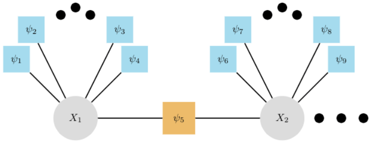
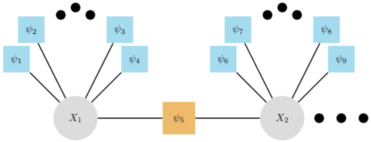
An Overview of Robust Optimization Techniques

The State Estimation Problem



$$\hat{X} = \underset{X}{\operatorname{argmax}} \prod_{n=1}^{N} \psi_n(A_n, B_n).$$

The State Estimation Problem

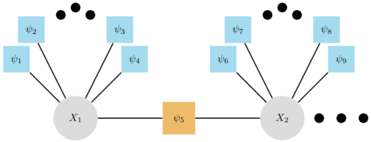


$$\hat{X} = \underset{X}{\operatorname{argmax}} \prod_{n=1}^{N} \psi_n(A_n, B_n).$$

Gaussian Assumption

In practice, this state estimation problem is simplified through the assumption that each factor adheres to a Gaussian uncertainty model (i.e., $\psi_n(A_n, B_n) \sim \mathcal{N}(\mu_n, \Lambda_n)$)

The State Estimation Problem



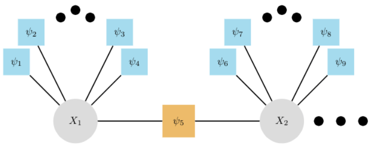
$$\hat{X} = \underset{X}{\text{argmax}} \prod_{n=1}^{N} \psi_n(A_n, B_n).$$

Gaussian Assumption

In practice, this state estimation problem is simplified through the assumption that each factor adheres to a Gaussian uncertainty model (i.e., $\psi_n(A_n, B_n) \sim \mathcal{N}(\mu_n, \Lambda_n)$)

$$\hat{X} = \underset{X}{\text{argmin}} \sum_{n=1}^{N} || \ r_n(X) \ ||_{\Lambda_n} \quad \text{s.t.} \quad r_n(X) \triangleq y_n - h_n(X),$$

The State Estimation Problem



$$\hat{X} = \underset{X}{\text{argmax}} \prod_{n=1}^{N} \psi_n(A_n, B_n).$$

Gaussian Assumption

In practice, this state estimation problem is simplified through the assumption that each factor adheres to a Gaussian uncertainty model (i.e., $\psi_n(A_n, B_n) \sim \mathcal{N}(\mu_n, \Lambda_n)$)

$$\hat{X} = \underset{X}{\text{argmin}} \sum_{n=1}^{N} || \; r_n(X) \; ||_{\Lambda_n} \quad \text{s.t.} \quad r_n(X) \triangleq y_n - h_n(X),$$

Factor Graph Optimization

In this form, we can utilize any N.L.L.S. optimizer to generate $\hat{X}$ estimate.

- Gauss-Newton
- Powell's dogleg
- Levenberg–Marquardt

$\hat{X}$ Estimation Under Non-Ideal Conditions

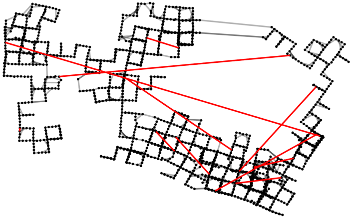
What are non-ideal conditions?

- Optimization with faulty sensors (i.e., optimization with sporadic erroneous measurements).
- Optimization with poor *a priori covariance estimates (i.e., optimization with unknown noise characteristics).*

$\hat{X}$ Estimation Under Non-Ideal Conditions

What are non-ideal conditions?

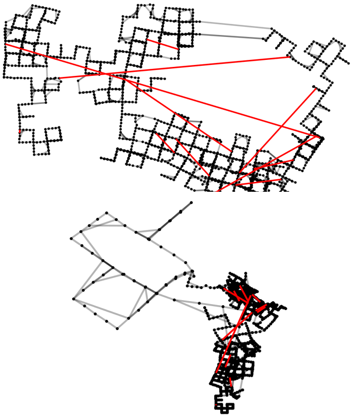
- Optimization with faulty sensors (i.e., optimization with sporadic erroneous measurements).
- Optimization with poor *a priori covariance estimates (i.e., optimization with unknown noise characteristics).*



$\hat{X}$ Estimation Under Non-Ideal Conditions

What are non-ideal conditions?

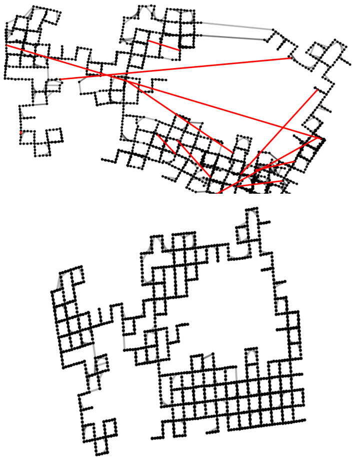
- Optimization with faulty sensors (i.e., optimization with sporadic erroneous measurements).
- Optimization with poor *a priori covariance estimates (i.e., optimization with unknown noise characteristics).*



$\hat{X}$ Estimation Under Non-Ideal Conditions

What are non-ideal conditions?

- Optimization with faulty sensors (i.e., optimization with sporadic erroneous measurements).
- Optimization with a poor *a prior covariance estimates (i.e., optimization with unknown noise characteristics).*



LS estimators are not robust

- From the previous slide, we can see that estimators using the LS criterion are not **robust**.
  - Very sensitive to inaccuracies in the dataset characterization
  - Very sensitive to misspecification of the underlying models
- This issue arises because the estimator is an unbounded function of the residuals.

- Within this slide deck, we'll more concretely define what makes an estimator robust & provide specific examples of robust estimators.
  - Looking for estimators that provide high efficiency & a large breakdown point.
  - Robust estimation is about finding the right compromise between those two quantities.

Background for Robust Estimation

$$\mathscr{L}(z;X) = \frac{1}{2}\sum_{n}^{N}\rho(||z_n - h_n(X)||^2)$$

$$J(z) = \underset{X}{\operatorname{argmin}}\,\mathscr{L}(z,X)$$

Here, $J(z)$ is a function that maps from observations to a solution (i.e., an estimation of the state space).

Given this mapping, we can define the bias of an estimator as

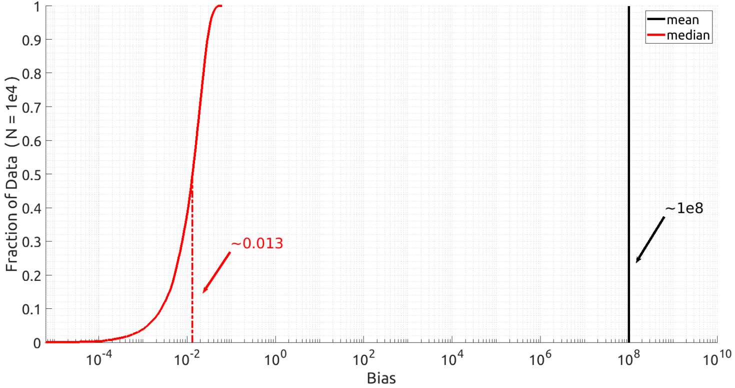
$$\boxed{\mathrm{B}_X(J) = \mathrm{E}_z[J(Z)] - X}$$

Estimator Bias

- One way to quantify the robustness of an estimator is via the induced bias when erroneous observations are included.
  - Where the bias is the systematic difference between the estimator and the true distribution.

Background for Robust Estimation

- Let's look at two simple estimators (the mean and the median) with very different characteristics w.r.t., robust estimation.
  - Simulate a dataset as $Z \sim (1.0 - 0.01)\mathcal{N}(0,1) + 0.01\mathcal{N}(1e^{10},1)$
  - We'll use the mean/median to attempt to recover the first moment of the uncontaminated distribution.



Background for Robust Estimation

Max Bias Function

- Given the bias function defined on the previous slide, we can now define the maximum bias function
  - The maximum of $||B_X(J)||^2$ for any error distribution.

