{n}^g + z_v^o + \epsilon_{cv}$$

The additional decision variables c_u, c_v are indexed by depth. The model linking the ADCP observations to current profiles and states is given by

$$z_u^o = A_c c_u - A_u x + \epsilon_{cu}$$

$$z_v^o = A_c c_v - A_v x + \epsilon_{cv}$$

where A_u, A_v select out the OTG velocities at the time the measurements were taken and A_c selects out the depths for the given measurement.

Depending on the discretization of c_u, c_v, there may be depths without associated measurements. To estimate currents at these depths and to smooth out the final ocean velocity estimate we impose regularization terms on c_u and c_v:

$$R(c_u, c_v) := \sum_{i=1}^{N-1} (c_{ui} - c_{ui+1})^2 + (c_{vi} - c_{vi+1})^2,$$

which can be written as

$$R(c_u, c_v) = ||A_r c_u||^2 + ||A_r c_v||^2$$

where A_r compute adjacent differences.

7.4.3 Comparing OTG and TTW Velocities

The difference between OTG and TTW Velocities depends on the current, so we add a least squares term for this estimation. The three variables are connected via the equations

$$\dot{e}^g = e\_ttw^g + c_u + \epsilon_e v$$

$$\dot{n}^g = n\_ttw^g + c_v + \epsilon_n v$$

These can be written using existing variables as follows

$$A_{OTGe} x - (A_{TTWe} x_{TTW} + A_{cvelu} c_u) + \epsilon_e v = 0$$

$$A_{OTGn} x - (A_{TTWn} x_{TTW} + A_{cvelv} c_v) + \epsilon_n v = 0,$$

where A_{OTG}, A_{TTW} select the appropriate velocities at each time and A_{cvel} selects the current for the depth at the given time.

7.4.4 Joint Inversion

Combining the state-space model with the ADCP observations, current profiles, and velocity comparisons gives the least squares problem

$$\begin{aligned}\min_{x,x_{TTW},c_u,c_v} &||Gx-w||^2_{Q^{-1}} + ||Hx-z||^2_{R^{-1}} \\ &+\eta_1||A_c c_v - A_v x - z_v^o||^2 + \eta_1||A_c c_u - A_u x - z_u^o||^2 \\ &+\eta_2||A_{OTGe}x - (A_{TTWe}x_{TTW} + A_{cvelu}c_u)||^2 \\ &+\eta_2||A_{OTGn}x - (A_{TTWn}x_{TTW} + A_{cvelv}c_v||^2 \\ &+\eta_3||A_r c_u||^2 + \eta_3||A_r c_v||^2\end{aligned} \tag{7.3}$$

The joint inverse problem is interesting because it violates the classic Kalman smoothing block tridiagonal structure. In particular, while G^TG is sparse and block tridiagonal, and H^TH is sparse and block diagonal, the matrix $A_v^TA_v$ is a generic sparse matrix. In the experiments, we exploit the sparsity of the final least squares problem (7.3) to solve the problem efficiently. We leave further structure-exploiting innovations for ADCP-informed navigation to future work.

7.5 Experimental Data Collection

As part of the Canada Basin Glider Experiment (CABAGE), two Seagliders, SG196 and SG198, were deployed on 6 August 2017 at the shelf break north of Prudhoe Bay, AK. From there, they flew up to and around the CANAPE mooring array until they were recovered on 17 September 2017, for a total of 49 days. Together the gliders covered approximately 1730 km over the course of 712 dives, with SG196 diving to 480 m depth and SG198 diving to 750 m. Figure 7.2 shows the glider tracklines for both a short test deployment in 2016 and the 2017 deployment.

Each glider was equipped with a Nortek Signature1000 1 MHz ADCP, as well as the standard suite of conductivity temperature (CT) sensor, pressure sensor, WHOI MicroModem, and custom-built passive marine acoustic recorders (PMARs). Figure 7.1 shows the gliders

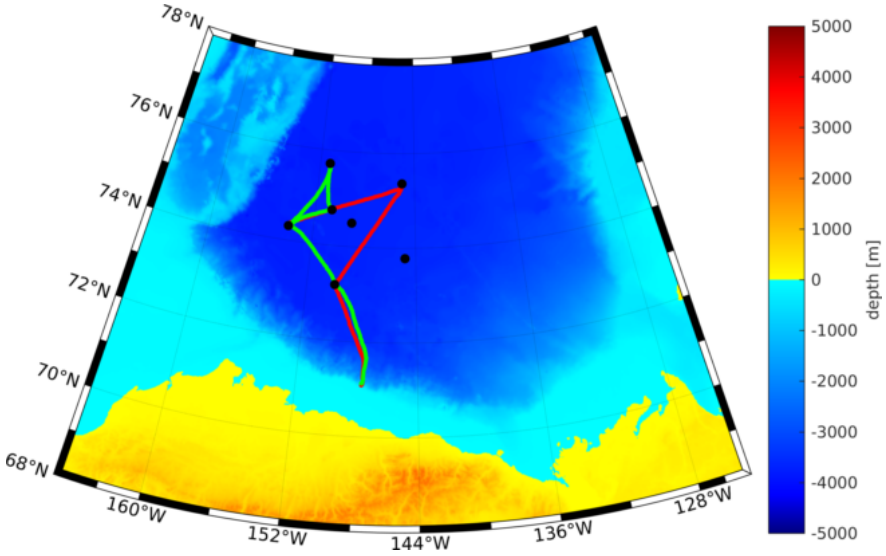


Figure 7.2: SG196 (green) and SG198 (red) were deployed at the shelf break north of Prudhoe Bay, AK. From there, they flew up to and around the CANAPE mooring array (black dots) for 49 days until they were recovered by the USCGC Healy.

loaded on the R/V Ukpik, ready for launch, with the upward-facing ADCPs, installed in the tail section, clearly visible.

