PLQ Modeling and Optimization
with applications to machine learning, system identification, and Kalman Smoothing

James V. Burke
University of Washington
jvburke@uw.edu

Aleksandr Y. Aravkin
Numerical Analysis and Optimization
IBM T.J. Watson Research Center
saravkin@us.ibm.com

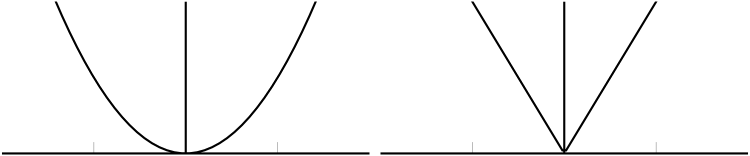
Gianluigi Pillonetto
Department of Information Engineering
University of Padova
giapi@dei.unipd.it

Vancouver Workshop and Michael Overton Fest 2013

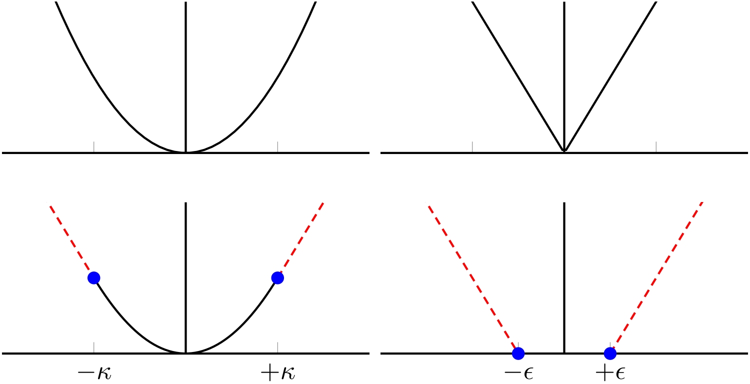
Outline

- Piecewise linear quadratic penalties
  - Examples and formulations
  - Dual representation
  - Representation calculus
  - Quadratic support functions
- Building a general interior point solver for the PLQ class
  - KKT system and IP strategy
  - Exploiting structure
  - Performance on simple problems
- Kalman smoothing
  - Brief introduction
  - PLQ formulation and efficiency
  - Numerical results