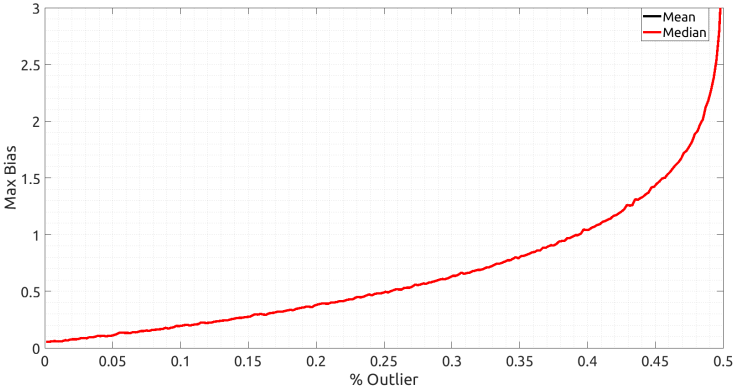
Say our observations are contaminated with a percentage ($\epsilon < 50\%$) of outliers, i.e.,

$$Z \sim (1-\epsilon)A + \epsilon B$$

Then, we can define the max possible bias for a given proportion of outliers as:

$$\boxed{\text{MB}(\epsilon, J) = \sup_b\{||B_X(J(Z))|| : Z\}}$$

Background for Robust Estimation



Background for Robust Estimation

The Breakdown Point

- Given the max bias function defined on the previous slide, we can now define the breakdown point of an estimator.
  - This tells us the maximum proportions of outliers that the estimator can handle without an arbitrarily large bias.
  - The metric for quantifying robustness in this presentation.
- As we'll see on the next several slides, commonly used estimators have very different levels of robustness (w.r.t. estimator breakdown)

$$\boxed{\mathrm{BP}(J) = \sup_\epsilon\{\epsilon : \mathrm{MB}(\epsilon, J) < \infty\}}$$

Background for Robust Estimation

Say we're given a set of data, $Z \sim (1-\epsilon)A + \epsilon B$, where $0 \geq \epsilon < \frac{1}{2}$

- What's the sensitivity to the magnitude of ϵ for two commonly used estimators?

Background for Robust Estimation

Say we're given a set of data, $Z \sim (1-\epsilon)A + \epsilon B$, where $0 \geq \epsilon < \frac{1}{2}$

- What's the sensitivity to the magnitude of ϵ for two commonly used estimators?

The Sample Mean

$$\mu = \frac{1}{n}\sum_{n}^{N} z_n = \frac{z_1 + ... + z_N}{N}$$

- Let $z_1 ..., z_{N-1} \sim A$ and let a single observation approach infinity $z_N \to \infty$.
- In this scenario, $\mu \to \infty$, which shows that the sample mean has a breakdown point of zero.
- So, a single large outlier breaks the sample mean.

Background for Robust Estimation

Say we're given a set of data, $Z \sim (1-\epsilon)A + \epsilon B$, where $0 \geq \epsilon < \frac{1}{2}$

- What's the sensitivity to the magnitude of ϵ for two commonly used estimators?

The Sample Mean

$$\mu = \frac{1}{n}\sum_{n}^{N} z_n = \frac{z_1 + ... + z_N}{N}$$

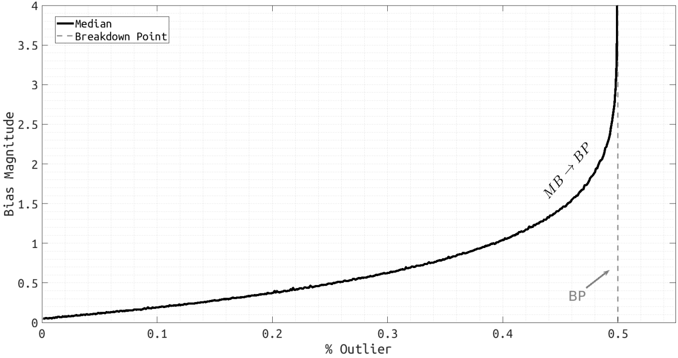
- Let $z_1 ..., z_{N-1} \sim A$ and let a single observation approach infinity $z_N \rightarrow \infty$.
- In this scenario, $\mu \rightarrow \infty$, which shows that the sample mean has a breakdown point of zero.
- So, a single large outlier breaks the sample mean.

The Sample Median

- let $m = \lfloor (N-1)/2 \rfloor$
- Then, we let $z_1 ..., z_m \sim A$ and let a minority of the observation approach infinity $z_{m+1}, ..., z_N \rightarrow \infty$.
- In this scenario, the median will stay with the majority of the observations, which shows that the median has an asymptotic breakdown point of $\frac{1}{2}$

Background for Robust Estimation

- We can also utilize the max bias curve to extract the breakdown point
  - The max bias curve asymptotically approaches the breakdown point



Background for Robust Estimation

Efficiency

Estimator Efficiency

- The efficiency of an estimator is the ratio of the minimal sample variance to the actual variance of the estimator

$$eff\left(J(z)\right) = \frac{1/I(X)}{\text{var}(J(z))}$$

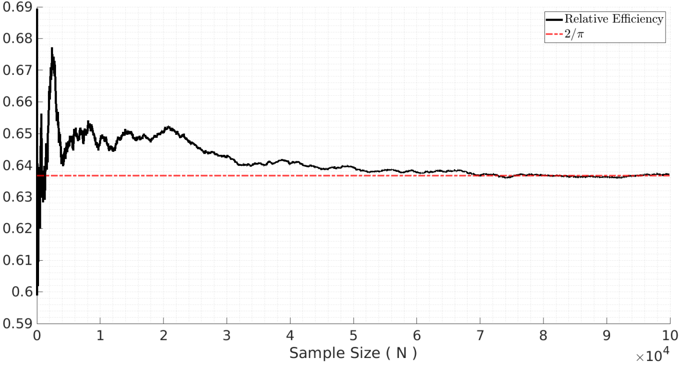
Here, $I(X)$ is the Fisher information – the reciprocal of the Fisher information is the Cramer-Rao Lower Bound (CRLB).

If we're interested in comparing two estimators, we can utilize the relative efficiency (the CRLB drops out)

$$\boxed{\text{eff}_{rel.}(J_1, J_2) = \frac{\text{eff}\left(J1(z)\right)}{\text{eff}\left(J2(z)\right)}}$$

Background for Robust Estimation

$$\text{eff}_{rel.}(\text{Mean}, \text{Median}) = \frac{\text{eff}(\text{median})}{\text{eff}(\text{mean})} = \frac{\text{var}(mean)}{\text{var}(median)} \sim \frac{2}{\pi}$$



Literature Review

ℓ^2-norm breakdown

let any arbitrary observation, z_n, significantly deviate from the model (i..e., $||z_n - h_n(x_n)||^2_{\Lambda_n} \rightarrow \infty$), which, in turn, will cause an arbitrarily large bias in the estimate

Literature Review

ℓ^2-norm breakdown

let any arbitrary observation, z_n, significantly deviate from the model (i..e., $||z_n - h_n(x_n)||^2_{\Lambda_n} \to \infty$), which, in turn, will cause an arbitrarily large bias in the estimate

Data Weighting Methods

- robust maximum likelihood estimators (m-estimators) [6]
- lifted optimization (switchable constraints) [12, 10]
- dynamic covariance scaling (DCS) [2]
- max mixtures (MM) [9]

Data Exclusion Methods

- random sample consensus (RANSAC) [5]
- realizing, reversing, recovering (RRR) [8]
- ℓ^1 relaxation [3]
- receiver autonomous integrity monitoring (RAIM) [11]

Literature Review

ℓ^2-norm breakdown

let any arbitrary observation, z_n, significantly deviate from the model (i..e., $||z_n - h_n(x_n)||^2_{\Lambda_n} \to \infty$), which, in turn, will cause an arbitrarily large bias in the estimate

Data Weighting Methods

- robust maximum likelihood estimators (m-estimators)
- lifted optimization (switchable constraints)
- dynamic covariance scaling (DCS)
- max mixtures (MM)

- We will focus on data weighting methods for this application.
- You can view data exclusion methods as an extreme version of data weighting (i.e., the weight is binary – 0 if excluded and 1 if included)

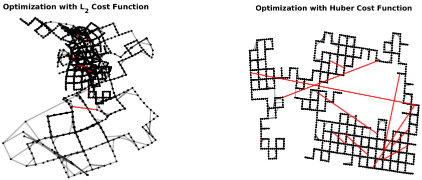
Robust Optimization Techniques: M-Estimators

Main Idea

De-weight observations based on their deviation from the assumed model.