For clarity during the discussion here, when referring to the ADCP data, we will use *trace* to refer to an individual profile collected by the ADCP, and *profile* to refer to the result of the inverse, i.e. the current profile for the entire dive. The ADCPs were programmed to collect a trace every 15 seconds with 2.0 m bins. Each trace typically covered 10-15 m depth, with the actual usable range varying with the amount of acoustic scatterers in the water. In practice any given depth bin was covered by 5-7 different traces. Figure 7.3 illustrates a section of the current profile with overlapping ADCP traces after alignment.

7.6 Results

In this section we compare the results obtained using the approaches detailed in Sections 7.3 and 7.4.

Figure 7.3 shows a short section of the current profile with the individual overlapping traces that have been aligned and averaged to produce the final current profile (black). This is from a 750 m dive, so the sections of data shown were collected 3.5 hrs apart. The difference in the current profile during ascent and descent are clearly visible.

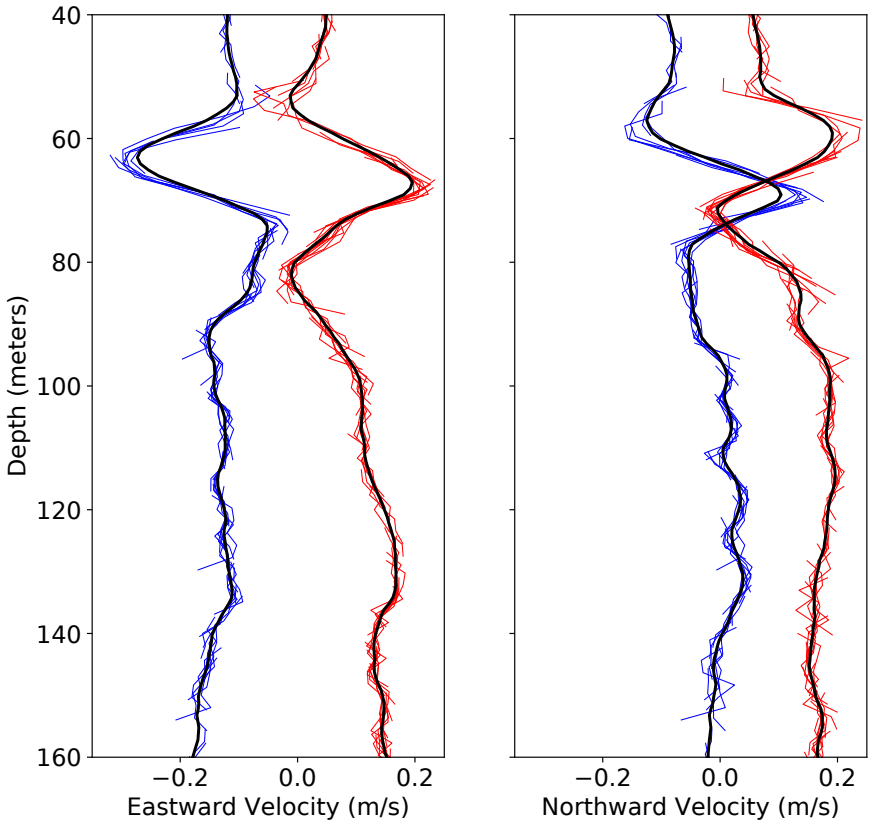


Figure 7.3: Overlapping ADCP traces during descent (blue) and ascent (red) for dive 99 of sg198, after alignment. The current profile produced by the state-space approach is shown as the thick black line.

A comparison of the results of the two methods for the entire dive is shown in Figure

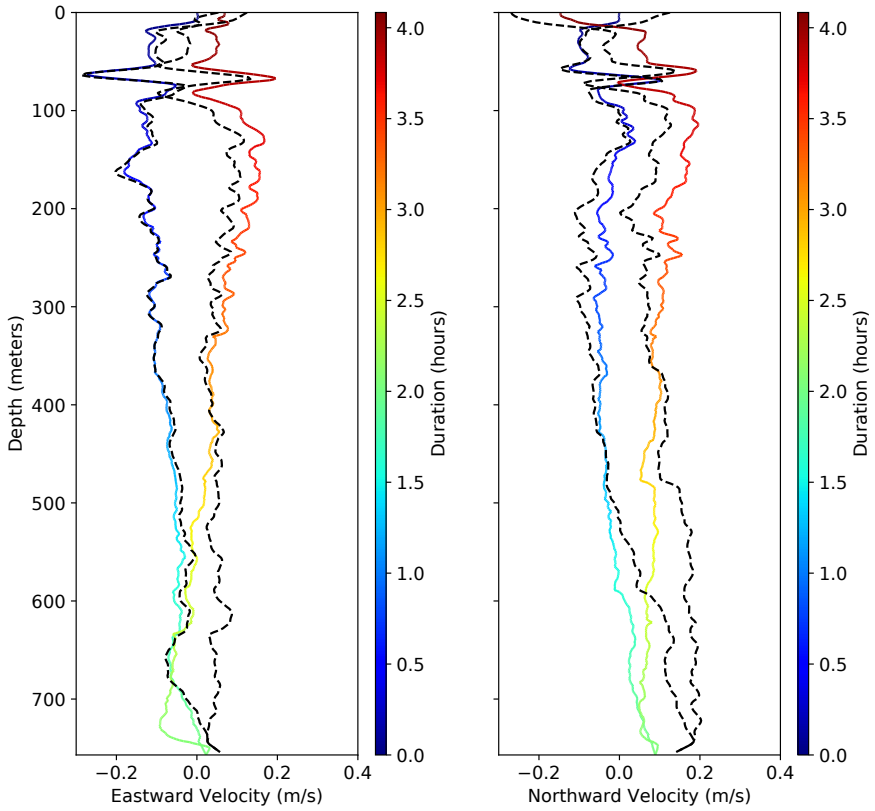


Figure 7.4: Comparison of the full results from the method of Section 7.3 (black dashed) to that of Section 7.4 (colored).