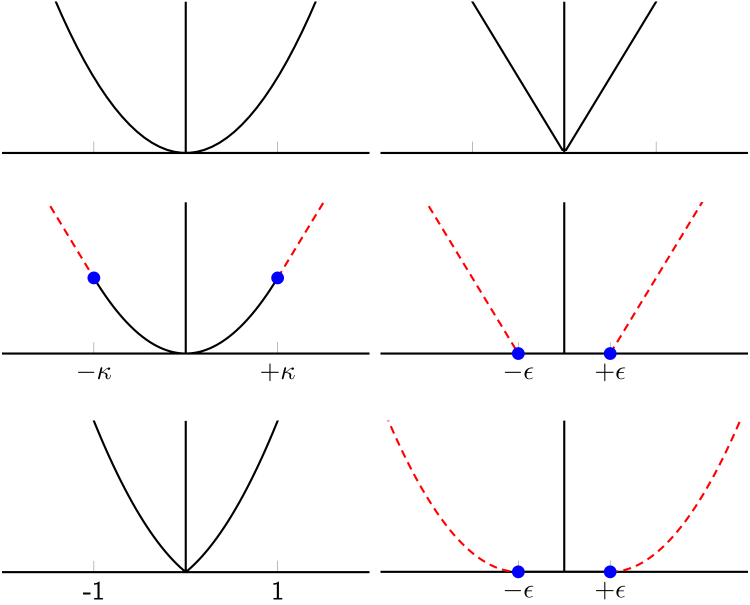
PLQ Examples



PLQ Examples



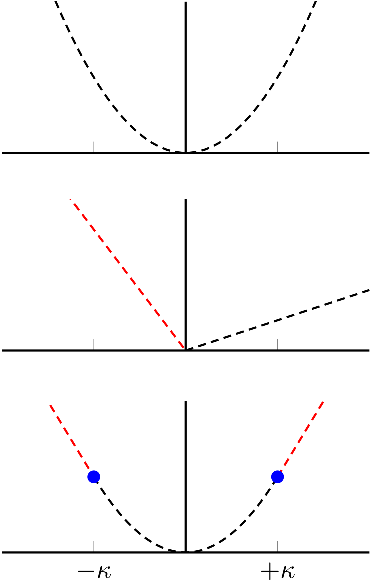
PLQ Examples



PLQ penalties in practice

Application	Objective	PLQs
Regression	$\|Ax-b\|^2$	L_2
Robust regression	$\rho_H(Ax-b)$	Huber
Quantile regression	$Q(Ax-b)$	Asymmetrical L_1
Lasso	$\|Ax-b\|^2+\lambda\|x\|_1$	$L_2 + L_1$
Robust lasso	$\rho_H(Ax-b)+\lambda\|x\|_1$	Huber + L_1
SVM	$\frac{1}{2}\|w\|^2+H(\mathbf{1}-Ax)$	L_1 + hinge loss
SVR	$\rho_V(Ax-b)$	Vapnik loss
Kalman smoother	$\|Gx-w\|^2_{Q^{-1}}+\|Hx-z\|^2_{R^{-1}}$	$L_2 + L_2$
Robust trend smoothing	$\|Gx-w\|_1+\rho_H(Hx-z)$	L_1 + Huber

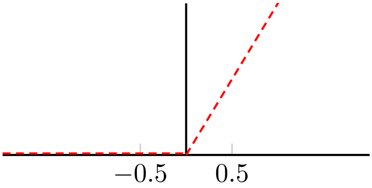
Dual representation of PLQs



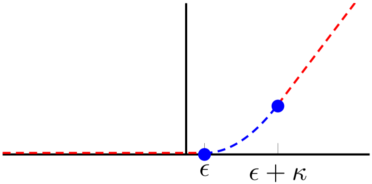
$$\frac{1}{2}x^2 = \sup_{u \in \mathbb{R}} \langle u, x \rangle - \frac{1}{2}u^2$$

$$Q_{0.8}(x) = \sup_{u \in [-0.8, 0.2]} \langle u, x \rangle$$

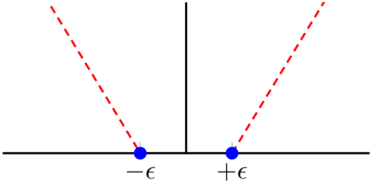
$$\rho_h(x) = \sup_{u \in [-\kappa, \kappa]} \langle u, x \rangle - \frac{1}{2}u^2$$



$$H(x) = \sup_{u \in [0,1]} \langle u, x \rangle$$



$$\rho_s(x) = \sup_{u \in [0,\kappa]} \langle u, x - \epsilon \rangle - \frac{1}{2} u^2$$



$$\rho_v(x) = \sup_{u \in [0,1]^2} \left\{ \left\langle \begin{bmatrix} y - \epsilon \\ -y - \epsilon \end{bmatrix}, u \right\rangle \right\}$$

PLQ Penalties

Definition: Piecewise Linear Quadratic Penalties (Rockafellar and Wets)

Define $\rho(U, M, b, B; \cdot) : \mathbb{R}^n \to \mathbb{R}$ as

$$\rho(U, b, B, M; y) = \sup_{u \in U} \left\{ \langle u, b + By \rangle - \frac{1}{2} \langle u, Mu \rangle \right\}$$

1. $M \in \mathbb{R}^{m \times m}$ is a symmetric positive semidefinite matrix.
2. $b + By$ is an injective affine transformation with $B \in \mathbb{R}^{m \times n}$.
3. $U \subset \mathbb{R}^m$ is a nonempty polyhedral set containing the origin.

PLQ Penalties

Definition: Piecewise Linear Quadratic Penalties (Rockafellar and Wets)

Define $\rho(U, M, b, B; \cdot) : \mathbb{R}^n \to \mathbb{R}$ as

$$\rho(U, b, B, M; y) = \sup_{u \in U} \left\{ \langle u, b + By \rangle - \frac{1}{2} \langle u, Mu \rangle \right\}$$

1. $M \in \mathbb{R}^{m \times m}$ is a symmetric positive semidefinite matrix.
2. $b + By$ is an injective affine transformation with $B \in \mathbb{R}^{m \times n}$.
3. $U \subset \mathbb{R}^m$ is a nonempty polyhedral set containing the origin.