$$\hat{X} = \underset{X}{\operatorname{argmin}} \sum_{n=1}^{N} \rho\left(\|r_n(X)\|^2_{\Lambda_n} \right)$$

$$= \underset{X}{\operatorname{argmin}} \sum_{n=1}^{N} \rho(e_n) \quad \text{s.t.} \quad e_n \triangleq \|r_n(X)\|^2_{\Lambda_n}$$

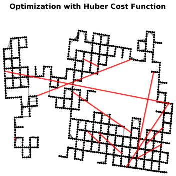


Robust Optimization Techniques: M-Estimators

Main Idea

De-weight observations based on their deviation from the assumed model.

$$\hat{X} = \operatorname*{argmin}_{X} \sum_{n=1}^{N} \rho\left(\|r_n(X)\|^2_{\Lambda_n} \right)$$
$$= \operatorname*{argmin}_{X} \sum_{n=1}^{N} \rho(e_n) \quad \text{s.t.} \quad e_n \triangleq \|r_n(X)\|^2_{\Lambda_n}$$



M-Estimator Definitions

$\rho(x)$ is the robust cost function,
$\psi(x) \triangleq \frac{\partial \rho(x)}{\partial x}$ is the influence function,
$w(x) \triangleq \frac{\psi(x)}{x}$ is the weighting function

Robust Optimization Techniques: M-Estimators

Main Idea

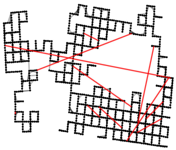
De-weight observations based on their deviation from the assumed model.

$$\hat{X} = \underset{X}{\operatorname{argmin}} \sum_{n=1}^{N} \rho\left(\|r_n(X)\|^2_{\Lambda_n} \right)$$

$$= \underset{X}{\operatorname{argmin}} \sum_{n=1}^{N} \rho(e_n) \quad \text{s.t.} \quad e_n \triangleq \|r_n(X)\|^2_{\Lambda_n}$$

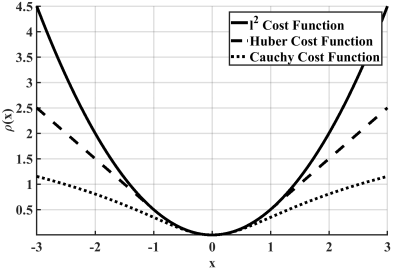
Optimization with L_2 Cost Function

Optimization with Huber Cost Function



M-Estimator Definitions

$\rho(x)$ is the robust cost function,
$\psi(x) \triangleq \frac{\partial \rho(x)}{\partial x}$ is the influence function,
$w(x) \triangleq \frac{\psi(x)}{x}$ is the weighting function



Robust Optimization Techniques: M-Estimators

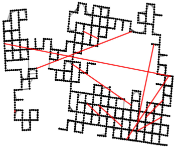
Main Idea

De-weight observations based on their deviation from the assumed model.

$$\hat{X} = \underset{X}{\operatorname{argmin}} \sum_{n=1}^{N} \rho\left(||r_n(X)||^2_{\Lambda_n} \right)$$
$$= \underset{X}{\operatorname{argmin}} \sum_{n=1}^{N} \rho(e_n) \quad \text{s.t.} \quad e_n \triangleq ||r_n(X)||^2_{\Lambda_n}$$

Optimization with L_2 Cost Function

Optimization with Huber Cost Function



M-Estimator Definitions

$\rho(x)$ is the robust cost function,
$\psi(x) \triangleq \frac{\partial \rho(x)}{\partial x}$ is the influence function,
$w(x) \triangleq \frac{\psi(x)}{x}$ is the weighting function

Covariance Adaptation

$$\frac{\partial J}{\partial X} = \sum_{n=1}^{N} \frac{\partial \rho}{\partial e_n} \frac{\partial e_n}{\partial r_n} \frac{\partial r_n}{\partial X}$$
$$= r_n(X)^T \left[\frac{1}{e_n} \left. \frac{\partial \rho}{\partial e} \right|_{e_n(X)} \Lambda_n^{-1} \right] \frac{\partial r_n}{\partial X}$$
$$= r_n(X)^T \left[w\left(e_n\right) \Lambda_n^{-1} \right] \frac{\partial r_n}{\partial X}$$

Robust Optimization Techniques: M-Estimators

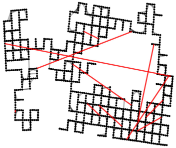
Main Idea

De-weight observations based on their deviation from the assumed model.

$$
\begin{aligned}
\hat{X} &= \underset{X}{\operatorname{argmin}} \sum_{n=1}^{N} \rho\left(\, ||r_n(X)||^2_{\Lambda_n} \, \right) \\
&= \underset{X}{\operatorname{argmin}} \sum_{n=1}^{N} \rho(e_n) \quad \text{s.t.} \quad e_n \triangleq ||r_n(X)||^2_{\Lambda_n}
\end{aligned}
$$

Optimization with L_2 Cost Function

Optimization with Huber Cost Function



M-Estimator Definitions

$\rho(x)$ is the robust cost function,
$\psi(x) \triangleq \frac{\partial \rho(x)}{\partial x}$ is the influence function,
$w(x) \triangleq \frac{\psi(x)}{x}$ is the weighting function

Covariance Adaptation

$$
\begin{aligned}
\frac{\partial J}{\partial X} &= \sum_{n=1}^{N} \frac{\partial \rho}{\partial e_n} \frac{\partial e_n}{\partial r_n} \frac{\partial r_n}{\partial X} \\
&= r_n(X)^T \left[\frac{1}{e_n} \left. \frac{\partial \rho}{\partial e} \right|_{e_n(X)} \Lambda_n^{-1} \right] \frac{\partial r_n}{\partial X} \\
&= r_n(X)^T \left[w\left(e_n\right) \Lambda_n^{-1} \right] \frac{\partial r_n}{\partial X}
\end{aligned}
$$

$$
\hat{X} = \underset{X}{\operatorname{argmin}} \sum_{n=1}^{N} w_n(e_n) \, || \, r_n(X) \, ||_{\Lambda_n}
$$

Robust Optimization Techniques: M-Estimators

- Commonly used m-estimators.
  - Ref: `http://www-sop.inria.fr/odyssee/software/old_robotvis/Tutorial-Estim/node24.html`

type	$\rho(x)$	$\psi(x)$	$w(x)$
L_2	$x^2/2$	x	1
L_1	$\lvert x\rvert$	$\mathrm{sgn}(x)$	$\frac{1}{\lvert x\rvert}$
$L_1 - L_2$	$2(\sqrt{1+x^2/2}-1)$	$\frac{x}{\sqrt{1+x^2/2}}$	$\frac{1}{\sqrt{1+x^2/2}}$
L_p	$\frac{\lvert x\rvert^\nu}{\nu}$	$\mathrm{sgn}(x)\lvert x\rvert^{\nu-1}$	$\lvert x\rvert^{\nu-2}$
"Fair"	$c^2[\frac{\lvert x\rvert}{c} - \log(1+\frac{\lvert x\rvert}{c})]$	$\frac{x}{1+\lvert x\rvert/c}$	$\frac{1}{1+\lvert x\rvert/c}$
Huber $\begin{cases}\text{if } \lvert x\rvert \le k \\ \text{if } \lvert x\rvert \ge k\end{cases}$	$\begin{cases} x^2/2 \\ k(\lvert x\rvert - k/2)\end{cases}$	$\begin{cases} x \\ k\,\mathrm{sgn}(x)\end{cases}$	$\begin{cases} 1 \\ k/\lvert x\rvert\end{cases}$
Cauchy	$\frac{c^2}{2}\log(1+(x/c)^2)$	$\frac{x}{1+(x/c)^2}$	$\frac{1}{1+(x/c)^2}$
Geman-McClure	$\frac{x^2/2}{1+x^2}$	$\frac{x}{(1+x^2)^2}$	$\frac{1}{(1+x^2)^2}$
Welsch	$\frac{c^2}{2}[1-\exp(-(x/c)^2)]$	$x\exp(-(x/c)^2)$	$\exp(-(x/c)^2))$
Tukey $\begin{cases}\text{if } \lvert x\rvert \le c \\ \text{if } \lvert x\rvert > c\end{cases}$	$\begin{cases} \frac{c^2}{6}(1-[1-(x/c)^2]^3) \\ (c^2/6)\end{cases}$	$\begin{cases} x[1-(x/c)^2]^2 \\ 0\end{cases}$	$\begin{cases} [1-(x/c)^2]^2 \\ 0\end{cases}$

Robust Optimization Techniques: M-Estimators

- Commonly used m-estimators.
  - Ref: `http://www-sop.inria.fr/odyssee/software/old_robotvis/Tutorial-Estim/node24.html`