7.4. Both approaches find highly correlated estimates of current profiles, which is reassuring, though there is a clear offset between the results that varies by depth. We believe that this is due in large part to the sensitivity of the state-space approach to the weighting applied to the different terms in the minimization, which are difficult challenging to vet without ground truth for the entire current profile.

The only ground truth available for this dive is for the upper 40 m, shown in Figure 7.5. These data are from an upward-facing 600 kHz moored ADCP that was approximately 13

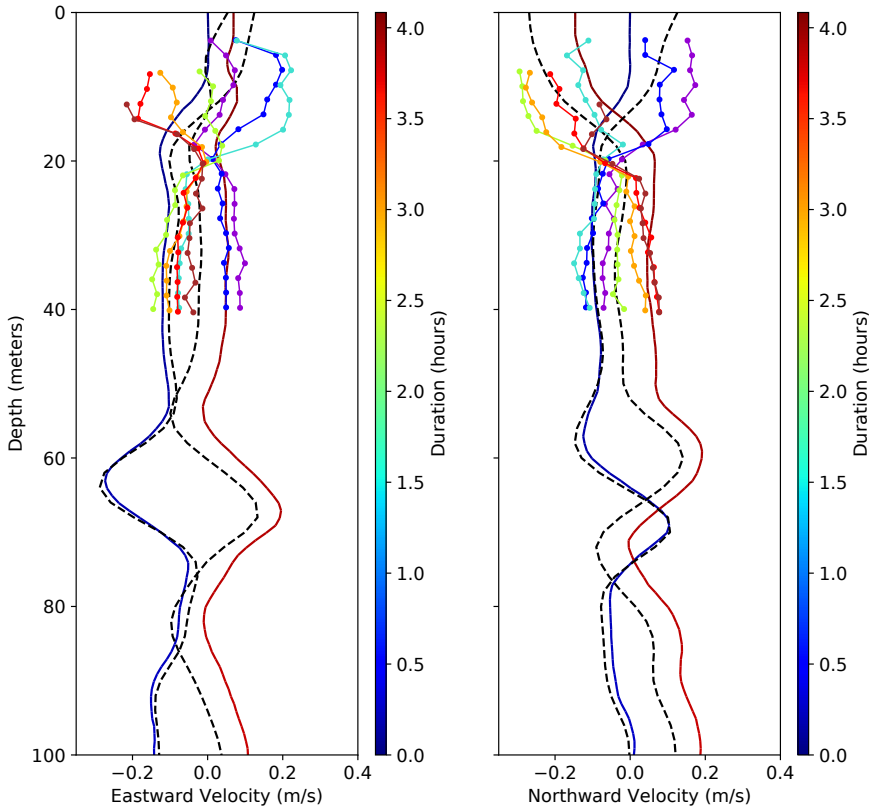


Figure 7.5: Zooming in on the upper 100m of the profiles in Figure 7.4, we use current profile data from a 600 kHz ADCP on a mooring that was approx. 13 km away during this dive. Current data was collected hourly. The time evolution of the surface currents over the 5 hours during the dive, is captured by the color scale, starting with blue at the beginning of the dive and ending with red.

km away during this dive. The time evolution of the surface currents, measured hourly, is captured by the same color scale used for the glider data, starting with blue at the beginning of the dive and ending with red. Current data, as shown here, has not yet been corrected for the motion of the mooring. This correction and aggregate comparisons for other dives that

are near moorings is the topic of future work.

The second approach also produces updated velocity profiles, which are shown in Figure 7.6 and the resulting glider trajectory, shown in Figure 7.7. Comparing the x-y position estimates from a glider trajectory computed using the naive approach of uniformly applying a single depth-averaged current estimate across the dive versus using the ADCP-based current profile to inform the glider position throughout the dive, shows a dramatic difference. The ADCP-based correction not only shifts the trajectory, but also compresses the dive and stretches the climb. This leads to a maximum horizontal offset of 449m in this case. We look forward to using range measurements from the CANAPE moorings to both improve and validate these results in future work.

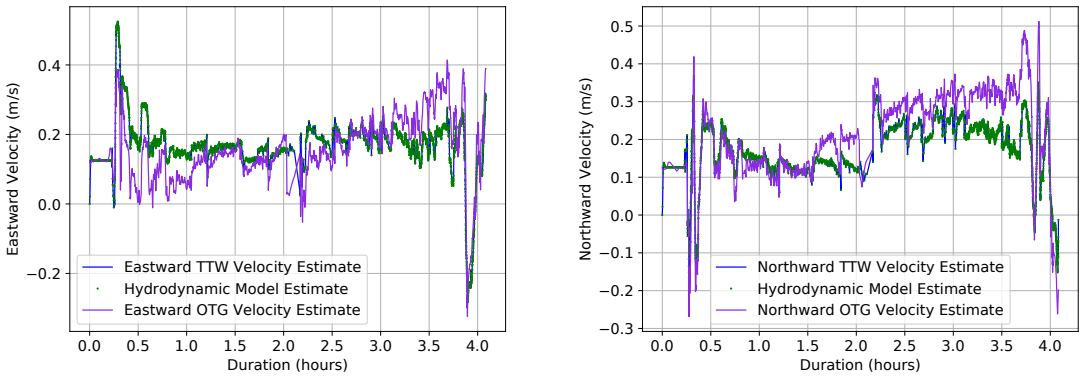


Figure 7.6: Updated velocities after processing the ADCP data.

7.7 Conclusions and Future Work

The main contribution of the work is to validate the idea of solving the state-space and current deconvolution problem using ADCP measurements. Two related approaches were developed, implemented, and compared; both made use of glider-estimated velocities, ADCP observations, and GPS fixes. The estimated current profiles were reasonably comparable to those obtained by a 'ground truth' ADCP on a nearby mooring.