Since U is polyhedral, it can be represented with a matrix and a vector:

$$U = \{u : Cu \leq c\} .$$

Fully represented PLQ object is given by

$$\rho(c, C, b, B, M; y) = \sup_{Cu \leq c} \left\{ \langle u, b + By \rangle - \frac{1}{2} \langle u, Mu \rangle \right\}$$

PLQ Calculus I: Addition

Given two PLQ penalties

$$\rho(c_1, C_1, B_1, b_1, M_1; y) \quad \text{and} \quad \rho(c_2, C_2, B_2, b_2, M_2; y)$$

their sum is also a PLQ penalty $\rho(c, C, B, b, M; y)$ with

$$c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}, \quad C = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad M = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix},$$

Vapnik:

$$(y - \epsilon)_+ := \sup_{u \in [0,1]} \langle u, y - \epsilon \rangle, \quad B_1 = 1, \quad b_1 = -\epsilon$$

$$(-y - \epsilon)_+ := \sup_{u \in [0,1]} \langle u, -y - \epsilon \rangle, \quad B_2 = -1, \quad b_2 = -\epsilon$$

$$\rho_v(x) = \sup_{u \in [0,1]^2} \left\{ \left\langle \begin{bmatrix} y - \epsilon \\ -y - \epsilon \end{bmatrix}, u \right\rangle \right\}, \quad B = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad b = \begin{bmatrix} -\epsilon \\ -\epsilon \end{bmatrix}$$

PLQ Calculus II: Affine composition

Given a PLQ penalty $\rho(c, C, b, B, M; y)$, consider $\rho(Px - p)$.

For example, given the penalty $\|\cdot\|^2$, consider $\|Px - p\|^2$.

$$\rho(c, C, b, B, M; Px - p) = \sup_{Cu \le c} \left\{ \langle u, b + B(Px - p) \rangle - \frac{1}{2} \langle u, Mu \rangle \right\}$$

The composite penalty is $\rho(c, C, \tilde{b}, \tilde{B}, M; y)$, where

$$\tilde{b} = b - Bp, \quad \tilde{B} = BP \; .$$

Bottom line: PLQ penalties are closed under addition and affine composition, and have a straightforward representation calculus.

Quadratic Support Functions

$$\rho(U, b, B, M; y) = \sup_{u \in U} \left\{ \langle u, b + By \rangle - \frac{1}{2} \langle u, Mu \rangle \right\}$$

Relax the assumption that U is polyhedral, and let U be an arbitrary closed convex set containing the origin.

Quadratic Support Functions

$$\rho(U, b, B, M; y) = \sup_{u \in U} \left\{ \langle u, b + By \rangle - \frac{1}{2} \langle u, Mu \rangle \right\}$$

Relax the assumption that U is polyhedral, and let U be an arbitrary closed convex set containing the origin.
This class contains

- All PLQ penalties (obviously).
- Support functions (let $M = 0$) to all convex sets containing the origin. In particular, we get all norms and gauges.
- If $M = LL^T$ with $\operatorname{rank}(L) = k$,

$$\rho(U, 0, I, M; y) = \inf_{s \in \mathbb{R}^k} \left[\frac{1}{2} \|s\|_2^2 + \gamma \left(y - Ls \mid U^\circ \right) \right] .$$

- If M^{-1} exists,

$$\rho(U, 0, I, M; y) = \frac{1}{2} \|P_M(M^{-1}y| \, U)\|_M^2 + \left\langle M^{-1}y - P_M(M^{-1}y| \, U), \, P_M(M^{-1}y| \, U) \right\rangle_M .$$

- ρ is the negative log-likelihood of a density with known mean and variance if

$$[B^T \operatorname{cone}(U)]^\circ = \{0\} .$$

Generalized Huber and Vapnik loss functions

Generalized Huber

Given covariance matrix V, take $M = V^{-1}$, and $U = \kappa \mathbb{B}_M$:

$$\rho(y) = \begin{cases} \frac{1}{2}\|y\|_M^2, & \text{if } \|y\|_M \leq \kappa \\ \kappa\|y\|_M - \frac{\kappa^2}{2}, & \text{if } \|y\|_M > \kappa \ . \end{cases}$$

Generalized Vapnik

$K \subset \mathbb{R}^n$ be a non-empty symmetric convex cone ($K^\circ = -K$).

$w <_K v \iff v - w \in \mathrm{intr}(K)$.