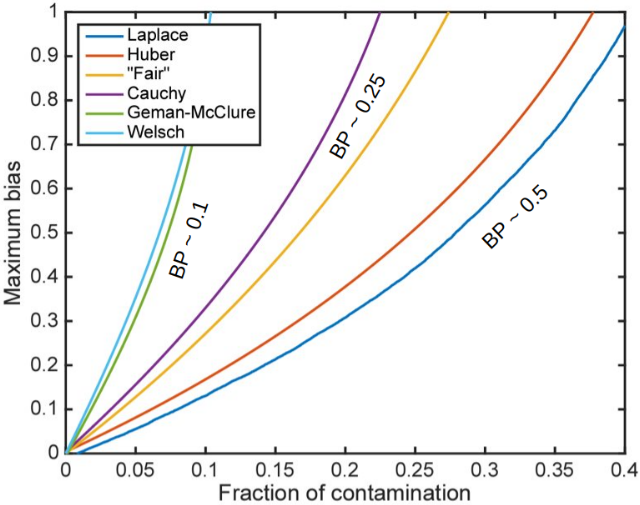
type	$\rho(x)$	$\psi(x)$	$w(x)$
L_2	$x^2/2$	x	1
L_1	$\lvert x\rvert$	$\mathrm{sgn}(x)$	$\frac{1}{\lvert x\rvert}$
$L_1 - L_2$	$2(\sqrt{1+x^2/2}-1)$	$\frac{x}{\sqrt{1+x^2/2}}$	$\frac{1}{\sqrt{1+x^2/2}}$
L_p	$\frac{\lvert x\rvert^\nu}{\nu}$	$\mathrm{sgn}(x)\lvert x\rvert^{\nu-1}$	$\lvert x\rvert^{\nu-2}$
"Fair"	$c^2[\frac{\lvert x\rvert}{c} - \log(1+\frac{\lvert x\rvert}{c})]$	$\frac{x}{1+\lvert x\rvert/c}$	$\frac{1}{1+\lvert x\rvert/c}$
Huber $\begin{cases}\text{if } \lvert x\rvert \le k \\ \text{if } \lvert x\rvert \ge k\end{cases}$	$\begin{cases}x^2/2 \\ k(\lvert x\rvert - k/2)\end{cases}$	$\begin{cases}x \\ k\,\mathrm{sgn}(x)\end{cases}$	$\begin{cases}1 \\ k/\lvert x\rvert\end{cases}$
Cauchy	$\frac{c^2}{2}\log(1+(x/c)^2)$	$\frac{x}{1+(x/c)^2}$	$\frac{1}{1+(x/c)^2}$
Geman-McClure	$\frac{x^2/2}{1+x^2}$	$\frac{x}{(1+x^2)^2}$	$\frac{1}{(1+x^2)^2}$
Welsch	$\frac{c^2}{2}[1-\exp(-(x/c)^2)]$	$x\exp(-(x/c)^2)$	$\exp(-(x/c)^2))$
Tukey $\begin{cases}\text{if } \lvert x\rvert \le c \\ \text{if } \lvert x\rvert > c\end{cases}$	$\begin{cases}\frac{c^2}{6}(1-[1-(x/c)^2]^3) \\ (c^2/6)\end{cases}$	$\begin{cases}x[1-(x/c)^2]^2 \\ 0\end{cases}$	$\begin{cases}[1-(x/c)^2]^2 \\ 0\end{cases}$

- Barron recently published a generalized robust loss function that encapsulates most m-estimators.
  - Barron, Jonathan T. "A general and adaptive robust loss function." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

$$\rho(x,\alpha,c) = \frac{|\alpha-2|}{\alpha}\left(\left(\frac{(\frac{x}{c})^2}{|\alpha-2|}+1\right)^{\alpha/2}-1\right)$$

- You can get the weighting function via $w(x,\alpha,c) = \frac{1}{x}\frac{\partial\rho}{\partial x}(x,\alpha,c)$
  - beware of singularities at $\alpha = \{0, 2\}$

Robust Optimization Techniques: **M-Estimators**



Robust Optimization Techniques: Lifted Optimization

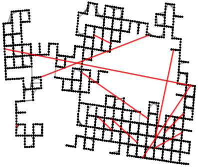
Main Idea

Reducing the influence of erroneous observations by augmenting the optimization problem by concurrently solving for the desired states and a set of measurement weighting value.

$$\hat{X}, \hat{S} = \sum_{n=1}^{N} \left[\| \ \psi(s_n) r_n(X) \ \|_{\Lambda_n}^2 + \| \ \gamma_n - s_n \ \|_{\Xi_n} \right]$$

Optimization with L_2 Cost Function

Optimization with Switch Constraint Function



Robust Optimization Techniques: Lifted Optimization

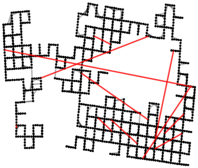
Main Idea

Reducing the influence of erroneous observations by augmenting the optimization problem by concurrently solving for the desired states and a set of measurement weighting value.

$$\hat{X}, \hat{S} = \sum_{n=1}^{N} \left[\| \psi(s_n) r_n(X) \|_{\Lambda_n}^2 + \| \gamma_n - s_n \|_{\Xi_n} \right]$$

Optimization with L_2 Cost Function

Optimization with Switch Constraint Function



Covariance Adaptation

Let's evaluate the effect that switchable constraints will have on measurement constraint in the modified cost function.

$$\| \psi(s_i) r_i(X) \|_{\Lambda_i} = \left[w_i r_i(X) \right]^T \Lambda_i^{-1} \left[w_i r_i(X) \right]$$

where $w_i \triangleq \psi(s_i)$.
Now, by noting that w_j is a scalar, we have

$$\| \psi(s_i) r_i(X) \|_{\Lambda_i} = r_i(X)^T \left[w_i^2 \Lambda_i^{-1} \right] r_i(X)$$

$$= r_i(X)^T \hat{\Lambda}_i^{-1} r_i(X)$$

Robust Optimization Techniques: **Dynamic Covariance Scaling**

Main Idea

- Dynamic Covariance Scaling [1] was introduced as an approximate closed-form solution to switchable constraint.
- The DCS approach reduces the confidence of an observable by de-weighting the corresponding element in the information matrix by the scale factor.

Robust Optimization Techniques: Dynamic Covariance Scaling

Main Idea

- Dynamic Covariance Scaling [1] was introduced as an approximate closed-form solution to switchable constraint.
- The DCS approach reduces the confidence of an observable by de-weighting the corresponding element in the information matrix by the scale factor.

$$w(x) = \begin{cases} 1, & \text{if } x^2 \leq k. \\ \frac{4k^2}{(x^2+k)^2}, & \text{otherwise,} \end{cases}$$

Robust Optimization Techniques: Dynamic Covariance Scaling

Main Idea

- Dynamic Covariance Scaling [1] was introduced as an approximate closed-form solution to switchable constraint.
- The DCS approach reduces the confidence of an observable by de-weighting the corresponding element in the information matrix by the scale factor.

M-Estimator Definitions

$\rho(x)$ is the robust cost function,
$\psi(x) \triangleq \frac{\partial \rho(x)}{\partial x}$ is the influence function,
$w(x) \triangleq \frac{\psi(x)}{x}$ is the weighting function

$$\boxed{w(x) = \begin{cases} 1, & \text{if } x^2 \leq k. \\ \frac{4k^2}{(x^2+k)^2}, & \text{otherwise,} \end{cases}}$$

Robust Optimization Techniques: Dynamic Covariance Scaling

Main Idea

- Dynamic Covariance Scaling [1] was introduced as an approximate closed-form solution to switchable constraint.
- The DCS approach reduces the confidence of an observable by de-weighting the corresponding element in the information matrix by the scale factor.

$$w(x) = \begin{cases} 1, & \text{if } x^2 \leq k. \\ \frac{4k^2}{(x^2+k)^2}, & \text{otherwise,} \end{cases}$$

M-Estimator Definitions

$\rho(x)$ is the robust cost function,
$\psi(x) \triangleq \frac{\partial \rho(x)}{\partial x}$ is the influence function,
$w(x) \triangleq \frac{\psi(x)}{x}$ is the weighting function

Covariance Adaptation

$$\rho(x) = \int x \; w(x)$$

$$\rho(x) = \begin{cases} \frac{1}{2}x^2, & \text{if } x^2 \leq k. \\ \frac{k\left(3x^2 - k\right)}{2\left(x^2+k\right)}, & \text{otherwise,} \end{cases}$$

$$\hat{X} = \underset{X}{\operatorname{argmin}} \sum_{n=1}^{N} w_n(e_n) \; || \; r_n(X) \; ||_{\Lambda_n}$$

Robust Optimization Techniques: Max Mixtures

Max-Mixtures Model: Ideally, we would like to utilize a multi-modal model that more accurately characterizes our measurement model.

$$p(z_i|x) = \sum_i \omega_i \mathcal{N}(\mu_j, \Lambda_j)$$

$$X^* = \operatorname{argmax}_x \sum_i \log \left(\sum_j \omega_j \mathcal{N}(\mu_j, \Lambda_j) \right)$$

Main Idea

The summation operator can be approximated by the max operator to generate a Gaussian mixture model.

$$\boxed{p(z_i|x) \approx \max_i \omega_i \mathcal{N}(\mu_j, \Lambda_j)}$$

Robust Optimization Techniques: Max Mixtures

Max-Mixtures Model: Ideally, we would like to utilize a multi-modal model that more accurately characterizes our measurement model.