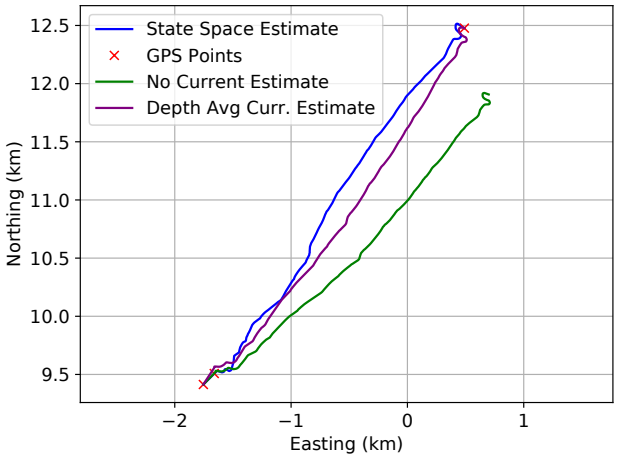


Figure 7.7: We compare position estimates from a glider trajectory computed without knowledge of current (dead reckoning), using the naive approach of applying a depth-averaged current uniformly across the dive, and using the ADCP-based current profile to inform the glider position throughout the dive. The ADCP-based correction not only shifts the trajectory, but also compresses the dive and stretches the climb, leading to a max horizontal offset of 449m.

The state-space approach in Section 7.4 is more informative than the first, since it also produces modified post-processed state estimates of the glider, see Figure 7.6. While there is no ground truth on the estimated velocities, the high degree of correlation between the deconvolution and inversion method for estimating current profiles validates the approach, and opens doors for future work using the state-space approach.

First, the state-space approach builds an explicit connection between variables used during navigation to those informed by the ADCP. This is a promising development for ADCP-aided navigation.

Second, the state-space framework allows a broad set of optimization-based tools to be applied to the deconvolution problem. In particular we can incorporate constraints [29], ro-

bust losses [18], and nonsmooth regularization [16, 10], as well as singular state-space models that make use of these innovations [76]. The state-space formulation also allows nonlinear models [29, 17], as needed by range measurements. All of these innovations make it possible to fuse information from noisy measurements along with statistical models that are robust to noisy data and constraints that incorporate prior knowledge. Efficient implementation of these formulations requires significant additional work in understanding the structure of the underlying linear algebra problem (7.3).

Acknowledgements

This work was completed as part of the Canada Basin Glider Experiment (CABAGE). Funding for this work was provided by the U.S. Office of Naval Research (ONR) through the Arctic and Global Predictions Program (Award #N00014-16-2596, PI: Dr. Sarah Webster, APL-UW) and the Defense Research and Development Canada (DRDC) (Contract #W7707-175902/001/HAL, PI: Dr. Sarah Webster, APL-UW). Additional funding for CABAGE was provided by ONR Ocean Acoustics Program (Award #N00014-17-1-2228, PI: Dr. Lora Van Uffelen, URI).

We would like to acknowledge Drs. Peter Worcester and Matthew Dzieciuch from Scripps Institution of Oceanography, who led the Canada Basin Acoustic Propagation Experiment (CANAPE). CANAPE provided all of the mooring infrastructure as well as all of the ice breaker logistics that enabled CABAGE.

This work could not have been completed without the engineers, technicians, and operators that made the Seaglider deployments possible. To that end, the authors would like to thank the Integrative Observational Platforms (IOP) Lab, in particular Craig Lee, who supplied the Seagliders used in the field experiment; Jason Gobat, the Senior Engineer; and Ben Jokinen, the Seaglider technician responsible for installing the ADCPs and preparing, testing, and launching the gliders. We are grateful for the support of Michael Flemming and Bill Kopplin, captain and current and former owners of the R/V Ukpik; the captain and crew of the USCGC Healy; and graduate student Wendy Snyder from University of Rhode

Island, who assisted with the glider recovery. In addition, Jason Gobat and Geoff Shilling assisted with glider piloting.

BIBLIOGRAPHY

[1] A. Aravkin, P. Kambadur, A. Lozano, and R. Luss. Orthogonal matching pursuit for sparse quantile regression. In *Data Mining (ICDM), International Conference on*, pages 11–19. IEEE, 2014. 106

[2] B. Ait-El-Fquih and F. Desbouvries. Fixed-interval Kalman smoothing algorithms in singular state–space systems. *Journal of Signal Processing Systems*, 65(3):469–478, 2011. 100

[3] D. J. Allerton and H. Jia. A review of multisensor fusion methodologies for aircraft navigation systems. *The Journal of Navigation*, 58(3):405–417, 2005. 69, 83

[4] M. Amitay and A. Glezer. Controlled transients of flow reattachment over stalled airfoils. *International Journal of Heat and Fluid Flow*, 23(5):690–699, 2002. 87

[5] X. An, D. R. Williams, J. Eldredge, and T. Colonius. Modeling dynamic lift response to actuation. In *54th AIAA Aerospace Sciences Meeting*, page 0058, 2016. 87

[6] B. D. Anderson and J. B. Moore. *Optimal control: linear quadratic methods*. Courier Corporation, 2007. 1, 17, 35

[7] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice Hall, 1979. 1, 2, 13, 17, 18, 19, 35

[8] C. F. Ansley and R. Kohn. A geometric derivation of the fixed interval smoothing algorithm. *Biometrika*, 69:486–487, 1982. 18

[9] C. F. Ansley and R. Kohn. A geometrical derivation of the fixed interval smoothing algorithm. *Biometrika*, 69(2):486–487, 1982. 100

[10] A. Aravkin, J. V. Burke, L. Ljung, A. Lozano, and G. Pillonetto. Generalized kalman smoothing: Modeling and algorithms. *Automatica*, 86:63–86, 2017. 18, 20, 30, 96, 98, 99, 103, 136

[11] A. Aravkin, J. V. Burke, and G. Pillonetto. Robust and trend-following kalman smoothers using student's t. *IFAC Proceedings Volumes*, 45(16):1215–1220, 2012. 74

[12] A. Aravkin, K. N. Ramamurthy, and G. Pillonetto. Kalman smoothing with persistent nuisance parameters. In *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2014. 10, 13