Set

$$U = (\mathbb{B}^\circ \cap K) \times (\mathbb{B}^\circ \cap K^\circ), \quad M = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad b = -\begin{pmatrix} v \\ w \end{pmatrix}, \quad \text{and} \quad B = \begin{bmatrix} I \\ I \end{bmatrix} .$$

Then

$$\rho(y) = \mathrm{dist}\left(y \mid [w, v]_K\right),$$

where $[w, v]_K$ is the order interval $\{y \mid w \leq_K y \leq_K v\}$.

Taking $\|\cdot\| = \|\cdot\|_1$, $K = \mathbb{R}^n_+$, and $v = \epsilon \mathbf{1}$=-w, returns the multivariate Vapnik loss function

PLQ Optimization

Consider now the minimization problem

$$\min_y \rho(c, C, b, B, M; y) \quad \text{s.t. } Ay \leq a.$$

Introduce slack variables s and r:

$$Cu + s = c, \quad Ay + r = a\ .$$

Let q, w be dual variables corresponding to these constraints.

PLQ Optimization

Consider now the minimization problem

$$\min_y \rho(c, C, b, B, M; y) \quad \text{s.t. } Ay \le a.$$

Introduce slack variables s and r:

$$Cu + s = c, \quad Ay + r = a \; .$$

Let q, w be dual variables corresponding to these constraints. The KKT system is given by

$$\begin{array}{rcl} 0 & = & B^T u + A^T w \\ 0 & = & By - Mu - C^T q + b \\ 0 & = & Cu + s - c \\ 0 & = & Ay + r - a \\ 0 & = & q_i s_i \; \forall i \; , \; q, s \ge 0 \\ 0 & = & w_i r_i \; \forall i \; , \; w, r \ge 0 \; . \end{array}$$

We have an interior point toolbox to work directly with such KKT systems available through *github/saravkin/ipSolver*.

Code and performance

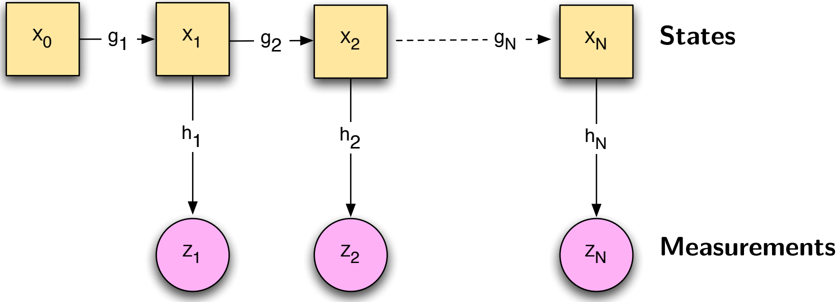
We compared the IP approach with ADMM for a small set of test problems. We used Stephen Boyd's Lasso implementation, and wrote code for the other examples following this template.

Problem	AD Iter	AD Inner	IP Iter	t_{AD} (s)	t_{IP} (s)	ObjDiff
Lasso						
$A : 1500 \times 5000$	15	—	18	2.0	58.3	0.0025
SVM						
$\kappa(A) = 7.7 \times 10^{10}$						
$A : 32561 \times 123$	653	—	77	41.2	23.9	0.17
Huber Lasso						
ADMM/ADMM						
$\kappa(A) = 5.8;\ A : 1000 \times 2000$	26	100	20	14.1	10.5	0.00006
$\kappa(A) = 1330;\ A : 1000 \times 2000$	27	100	24	40.0	13.0	0.0018
ADMM/L-BFGS						
$\kappa(A) = 5.8;\ A : 1000 \times 2000$	18	—	20	2.8	10.3	1.02
$\kappa(A) = 1330;\ A : 1000 \times 2000$	22	—	24	21.2	13.1	1.24
L1 Lasso						
ADMM/ADMM						
$\kappa(A) = 2.2;\ A : 500 \times 2000$	104	100	29	57.4	5.9	0.06
$\kappa(A) = 1416;\ A : 500 \times 2000$	112	100	29	81.4	5.6	0.21

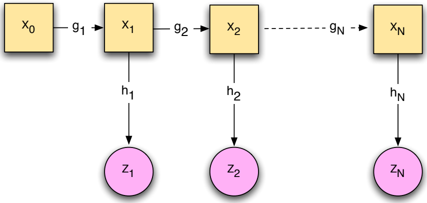
PLQ Kalman Smoothing

Graphical Overview of Dynamic Systems

- Goal: to obtain estimates on states $\{x_k\}$ given measurements $\{z_k\}$
- State evolution models $x_k = g_k(x_{k-1}) + w_k$.
- Initialization: $x_1 = x_0 + w_1$.
- Measurement model: $z_k = h_k(x_k) + v_k$