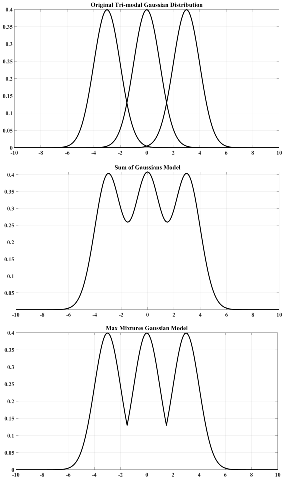
$$p(z_i|x) = \sum_i \omega_i \mathcal{N}(\mu_j, \Lambda_j)$$

$$X^* = \text{argmax}_x \sum_i \log \left(\sum_j \omega_j \mathcal{N}(\mu_j, \Lambda_j) \right)$$

Main Idea

The summation operator can be approximated by the max operator to generate a Gaussian mixture model.

$$\boxed{p(z_i|x) \approx \ \max_i \ \omega_i \mathcal{N}(\mu_j, \Lambda_j)}$$



Robust Optimization Techniques: A simple test case

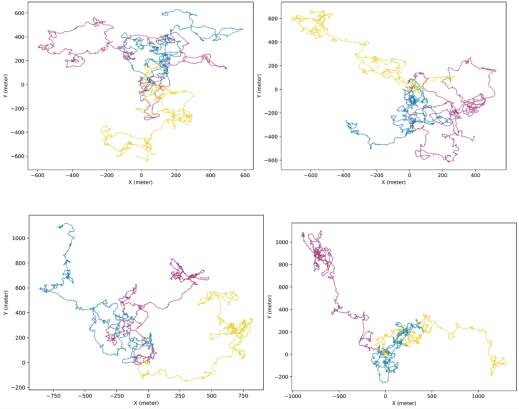
Relative Localization:

- **Goal:** Use VO coupled with radio ranging in a distributed system, to produce an accurate estimate of localization
- **Inputs**
  - VO provides motion constraints
    - short term high accuracy but drifts
  - RF range measurements provide robot-to-robot constraints
    - low accuracy but low drift
  - The factor graph serves as the underlying optimization framework to optimally fuse the odometry and ranging data
    - "anchor" individual robot coordinate frames
    - estimate positions of all robots in the network

Robust Optimization Techniques: **A simple test case**

Relative Localization:

- **Goal:** Use VO coupled with radio ranging in a distributed system, to produce an accurate estimate of localization
- **Inputs**
  - VO provides motion constraints
    - short term high accuracy but drifts
  - RF range measurements provide robot-to-robot constraints
    - low accuracy but low drift
  - The factor graph serves as the underlying optimization framework to optimally fuse the odometry and ranging data
    - "anchor" individual robot coordinate frames
    - estimate positions of all robots in the network



Robust Optimization Techniques: A simple test case

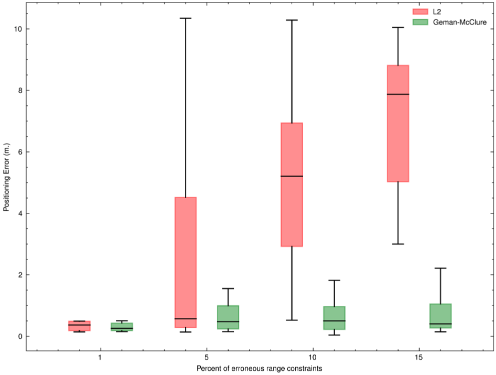
Sensitivity to Ranging Observation Outliers:

- **Goal:** quantify the positioning sensitivity of the proposed approach as a function of the number of erroneous observation
- **Approach**
  - Run the simulation with varying frequencies of faulty ranging observation updates
    - faulty range observations are modeled as: $\rho_f = \rho_t + \mathcal{N}(10m, 1m)$
  - Quantify performance with traditional and robust cost functions
    - Traditional = L2
    - Robust = Geman-McClure

Robust Optimization Techniques: A simple test case

Sensitivity to Ranging Observation Outliers:

- **Goal:** quantify the positioning sensitivity of the proposed approach as a function of the number of erroneous observation
- **Approach**
  - Run the simulation with varying frequencies of faulty ranging observation updates
    - faulty range observations are modeled as: $\rho_f = \rho_t + \mathcal{N}(10m, 1m)$
  - Quantify performance with traditional and robust cost functions
    - Traditional = L2
    - Robust = Geman-McClure



When Do Robust Optimization Techniques Break Down?

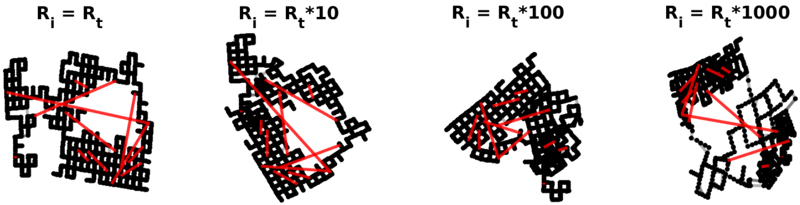
- As we've seen, all of the previously discussed methodologies work fairly well for erroneous observation rejection.
  - However, they all assume the measurement uncertainty model is known *a priori*.

What happens when the measurement uncertainty model is not known accurately *a priori* ?

When Do Robust Optimization Techniques Break Down?

- As we've seen, all of the previously discussed methodologies work fairly well for erroneous observation rejection.
  - However, they all assume the measurement uncertainty model is known *a priori.*

What happens when the measurement uncertainty model is not known accurately *a priori ?*



Handling inaccurate uncertainty models?

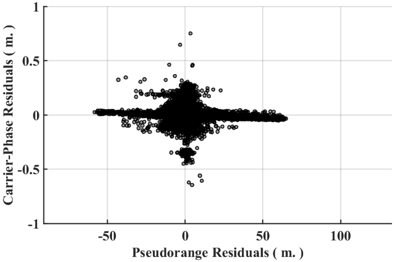
- As we've seen, having an incorrect uncertainty model invalidates the existing robust estimation techniques.
- **To combat this issue, we can explore the concept of learning the uncertainty model during run-time.**

Handling inaccurate uncertainty models?

- As we've seen, having an incorrect uncertainty model invalidates the existing robust estimation techniques.
- **To combat this issue, we can explore the concept of learning the uncertainty model during run-time.**

Main Idea

- We can learn the uncertainty model from the state estimation residuals.
  - $r_n \triangleq z_n - h_n(x)$

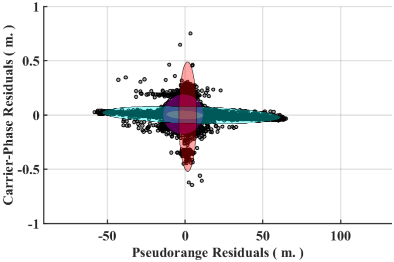


Handling inaccurate uncertainty models?

- As we've seen, having an incorrect uncertainty model invalidates the existing robust estimation techniques.
- **To combat this issue, we can explore the concept of learning the uncertainty model during run-time.**

Main Idea

- We can learn the uncertainty model from the state estimation residuals.
  - $r_n \triangleq z_n - h_n(x)$

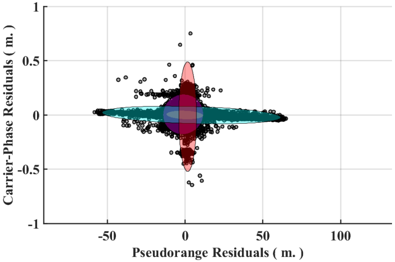


Handling inaccurate uncertainty models?