[13] A. Y. Aravkin, B. M. Bell, J. V. Burke, and G. Pillonetto. An-laplace robust kalman smoother. *Automatic Control, IEEE Transactions on*, 56(12):2898–2911, 2011. 10, 11, 55

[14] A. Y. Aravkin, B. M. Bell, J. V. Burke, and G. Pillonetto. An ℓ_1-Laplace robust Kalman smoother. *IEEE Transactions on Automatic Control*, 56(12):2898–2911, 2011. 97

[15] A. Y. Aravkin, J. V. Burke, and G. Pillonetto. Sparse/robust estimation and kalman smoothing with nonsmooth log-concave densities: Modeling, computation, and theory. *Journal of Machine Learning Research*, 14:2689–2728, 2013. 11, 26, 55, 62, 63, 64

[16] A. Y. Aravkin, J. V. Burke, and G. Pillonetto. Sparse/robust estimation and Kalman smoothing with nonsmooth log-concave densities: Modeling, computation, and theory. *J. Mach. Learn. Res.*, 14(1):2689–2728, 2013. 97, 136

[17] A. Y. Aravkin, J. V. Burke, and G. Pillonetto. Optimization viewpoint on kalman smoothing with applications to robust and sparse estimation. In *Compressed sensing & sparse filtering*, pages 237–280. Springer, 2014. 5, 6, 74, 136

[18] A. Y. Aravkin, J. V. Burke, and G. Pillonetto. Robust and trend-following student's t kalman smoothers. *SIAM Journal on Control and Optimization*, 52(5):2891–2916, 2014. 12, 41, 55, 97, 136

[19] A. Y. Aravkin, D. Drusvyatskiy, and T. van Leeuwen. Efficient quadratic penalization through the partial minimization technique. *IEEE Transactions on Automatic Control*, 63(7):2131–2138, 2017. 37

[20] A. Y. Aravkin and T. Van Leeuwen. Estimating nuisance parameters in inverse problems. *Inverse Problems*, 28(11):115016, 2012. 37

[21] T. Askham and J. N. Kutz. Variable projection methods for an optimized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 17(1):380–416, 2018. 74

[22] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, 2001. 20, 32, 108

[23] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation*. John Wiley & Sons, Inc., New York, 2001. 1, 17, 35

[24] B. Bell. The Iterated {Kalman} Smoother as a {Gauss-Newton} Method. *SIAM J. Optimization*, 4(3):626–636, Aug. 1994. 97

[25] B. M. Bell. The iterated kalman smoother as a gauss–newton method. *SIAM Journal on Optimization*, 4(3):626–636, 1994. 8

[26] B. M. Bell. The marginal likelihood for parameters in a discrete gauss-markov process. *IEEE Transactions on Signal Processing*, 48(3):870–873, 2000. 36, 53

[27] B. M. Bell and J. V. Burke. Algorithmic differentiation of implicit functions and optimal values. In *Advances in Automatic Differentiation*, pages 67–77. Springer, 2008. 39, 40

[28] 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, Jan. 2008. 20

[29] 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, 2009. 10, 55, 97, 135, 136

[30] B. M. Bell and F. W. Cathey. The iterated kalman filter update as a gauss-newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, 1993. 8

[31] Y. Bengio, I. Goodfellow, and A. Courville. *Deep learning*, volume 1. MIT press, 2017. 70

[32] G. Bottegal, A. Aravkin, H. Hjalmarsson, and G. Pillonetto. Robust em kernel-based methods for linear system identification. *arXiv prepring arXiv:1411.5915v3*, 2016. 37

[33] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and regression trees. wadsworth int. *Group*, 37(15):237–251, 1984. 80

[34] R. G. Brown, P. Y. Hwang, et al. *Introduction to random signals and applied Kalman filtering*, volume 3. Wiley New York, 1992. 4

[35] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz. Chaos as an intermittently forced linear system. *Nature Communications*, 8(1), Dec. 2017. 71, 74, 79

[36] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLoS ONE*, 11(2):e0150171, 2016. 79

[37] S. L. Brunton and J. N. Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019. 70

[38] S. L. Brunton, B. R. Noack, and P. Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020. 70

[39] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, Apr. 2016. 70, 96

[40] S. L. Brunton, C. W. Rowley, and D. R. Williams. Reduced-order unsteady aerodynamic models at low Reynolds numbers. *Journal of Fluid Mechanics*, 724:203–233, 2013. 70

[41] J. Burke and A. Engle. Strong metric (sub)regularity of kkt mappings for piecewise linear-quadratic convex-composite optimization. *arXiv:1805.01073v2*, 2017. 55, 59, 60, 62

[42] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11–1–11–37, 2011. 96

[43] K. Champion, S. L. Brunton, and J. N. Kutz. Discovery of nonlinear multiscale systems: Sampling strategies and embeddings. *SIAM Journal on Applied Dynamical Systems*, 18(1):312–333, 2019. 79

[44] W. Chu, S. S. Keerthi, and C. J. Ong. A unified loss function in bayesian framework for support vector regression. *Epsilon*, 1(1.5):2, 2001. 106

[45] C. Chui and G. Chen. *{K}alman Filtering with Real-Time Applications*. Springer, 4 edition, 2009. 105

[46] C. K. Chui, G. Chen, et al. *Kalman filtering*. Springer, 2017. 100

[47] T. Colonius and D. R. Williams. Control of vortex shedding on two-and three-dimensional aerofoils. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 369(1940):1525–1539, 2011. 87

[48] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011. 24, 25, 102

[49] D. Davis and W. Yin. Convergence rate analysis of several splitting schemes. In *Splitting Methods in Communication, Imaging, Science, and Engineering*, pages 115–163. Springer, 2016. 24, 101

[50] R. E. Davis. On the coastal-upwelling overturning cell. *J. Marine Res.*, 68(3-4):369–385, May 2010. 119