PLQ Smoothers



- We consider the entire class of PLQ smoothers $\begin{bmatrix} \mu = g(x) - w \\ z = h(x) + v \end{bmatrix}$, where both w and v PLQ densities.
- When $g(x) = Gx$ and $h(x) = Hx$ are linear, this corresponds to the optimization problem

$$\min_{x} \rho_w[\mu - Gx] + \rho_v[z - Hx] \ .$$

where ρ_w and ρ_v are PLQ penalties.

Block tridiagonal systems

In the classic formulation, ρ_w, ρ_v are quadratics, and above objective reduces to

$$\min_x \|[\mu - Gx]\|^2_{Q^{-1}} + \|[z - Hx]\|^2_{R^{-1}} \ .$$

$$G = \begin{bmatrix} I & 0 & & \\ -G_2 & I & \ddots & \\ & \ddots & \ddots & 0 \\ & & -G_N & I \end{bmatrix}, \quad H = \begin{bmatrix} H_1 & & \\ & \ddots & \\ & & H_N \end{bmatrix}$$

To recover x, we must solve a system of form

$$\left(G^T Q^{-1} G + H^T R^{-1} H\right) x = r.$$

This system is **block tridiagonal positive definite**, and can be solved in $O(n^3 N)$ operations.

Block tridiagonal systems

In the classic formulation, ρ_w, ρ_v are quadratics, and above objective reduces to

$$\min_x \|[\mu - Gx]\|^2_{Q^{-1}} + \|[z - Hx]\|^2_{R^{-1}} \ .$$

$$G = \begin{bmatrix} I & 0 & & \\ -G_2 & I & \ddots & \\ & \ddots & \ddots & 0 \\ & & -G_N & I \end{bmatrix}, \quad H = \begin{bmatrix} H_1 & & \\ & \ddots & \\ & & H_N \end{bmatrix}$$

To recover x, we must solve a system of form

$$\left(G^T Q^{-1} G + H^T R^{-1} H\right) x = r.$$

This system is **block tridiagonal positive definite**, and can be solved in $O(n^3 N)$ operations.

For **any PLQ** ρ_w, ρ_v, the general IP approach preserves the structure of the problem, and inherits the $O(n^3 N)$ efficiency *per iteration*.

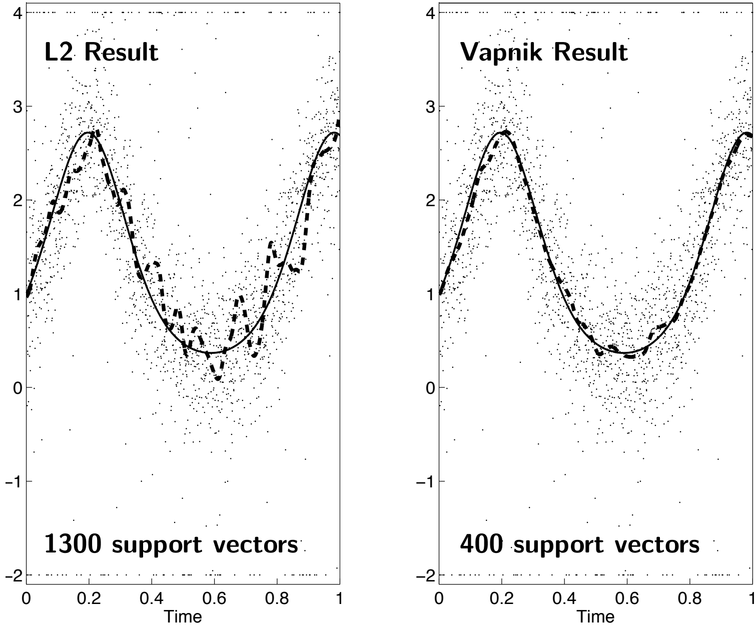
Functional recovery

- Goal: to recover a representation of $\exp(8\sin(t))$ from noisy measurements.
- Process model: integrated brownian noise. For $\Delta t = 1/2000$,

$$G_k(x_{k-1}) = \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} x_{k-1}\,, \quad Q_k = \lambda^2 \begin{bmatrix} \Delta t & \Delta t^2/2 \\ \Delta t^2/2 & \Delta t^3/3 \end{bmatrix}\,.$$