- As we've seen, having an incorrect uncertainty model invalidates the existing robust estimation techniques.
- **To combat this issue, we can explore the concept of learning the uncertainty model during run-time.**

Main Idea

- We can learn the uncertainty model from the state estimation residuals.
  - $r_n \triangleq z_n - h_n(x)$
- With a multi-modal characterization of the uncertainty model, we can de-weight subsets of the collected observations.

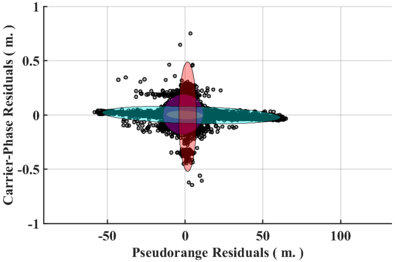


Handling inaccurate uncertainty models?

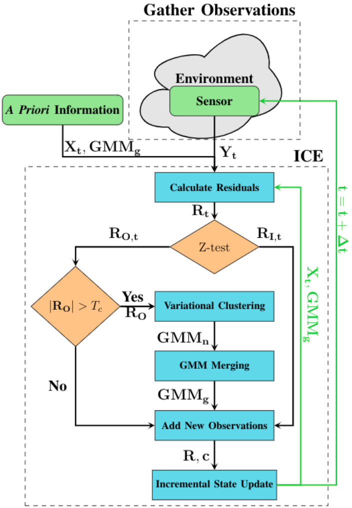
- As we've seen, having an incorrect uncertainty model invalidates the existing robust estimation techniques.
- **To combat this issue, we can explore the concept of learning the uncertainty model during run-time.**

Main Idea

- We can learn the uncertainty model from the state estimation residuals.
  - $r_n \triangleq z_n - h_n(x)$
- With a multi-modal characterization of the uncertainty model, we can de-weight subsets of the collected observations.
- We'll discuss an implementation
  - Incremental Covariance Estimation (ICE)

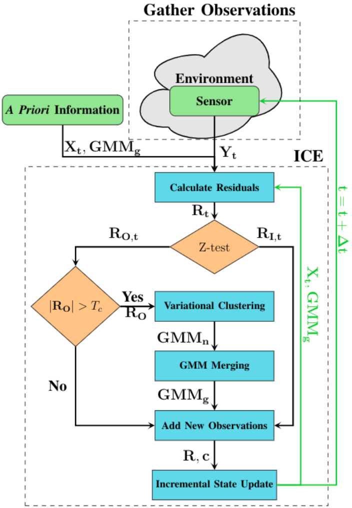


Incremental Covariance Estimation



Algorithm Discussion:

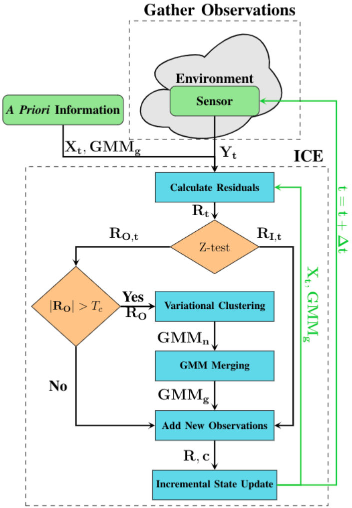
Incremental Covariance Estimation



Algorithm Discussion:

- Partition the observations,
  - $\mathbf{R}_{\mathbf{I},\mathbf{t}} = \{r \mid r \in \mathbf{R}_\mathbf{t},\ Z(r, \phi) < T_r\}$
  - $\mathbf{R}_{\mathbf{O},\mathbf{t}} = \{r \mid r \in \mathbf{R}_\mathbf{t},\ r \notin \mathbf{R}_{\mathbf{I},\mathbf{t}}\}$

Incremental Covariance Estimation



Algorithm Discussion:

- Partition the observations,
  - $\mathbf{R}_{\mathrm{I},t} = \{r \mid r \in \mathbf{R}_t,\ Z(r,\phi) < T_r\}$
  - $\mathbf{R}_{\mathrm{O},t} = \{r \mid r \in \mathbf{R}_t,\ r \notin \mathbf{R}_{\mathrm{I},t}\}$

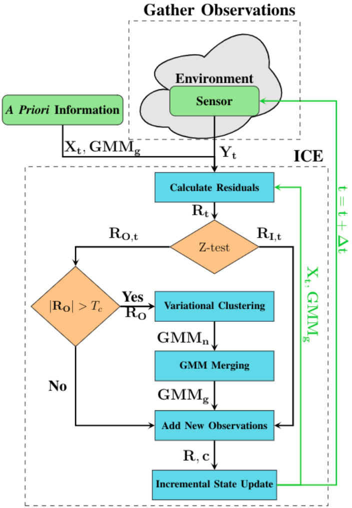
if $|\mathbf{R}_\mathrm{O}| > T_r$

- Learn new uncertainty model

$$\log \mathrm{p}(\mathbf{R}_\mathrm{O}) = \log \int \mathrm{p}(\mathbf{R}_\mathrm{O}, \boldsymbol{\theta}, \mathbf{Z}) d\mathbf{Z} d\boldsymbol{\theta}$$

- Merge new uncertainty model into prior model ($\mathbf{GMM}_g$)

Incremental Covariance Estimation



Algorithm Discussion:

- Partition the observations,
  - $\mathbf{R}_{\mathbf{I},\mathbf{t}} = \{r \mid r \in \mathbf{R}_\mathbf{t},\ Z(r,\phi) < T_r\}$
  - $\mathbf{R}_{\mathbf{O},\mathbf{t}} = \{r \mid r \in \mathbf{R}_\mathbf{t},\ r \notin \mathbf{R}_{\mathbf{I},\mathbf{t}}\}$

if $|\mathbf{R}_\mathbf{O}| > T_r$

  - Learn new uncertainty model

$$\log p(\mathbf{R}_\mathbf{O}) = \log \int p(\mathbf{R}_\mathbf{O}, \boldsymbol{\theta}, \mathbf{Z}) d\mathbf{Z} d\boldsymbol{\theta}$$

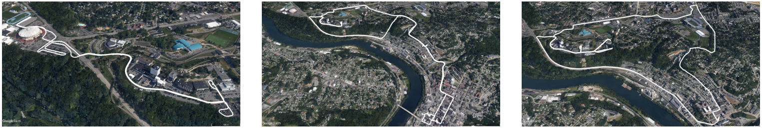
  - Merge new uncertainty model into prior model ($\mathbf{GMM}_g$)

- Incorporate inlier observations ($\mathbf{R}_{\mathbf{I},\mathbf{t}}$).
  - Each observations uncertainty is characterized by a component from $\mathbf{GMM}_g$

$$r_n \sim \underset{\theta \in \mathbf{GMM}_g}{\operatorname{argmax}} \mathscr{L}(\theta | r_n)$$

- Calculate an incremental update.

Experimental Setup: Data Collection

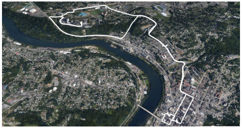


(a) Ground trace for first data collect. (b) Ground trace for second data collect. (c) Ground trace for third data collect.

Figure: Ground trace of three kinematic driving data sets collected in the Morgantown, WV area.

Experimental Setup: Data Collection

(a) Ground trace for first data collect.



(b) Ground trace for second data collect.

(c) Ground trace for third data collect.

Figure: Ground trace of three kinematic driving data sets collected in the Morgantown, WV area.

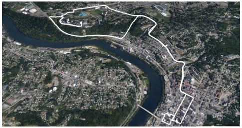
Data Collection

Three kinematic GNSS data sets were collected in the Morgantown, WV area.

- The binary In-phase and Quadrature (IQ) data in the L1-band was recorded using a LabSat-3GPS record and playback device [7].
- These IQ data were played back into two receivers
  - A geodetic-grade GNSS receiver (Novatel OEM-638)
  - A open-source GPS software defined radio [4]

Experimental Setup: Data Collection

(a) Ground trace for first data collect.



(b) Ground trace for second data collect.

(c) Ground trace for third data collect.

Figure: Ground trace of three kinematic driving data sets collected in the Morgantown, WV area.

Data Collection

Three kinematic GNSS data sets were collected in the Morgantown, WV area.