[51] R. Dunne and B. J. McKeon. Dynamic stall on a pitching and surging airfoil. *Experiments in Fluids*, 56(8):1–15, 2015. 87

[52] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992. 101

[53] B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004. 106

[54] J. D. Eldredge and A. R. Jones. Leading-edge vortices: mechanics and modeling. *Annual Review of Fluid Mechanics*, 51:75–104, 2019. 87

[55] L. V. Eykeren and Q. P. Chu. Sensor fault detection and isolation for aircraft control systems by kinematic relations. *Control Engineering Practice*, 31:200–210, 2014. 70, 72, 73, 74, 87

[56] L. Fahrmeir and H. Kaufmann. On kalman filtering, posterior mode estimation and fisher scoring in dynamic exponential family regression. *Metrika*, 38(1):37–60, 1991. 9

[57] L. Fahrmeir and H. Kaufmann. On {K}alman Filtering, Posterior Mode Estimation, and {F}isher Scoring in Dynamic Exponential Family Regression. *Metrika*, pages 37–60, 1991. 97

[58] L. Fahrmeir and R. Künstler. Penalized likelihood smoothing in robust state space models. *Metrika*, 49(3):173–191, 1999. 8

[59] S. Farahmand, G. B. Giannakis, and D. Angelosante. Doubly robust smoothing of dynamical processes via outlier sparsity constraints. *IEEE Transactions on Signal Processing*, 59(10):4529–4543, 2011. 8

[60] S. Farahmand, G. B. Giannakis, and D. Angelosante. Doubly Robust Smoothing of Dynamical Processes via Outlier Sparsity Constraints. *IEEE Transactions on Signal Processing*, 59:4529–4543, 2011. 97

[61] E. Firing and R. L. Gordon. Deep ocean acoustic Doppler current profiling. In *Proceedings of the IEEE Fourth Working Conference on Current Measurement*, pages 192–201, April 1990. 119

[62] J. Fischer and M. Visbeck. Deep velocity profiling with self-contained ADCPs. *Journal of Atmospheric and Oceanic Technology*, 10(5):764–773, 1993. 119

[63] D. Fraser and J. Potter. The optimum linear smoother as a combination of two optimum linear filters. *IEEE Transactions on automatic control*, 14(4):387–390, 1969. 97

[64] D. A. Freedman. *Statistical models: theory and practice*. Cambridge University Press, 2009. 106

[65] S. Gillijns, O. B. Mendoza, J. Chandrasekar, B. L. R. D. Moor, D. S. Bernstein, and A. Ridley. What is the ensemble kalman filter and how well does it work? In *Proceedings of the 2006 American Control Conference*, pages 4448–4453, 2006. 4

[66] D. D. Gobbo, M. Napolitano, P. Famouri, and M. Innocenti. Experimental application of extended Kalman filtering for sensor validation. *IEEE Transactions on Control Systems Technology*, 9(2), 2001. 70, 72, 73, 74

[67] M. Goman and A. Khrabrov. State-space representation of aerodynamic characteristics of an aircraft at high angles of attack. *Journal of Aircraft*, 31(5):1109–1115, 1994. 87, 88

[68] P. Goupil and A. Marcos. Advanced diagnosis for sustainable flight guidance and control: the European ADDSAFE project. Technical paper 2011-01-2804, SAE, 2011. 70

[69] C. Hajiyev. Testing the covariance matrix of the innovation sequence with sensor/actuator fault detection applications. *International Journal of Adaptive Control and Signal Processing*, 24:717–730, 2010. 70, 72, 73, 74

[70] T. Hastie and R. Tibshirani. Generalized additive models. In *Monographs on Statistics and Applied Probability*, volume 43. Chapman and Hall, London, UK, 1990. 106

[71] M. S. Hemati, S. T. Dawson, and C. W. Rowley. Parameter-varying aerodynamics models for aggressive pitching-response prediction. *AIAA Journal*, pages 1–9, 2016. 70

[72] R. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder. *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008. 100

[73] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439–454, 2002. 1, 17, 18, 35, 36

[74] A. Isaksson, D. Törnqvist, J. Sjöberg, and L. Ljung. Grey-box identification based on horizon estimation and nonlinear optimization. In *Proceedings of the 41st ISCIE Internation Symposium on Stochastic Systems theory and its applications*, pages 1–6, 2010. 36, 53

[75] A. Jazwinski. *Stochastic Processes and Filtering Theory*. Dover Publications, Inc, 1970. 20

[76] J. Jonker, A. Aravkin, J. V. Burke, G. Pillonetto, and S. Webster. Fast robust methods for singular state-space models. *Automatica*, 105:399–405, 2019. 42, 43, 98, 99, 100, 102, 103, 136

[77] M. R. Jovanović, P. J. Schmid, and J. W. Nichols. Sparsity-promoting dynamic mode decomposition. *Physics of Fluids*, 26, 2014. 74

[78] J. N. Juang and R. S. Pappa. An eigensystem realization algorithm for modal parameter identification and model reduction. *J. of Guidance, Control, and Dynamics*, 8(5):620–627, 1985. 70, 79

[79] J. N. Juang, M. Phan, L. G. Horta, and R. W. Longman. Identification of observer/Kalman filter Markov parameters: theory and experiments. Technical Memorandum 104069, NASA, 1991. 70, 79

[80] E. Kaiser, J. N. Kutz, and S. L. Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society of London A*, 474(2219), 2018. 96

[81] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. 97

[82] 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. 1, 18, 54

[83] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960. 72

[84] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Trans. ASME J. Basic Eng*, 83:95–108, 1961. 97

[85] S. A. Kassam and H. V. Poor. Robust Techniques for Signal Processing: A Survey. *Proceedings of the IEEE*, 73(3):433–481, Mar. 1985. 97

[86] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky. ℓ_1 trend filtering. *SIAM review*, 51(2):339–360, 2009. 97

[87] S. Ko and R. R. Bitmead. State estimation for linear systems with state equality constraints. *Automatica*, 43(8):1363–1368, 2007. 100