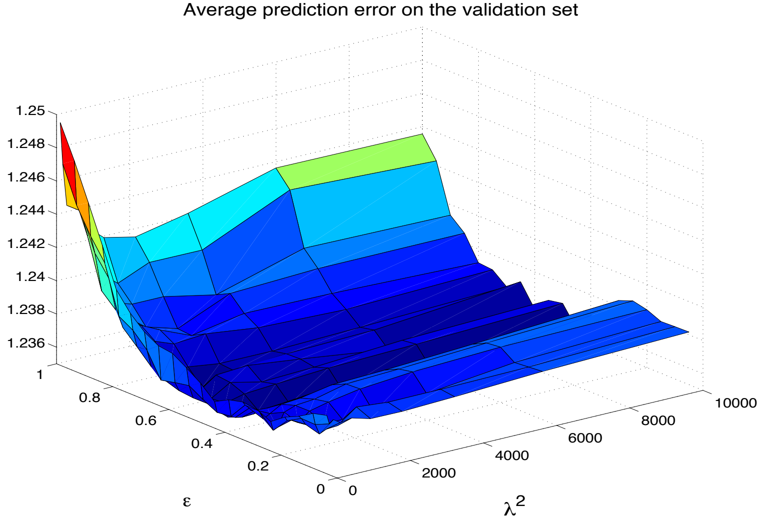
  where λ^2 is an unknown scale factor to be estimated from the data by cross-validation (efficiency essential!)
- Direct observation of function values: $H_k(x_k) = [0 \;\; 1]x_k$.
- In the smoother, we model w as Gaussian, and v as Vapnik with unknown ϵ (also estimated by cross-validation).
- Vapnik plays two important roles:
  - Measurements are contaminated by large $N(0, 25)$ outliers and
  - The function we recover has a sparser representation in terms of the data, since only 'active' data points are used to evaluate the function.

Functional Recovery Results



Training and cross validation for parameter selection

200 mesh points each with 1300 training and 700 validation points.
"Optimal" $\mathrm{L}_2 + \rho_V$ fitting values $\lambda^2 = 2.15 \times 10^3$ and $\epsilon = 0.45$.



Sparse and Robust PLQ Regression

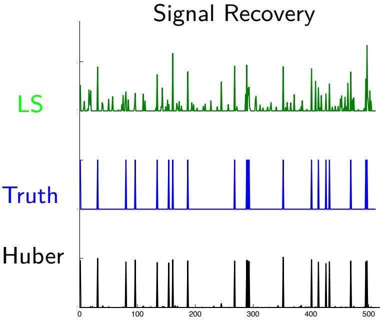
$$\text{HBP}_\sigma: \quad \min_{0 \le x} \quad \|x\|_1 \quad \text{st} \quad \rho(b - Ax) \le \boldsymbol{\sigma}$$

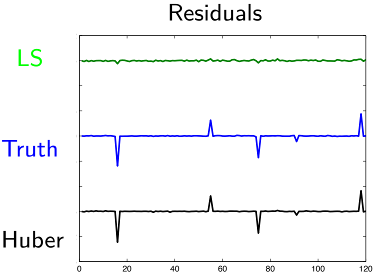
Problem Specification

- x 20-sparse spike train in $\mathbb{R}^{512}_{+}$
- b measurements in $\mathbb{R}^{120}$
- A Measurement matrix satisfying RIP
- ρ Huber function
- σ error level set at $.01$
- 5 outliers

Results

In the presence of outliers, the robust formulation recovers the spike train, while the standard formulation does not.





Thank you!

- **Papers:**
  - A.Y. Aravkin, J.V. Burke, G. Pillonetto, *Sparse/Robust Estimation and Kalman Smoothing with Nonsmooth Log-Concave Densities: Modeling, Computation, and Theory*, to appear in the Journal of Machine Learning, 2013.
  - A.Y. Aravkin, J.V. Burke, G. Pillonetto, *System Identification with PLQ Penalties*, to appear in Conference on Decision and Control Proceedings 2013.
- **Software:**
  - **CKBS**, (Robust & constrained Kalman smoothing). https://projects.coin-or.org/CoinBazaar/wiki/Projects/ckbs
  - **IPsolver**: github/saravkin/IPsolver.