- The binary In-phase and Quadrature (IQ) data in the L1-band was recorded using a LabSat-3GPS record and playback device [7].
- These IQ data were played back into two receivers
  - A geodetic-grade GNSS receiver (Novatel OEM-638)
  - A open-source GPS software defined radio [4]

Table: GPS SDR observation tracking configurations utilized within this study. f_s is the IQ sampling frequency, D_{EL} is the early and late correlator spacing, B_ρ is the range loop bandwidth, and B_ϕ is the carrier loop bandwidth.

GPS Quality	f_s (MHz)	D_{EL} (chips)	B_ρ (Hz)	B_ϕ (Hz)
Low	4.092	0.5	2	50
High	16.368	0.2	1	25

Table: Zero-baseline observation comparison.

Zero-Baseline Comparison Mean 3D Error (m.)	Collect 1	Collect 2	Collect 3
Low	16.20	37.44	29.43
High	0.59	18.03	17.48

Experimental Setup: **Constructing the GNSS FG**

GPS Observation Model

$$\rho^{i}_{L_1} = ||X_s - X_u|| + c(\delta t_u - \delta t_s) + T_{z,d}\mathscr{M}_d(el^j) - I_{L_1} + \delta_{Rel.} + \delta_{P.C.} + \delta_{D.C.B} + \epsilon^j_\rho$$

$$\phi^{i}_{L_1} = ||X_s - X_u|| + c(\delta t_u - \delta t_s) + T_{z,d}\mathscr{M}_d(el^j) - I_{L_1} + \delta_{Rel.} + \delta_{P.C.} + \delta_{W.U.} + \lambda_{IF} N^j_{IF} + \epsilon^j_\phi$$

Experimental Setup: **Constructing the GNSS FG**

GPS Observation Model

$$\rho^{i}_{L_1} = ||X_s - X_u|| + c(\delta t_u - \delta t_s) + T_{z,d}\mathscr{M}_d(el^j) - I_{L_1} + \delta_{Rel.} + \delta_{P.C.} + \delta_{D.C.B} + \epsilon^j_\rho$$

$$\phi^{i}_{L_1} = ||X_s - X_u|| + c(\delta t_u - \delta t_s) + T_{z,d}\mathscr{M}_d(el^j) - I_{L_1} + \delta_{Rel.} + \delta_{P.C.} + \delta_{W.U.} + \lambda_{IF} N^j_{IF} + \epsilon^j_\phi$$

$$\boxed{\psi(i) = ||y - \hat{y}||_\Lambda}$$

REF: Watson, Ryan M., and Jason N. Gross. "Evaluation of kinematic precise point positioning convergence with an incremental graph optimizer." 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS). IEEE, 2018.

Experimental Setup: Constructing the GNSS FG

GPS Observation Model

$$\rho^{j}_{L_1} = ||X_s - X_u|| + c(\delta t_u - \delta t_s) + T_{z,d}\mathscr{M}_d(el^j) - I_{L_1} + \delta_{Rel.} + \delta_{P.C.} + \delta_{D.C.B} + \epsilon^{j}_{\rho}$$

$$\phi^{j}_{L_1} = ||X_s - X_u|| + c(\delta t_u - \delta t_s) + T_{z,d}\mathscr{M}_d(el^j) - I_{L_1} + \delta_{Rel.} + \delta_{P.C.} + \delta_{W.U.} + \lambda_{IF} N^{j}_{IF} + \epsilon^{j}_{\phi}$$

$$\boxed{\psi(i) = ||y - \hat{y}||_{\Lambda}}$$

REF: Watson, Ryan M., and Jason N. Gross. "Evaluation of kinematic precise point positioning convergence with an incremental graph optimizer." 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS). IEEE, 2018.

$$X = \begin{pmatrix} \delta P \\ T_{z,w} \\ \delta t_u \\ N_1 \\ \vdots \\ N_n \end{pmatrix}$$

Experimental Setup: Constructing the GNSS FG

GPS Observation Model

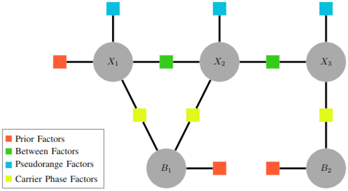
$$\rho^j_{L_1} = ||X_s - X_u|| + c(\delta t_u - \delta t_s) + T_{z,d}\mathscr{M}_d(el^j) \\ - I_{L_1} + \delta_{Rel.} + \delta_{P.C.} + \delta_{D.C.B} + \epsilon^j_\rho$$

$$\phi^j_{L_1} = ||X_s - X_u|| + c(\delta t_u - \delta t_s) + T_{z,d}\mathscr{M}_d(el^j) \\ - I_{L_1} + \delta_{Rel.} + \delta_{P.C.} + \delta_{W.U.} + \lambda_{IF} N^j_{IF} + \epsilon^j_\phi$$

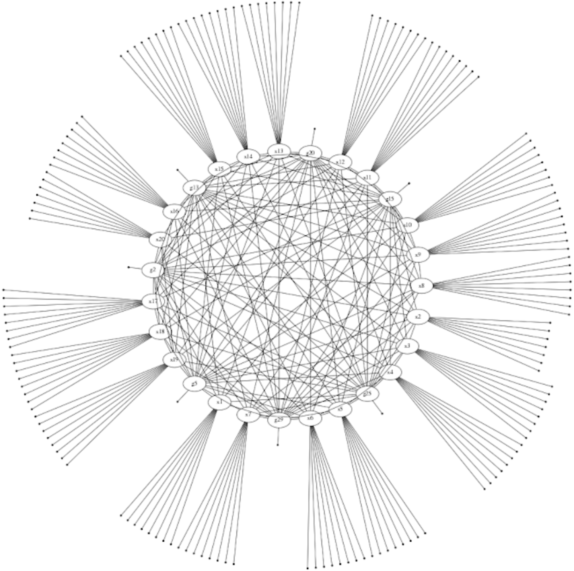
$$\boxed{\psi(i) = ||y - \hat{y}||_\Lambda}$$

REF: Watson, Ryan M., and Jason N. Gross. "Evaluation of kinematic precise point positioning convergence with an incremental graph optimizer." 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS). IEEE, 2018.

$$X = \begin{pmatrix} \delta P \\ T_{z,w} \\ \delta t_u \\ N_1 \\ \vdots \\ N_n \end{pmatrix}$$



Robust Estimation Experimental Evaluation



Robust Estimation Experimental Evaluation

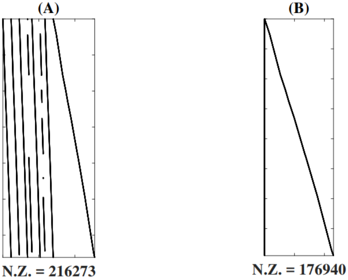


Fig. 2: Sparse measurement Jacobian for the PPP processing strategy. Figure (A) shows the measurement Jacobian when a new carrier-phase ambiguity state is added for each epoch. Figure (B) shows the measurement Jacobian when a new carrier-phase ambiguity is added only when a new satellite is tracked or if a carrier-phase cycle-slip occurs.

Robust Estimation Experimental Evaluation

TABLE I: Horizontal RSOS localization error results when low fidelity receiver tracking parameters are utilized to generate the observations. The green and red cell entries correspond to the minimum and maximum statistic, respectively.

(a) Localization results for data collect 1.

(m.)	L_2	DCS	MM	ICE
mean	2.51	0.99	1.66	0.73
median	2.57	0.64	1.63	0.56
std. dev.	1.41	0.98	1.05	0.72
max	10.78	9.71	10.06	13.19

(b) Localization results for data collect 2.

(m.)	L_2	DCS	MM	ICE
mean	4.00	4.00	3.12	2.11
median	2.48	2.08	1.94	0.93
std. dev.	3.87	4.59	3.92	2.10
max	29.18	31.05	31.40	23.02

(c) Localization results for data collect 3.

(m.)	L_2	DCS	MM	ICE
mean	4.94	4.16	4.51	4.35
median	4.41	2.82	3.62	1.48
std. dev.	2.97	3.54	3.33	5.23
max	29.53	30.38	28.30	26.61

TABLE II: Horizontal RSOS localization error results when high fidelity receiver tracking parameters are utilized to generate the observations. The green and red cell entries correspond to the minimum and maximum statistic, respectively.

(a) Localization results for data collect 1.