[88] R. Koenker and G. Bassett. Regression quantiles. *Econometrica*, pages 33–50, 1978. 106

[89] R. Koenker and O. Geling. Reappraising medfly longevity: A quantile regression survival analysis. *Journal of the American Statistical Association*, 96:458–468, 2001. 106

[90] S. J. Koopman. Exact initial Kalman filtering and smoothing for nonstationary time series models. *Journal of the American Statistical Association*, 92(440):1630–1638, 1997. 100

[91] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*, volume 149. SIAM, 2016. 71, 72

[92] P. Latafat, N. Freris, and P. Patrinos. A new randomized block-coordinate primal-dual proximal algorithm for distributed optimization. *arXiv preprint arXiv:1706.02882*, 2017. 26, 28

[93] Y.-J. Lee, W.-F. Hsieh, and C.-M. Huang. ϵ-ssvr: a smooth support vector machine for ϵ;-insensitive regression. *Knowledge and Data Engineering, IEEE Transactions on*, 17(5):678–685, 2005. 106

[94] J. G. Leishman. *Principles of Helicopter Aerodynamics*. Cambridge University Press, 2002. 87

[95] R. Li and J. H. Olson. Fault detection and diagnosis in a closed-loop nonlinear distillation process: Application of extended Kalman filters. *Industrial & Engineering Chemistry Research*, 30:898–908, 1991. 70

[96] A. Liaw, M. Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002. 96

[97] L. Ljung. Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems. *IEEE Transactions on Automatic Control*, 24(1):36–50, 1979. 70

[98] L. Ljung. *System Identification - Theory for the User*. Prentice-Hall, Upper Saddle River, N.J., 2nd edition, 1999. 6

[99] S. Lomax and S. Vadera. A survey of cost-sensitive decision tree induction algorithms. *ACM Computing Surveys (CSUR)*, 45(2):1–35, 2013. 96

[100] K. Madsen, H. Nielsen, and O. Tingleff. Methods for non-linear least squares problems (2nd ed.). page 60, 01 2004. 60

[101] J. Magill, M. Bachmann, and G. Rixon. Dynamic stall control using a model-based observer. *Journal of Aircraft*, 40(2):355–362, 2003. 87

[102] S. Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150203, 2016. 70

[103] R. A. Maronna, D. Martin, and Yohai. *Robust Statistics*. Wiley Series in Probability and Statistics. Wiley, 2006. 106

[104] R. A. Maronna, R. D. Martin, V. J. Yohai, et al. *Robust statistics: theory and methods (with R)*. John Wiley & Sons, 2019. 12

[105] C. J. McFarland, M. V. Jakuba, S. Suman, J. C. Kinsey, and L. L. Whitcomb. Toward ice-relative navigation of underwater robotic vehicles under moving sea ice: Experimental evaluation in the arctic sea. In *Proc. IEEE Intl. Conf. Robot. Auto. (ICRA)*, pages 1527–1534, May 2015. 118

[106] L. Medagoda, M. V. Jakuba, O. Pizarro, and S. B. Williams. Water column current profile aided localisation for autonomous underwater vehicles. In *Proc. IEEE/MTS OCEANS Conf. Exhib.*, page 10 pp., Sydney, Australia, May 2010. 119

[107] R. K. Mehra and J. Peschon. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7(637-640), 1971. 70, 73

[108] MIL-HDBK-1797. *Flying qualities of piloted aircraft*. Dept. of Defense, 1990. 92

[109] H. N. Mocan. Structural unemployment, cyclical unemployment, and income inequality. *Review of Economics and Statistics*, 81(1):122–134, 1999. 37

[110] M. R. Napolitano. Online learning neural architectures and cross-correlation analysis for actuator failure detection and identification. *International Journal of Control*, 63(3):433–455, 1996. 70

[111] M. R. Napolitano, C. Chen, and S. Naylo. Aircraft failure detection and identification using neural networks. *Journal of Guidance, Control, and Dynamics*, 16(6):999–1009, 1993. 70

[112] J. Nathan Kutz, J. L. Proctor, and S. L. Brunton. Applied koopman theory for partial differential equations and data-driven modeling of spatio-temporal systems. *Complexity*, 2018, 2018. 74, 79

[113] H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd. Smoothed state estimates under abrupt changes using sum-of-norms regularization. *Automatica*, 48:595–605, 2012. 6, 13

[114] H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd. Smoothed state estimates under abrupt changes using sum-of-norms regularization. *Automatica*, 48:595–605, 2012. 20, 21

[115] B. Oksendal. *Stochastic Differential Equations*. Springer, sixth edition, 2005. 20

[116] M. V. OL, A. Altman, J. D. Eldredge, D. J. Garmann, and Y. Lian. Résumé of the AIAA FDTC low Reynolds number discussion group's canonical cases. AIAA Paper 2010-1085, 48th Aerospace Sciences Meeting, January 2010. 87

[117] C. Paige. Computer solution and perturbation analysis of generalized linear least squares problems. *Mathematics of Computation*, 33:171–183, jan 1979. 19, 20

[118] C. C. Paige. Covariance matrix representation in linear filtering. *Special Issue of Contemporary Mathematics on Linear Algebra and its Role in Systems Theory, AMS*, 1985. 8

[119] C. C. Paige and M. Saunders. Least squares estimation of discrete linear dynamic systems using orthogonal transformations. *SIAM Journal on Numerical Analysis*, 14(2):180–193, 1977. 8

[120] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 80

[121] R. R. Perez, V. F. Batlle, and L. S. Rodriguez. Robust system identification of an irrigation main canal. *Advances in Water Resources*, 30:1785–1796, 2007. 37

[122] J. L. Proctor, S. L. Brunton, and J. N. Kutz. Dynamic mode decomposition with control. *SIAM Journal of Applied Dynamical Systems*, 15(1):142–161, 2016. 74, 77, 78

[123] H. E. Rauch, C. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965. 97

[124] H. E. Rauch, F. Tung, and C. T. Striebel. Maximum Likelihood estimates of linear dynamic systems. *AIAA J.*, 3(8):1145–1150, 1965. 4, 18, 54

[125] H. Raza, P. Ioannou, and H. M. Youssef. Surface failure detection for an F/A-18 aircraft using neural networks and fuzzy logic. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 5, pages 3363–3368, 1994. 70

[126] R. T. Rockafellar and R. J. B. Wets. *Variational Analysis*, volume 317. Springer, 1998. 11, 26, 55, 56

[127] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009. 103

[128] J. L. Rojo-Álvarex, M. Martínez-Ramón, M. de Prado-Cumplido, A. Artés-Rodríguez, and A. R. Rigueriras-Vidal. Support vector method for robust arma system identification. *IEEE Transactions on Signal Processing*, 52:155–164, 2004. 37

[129] L. Rokach and O. Z. Maimon. *Data mining with decision trees: theory and applications*, volume 69. World scientific, 2008. 79

[130] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. Henningson. Spectral analysis of nonlinear flows. *J. Fluid Mech.*, 645:115–127, 2009. 72, 74

[131] S. R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991. 71, 79

[132] I. Scherl, B. Strom, J. K. Shang, O. Williams, B. L. Polagye, and S. L. Brunton. Robust principal component analysis for particle image velocimetry. *arXiv preprint arXiv:1905.07062*, 2019. 96

[133] I. C. Schick and S. K. Mitter. Robust Recursive Estimation in the Presence of Heavy-Tailed Observation Noise. *The Annals of Statistics*, 22(2):1045–1080, June 1994. 97

[134] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010. 70, 71, 72, 74

[135] M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009. 70, 96

[136] G. Seber and C. Wild. *Nonlinear Regression*. Wiley Series in Probability and Statistics. Wiley, 2003. 106

[137] A. Seifert, A. Darabi, and I. Wygnanski. Delay of airfoil stall by periodic excitation. 33(4):691–698, 1996. 87

[138] Z. Shun and P. McCullagh. Laplace approximation of high dimensional integrals. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(4):749–760, 1995. 53

[139] M. Stanway. Water profile navigation with an acoustic doppler current profiler. In *Proc. IEEE/MTS OCEANS Conf. Exhib.*, pages 1–5, Sydney, Australia, May 2010. 119

[140] M. J. Stanway. Dead reckoning through the water column with an acoustic doppler current profiler: Field experiences. In *Proc. IEEE/MTS OCEANS Conf. Exhib.*, pages 1–8, Kona, HI, Sep. 2011. 119

[141] R. F. Stengel. *Flight Dynamics*. Princeton University Press, 2004. 92

[142] A. Surana. Koopman operator framework for time series modeling and analysis. *Journal of Nonlinear Science*, 2018. 70, 72, 73, 79

[143] A. Surana and A. Banaszuk. Linear observer synthesis for nonlinear systems using koopman operator framework. *IFAC-PapersOnLine*, 49(18):716–723, 2016. 70, 79

[144] K. Taira, S. L. Brunton, S. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley. Modal analysis of fluid flows: An overview. *AIAA Journal*, 55(12):4013–4041, 2017. 87

[145] K. Taira and T. Colonius. Effect of tip vortices in low-Reynolds-number poststall flow control. *AIAA Journal*, 47(3):749–756, 2008. 87

[146] F. Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981. 79

[147] A. M. Thurnherr, D. Symonds, and L. St. Laurent. Processing explorer ADCP data collected on Slocum gliders using the LADCP shear method. In *2015 IEEE/OES Eleveth Current, Waves and Turbulence Measurement (CWTM)*, pages 1–7, March 2015. 119

[148] R. E. Todd, D. L. Rudnick, M. R. Mazloff, R. E. Davis, and B. D. Cornuelle. Poleward flows in the southern California Current system: Glider observations and numerical simulation. *Journal of Geophysical Research: Oceans*, 116(C02026):2156–2202, 2011. 119, 120

[149] R. E. Todd, D. L. Rudnick, J. Sherman, W. Owens, and L. George. Absolute velocity estimates from autonomous underwater gliders equipped with doppler current profilers. *J. Atmos. Oceanic Tech.*, 34(2):309–333, 2017. 119

[150] R. S. Tsay. *Analysis of financial time series*, volume 543. John Wiley & Sons, 2005. 1, 17, 18, 35, 36

[151] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014. 70, 72, 74

[152] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, USA, 1998. 106

[153] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri. A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computational Chemical Engineering*, 27(3):293–311, 2003. 70

[154] M. Visbeck. Deep velocity profiling using lowered acoustic Doppler current profilers: Bottom track and inverse solutions. *Journal of Atmospheric and Oceanic Technology*, 19(5):794–807, 2002. 119, 120, 122

[155] B. K. Walker and K. Huang. FDI by extended Kalman filter parameter estimation for the industrial actuator benchmark. In *IFAC Symposiym SAFE-PROCESS*, pages 481–487, 1994. 70

[156] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. Ieee, 2000. 4

[157] S.-T. Wang and W. Frost. Atmospheric turbulence simulation techniques with application to flight analysis. Technical report, NASA, 1980. 92

[158] G. Welch, G. Bishop, et al. An introduction to the kalman filter. 1995. 70

[159] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, Dec. 2015. 74, 78

[160] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis. A kernel-based method for data-driven Koopman spectral analysis. *Journal of Computational Dynamics*, 2(2):247–265, 2015. 74

[161] A. S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12(6):601–611, 1976. 70

[162] S. J. Wright. *Primal-Dual Interior-Point Methods*. Siam, Englewood Cliffs, N.J., USA, 1997. 10

[163] D. R. Yoerger, A. M. Bradley, B. B. Walden, H. Singh, and R. Bachmeyer. Surveying a subsea lava flow using the autonomous benthic explorer (abe). *International Journal of Systems Science*, 29(10):1031–1044, 1998. 118

[164] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005. 106