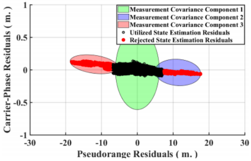
(m.)	L_2	DCS	MM	ICE
mean	0.44	0.43	0.41	0.42
median	0.37	0.36	0.35	0.35
std. dev.	0.30	0.27	0.29	0.28
max	5.38	5.33	5.35	5.22

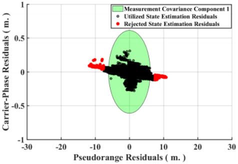
(b) Localization results for data collect 2.

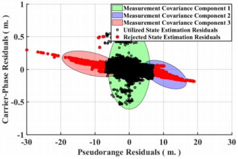
(m.)	L_2	DCS	MM	ICE
mean	0.79	0.81	0.84	0.79
median	0.82	0.81	0.84	0.83
std. dev.	0.46	0.46	0.50	0.46
max	3.97	3.93	10.77	2.95

(c) Localization results for data collect 3

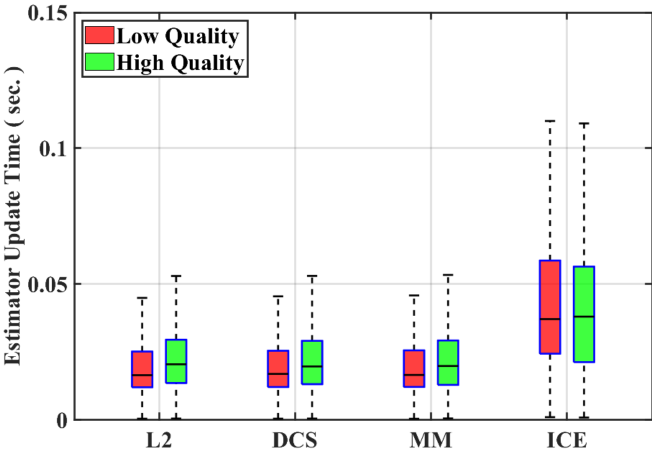
(m.)	L_2	DCS	MM	ICE
mean	1.09	1.10	1.11	1.07
median	0.96	0.95	1.00	0.89
std. dev.	0.67	0.73	0.72	0.66
max	7.83	7.83	18.08	7.82







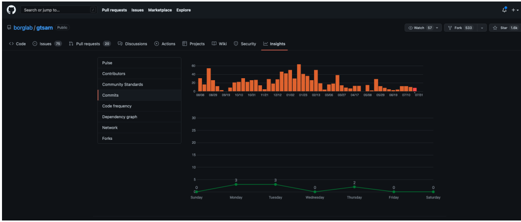
Robust Estimation Experimental Evaluation: Run Time



Factor Graph Software

GTSAM: Georgia Tech Smoothing and Mapping

- After you get a handle on the underlying theory, there are tools available to help with the implementation of factor graphs.
- Our favorite is Georgia Tech Smoothing and Mapping (GTSAM).
  - “BSD-licensed C++ library that implements sensor fusion for robotics and computer vision applications, including SLAM (Simultaneous Localization and Mapping), VO (Visual Odometry), and SFM (Structure from Motion).“
  - Developers are actively maintaining and developing the code base.
- References:
  - `https://gtsam.org/`
  - `https://github.com/borglab/gtsam`
  - Dellaert, Frank, and Michael Kaess. "Factor graphs for robot perception." Foundations and Trends® in Robotics 6.1-2 (2017): 1-139.



GTSAM

- The easiest way to get a handle on how to utilize GTSAM for your projects is by working through some of the available examples.
- So, we'll walk through three navigation examples as implemented with GTSAM.
  1. A simple GTSAM localization example
     - Only utilizes built-in features.
     - Will give an overview of the general structure of GTSAM code.
     - `https://colab.research.google.com/drive/1rZHFgYjETuXp3ktxCgNZvpNmSLhPbmqn?usp=sharing`
  2. A GNSS specific example
     - Will show how to develop a new factor graph model to accommodate GNSS range/phase observations.
     - `https://github.com/wvu-navLab/PPP-BayesTree`
  3. A python3 example with UWB ranging radios.
     - Will provide an overview of how to utilize the python3 wrapper for GTSAM.
     - `https://colab.research.google.com/drive/17Ht31XTYvp2Buy5iF5tH9X2zfbxxx-Dx?`

```cpp
using namespace std;
using namespace gtsam;

int main(int argc, char** argv) {
  // Create an empty nonlinear factor graph
  NonlinearFactorGraph graph;

  // Add a prior on the first pose, setting it to the origin
  // A prior factor consists of a mean and a noise model (covariance matrix)
  Pose2 priorMean(0.0, 0.0, 0.0);  // prior at origin
  auto priorNoise = noiseModel::Diagonal::Sigmas(Vector3(0.3, 0.3, 0.1));
  graph.addPrior(1, priorMean, priorNoise);

  // Add odometry factors
  Pose2 odometry(2.0, 0.0, 0.0);
  // For simplicity, we will use the same noise model for each odometry factor
  auto odometryNoise = noiseModel::Diagonal::Sigmas(Vector3(0.2, 0.2, 0.1));
  // Create odometry (Between) factors between consecutive poses
  graph.emplace_shared<BetweenFactor<Pose2> >(1, 2, odometry, odometryNoise);
  graph.emplace_shared<BetweenFactor<Pose2> >(2, 3, odometry, odometryNoise);
  graph.print("\nFactor Graph:\n");  // print

  // Create the data structure to hold the initialEstimate estimate to the solution
  // For illustrative purposes, these have been deliberately set to incorrect values
  Values initial;
  initial.insert(1, Pose2(0.5, 0.0, 0.2));
  initial.insert(2, Pose2(2.3, 0.1, -0.2));
  initial.insert(3, Pose2(4.1, 0.1, 0.1));
  initial.print("\nInitial Estimate:\n");  // print

  // optimize using Levenberg-Marquardt optimization
  Values result = LevenbergMarquardtOptimizer(graph, initial).optimize();
  result.print("Final Result:\n");

  // Calculate and print marginal covariances for all variables
  cout.precision(2);
  Marginals marginals(graph, result);
  cout << "x1 covariance:\n" << marginals.marginalCovariance(1) << endl;
  cout << "x2 covariance:\n" << marginals.marginalCovariance(2) << endl;
  cout << "x3 covariance:\n" << marginals.marginalCovariance(3) << endl;

  return 0;
}
```

[1] P. Agarwal et al. "Robust Map Optimization Using Dynamic Covariance Scaling". In: **International Conference on Robotics and Automation**. 2013.

[2] Pratik Agarwal. "Robust graph-based localization and mapping". PhD thesis. Dissertation, Albert-Ludwigs-Universität Freiburg, 2015.

[3] Luca Carlone, Andrea Censi, and Frank Dellaert. "Selecting good measurements via ℓ-1 relaxation: A convex approach for robust estimation over graphs". In: **2014 IEEE/RSJ International Conference on Intelligent Robots and Systems**. IEEE. 2014, pp. 2667–2674.

[4] Carles Fernandez-Prades et al. "GNSS-SDR: an open source tool for researchers and developers". In: **Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)**. 2001, pp. 780–0.

[5] Martin A Fischler and Robert C Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: **Communications of the ACM** 24.6 (1981), pp. 381–395.

[6] Peter J Huber. **Robust Statistics**. Wiley New York, 1981.

[7] **LabSat 3 GPS SImulator**. `https://www.labsat.co.uk/index.php/en/products/labsat-3`. Accessed 3/1/19.

[8] Yasir Latif, César Cadena, and José Neira. "Realizing, reversing, recovering: Incremental robust loop closing over time using the irrr algorithm". In: **2012 IEEE/RSJ International Conference on Intelligent Robots and Systems**. IEEE. 2012, pp. 4211–4217.

[9] E. Olson and P. Agarwal. "Inference on Networks of Mixtures for Robust Robot Mapping". In: 2012.

[10] Niko Sünderhauf et al. "Multipath mitigation in GNSS-based localization using robust optimization". In: **2012 IEEE Intelligent Vehicles Symposium**. IEEE. 2012, pp. 784–789.

[11] Frank Van Diggelen and Alison Brown. "Mathematical aspects of GPS RAIM". In: **Proceedings of 1994 IEEE Position, Location and Navigation Symposium-PLANS'94**. IEEE. 1994, pp. 733–738.

[12] Christopher Zach and Guillaume Bourmaud. "Iterated lifting for robust cost optimization". In: **BMVC**. 2017.