Ingeniería Eléctrica
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS UNIVERSIDAD DE CHILE

Thesis Defense

GraphSLAM Algorithm Implementation for Solving Simultaneous Localization and Mapping

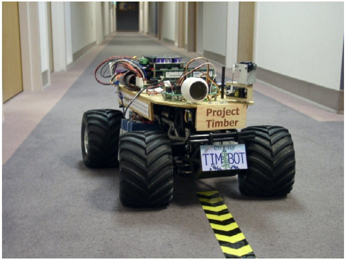
Author: Franco Curotto
Thesis Adviser: Martin Adams
Commission Members: Marcos Orchard
Jorge Silva

Departamento de Ingeniería Eléctrica
Facultad de Ciencias Físicas y Matemáticas
Universidad de Chile

April 18, 2016

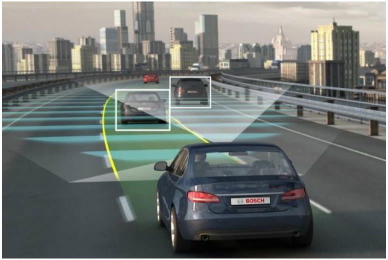
Motivation

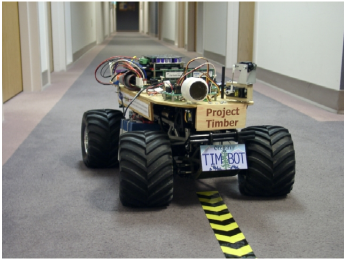
Robots Before

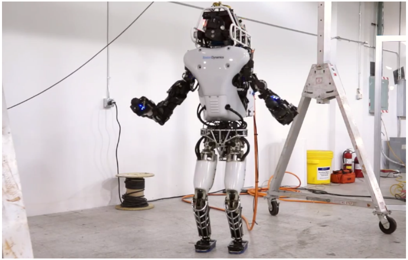


Motivation

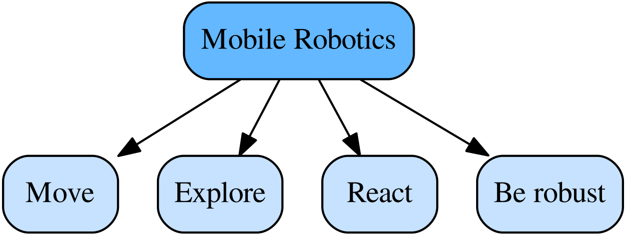
Robots Before	Robots Today

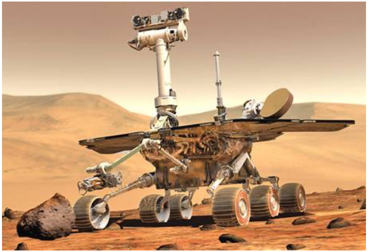


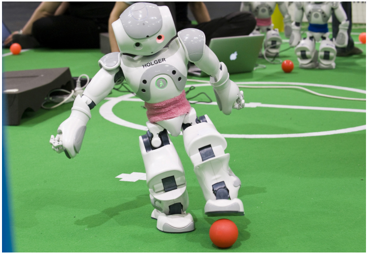




Motivation







Simultaneous Localization and Mapping

SLAM

The problem were an agent must simultaneously estimate its current position (localization), and construct a map of its environment (mapping)

Simultaneous Localization and Mapping

SLAM

The problem were an agent must simultaneously estimate its current position (localization), and construct a map of its environment (mapping)

We have (information):

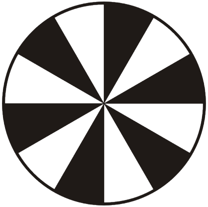
- Robot odometry
- Landmarks measurements

We need (states):

- Robot position
- Robot orientation

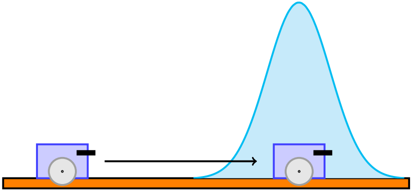
Robot pose

- Landmarks position



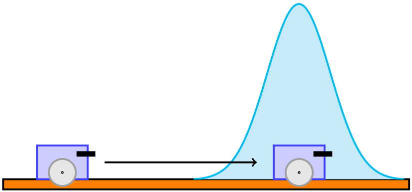
Motion Model:

$$\boldsymbol{x}_k = \boldsymbol{g}(\boldsymbol{u}_k, \boldsymbol{x}_{k-1}) + \boldsymbol{\delta}_k$$



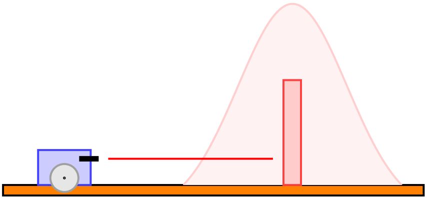
Motion Model:

$$\boldsymbol{x}_k = \boldsymbol{g}(\boldsymbol{u}_k, \boldsymbol{x}_{k-1}) + \boldsymbol{\delta}_k$$



Measurement Model:

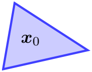
$$\boldsymbol{z}_k^i = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{m}_j, i) + \boldsymbol{\varepsilon}_k^i$$



Both contaminated with Gaussian noise.

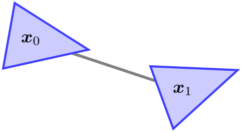
GraphSLAM

GraphSLAM: SLAM $\Rightarrow$ graph



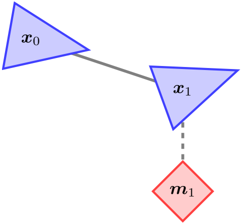
GraphSLAM

GraphSLAM: SLAM $\Rightarrow$ graph



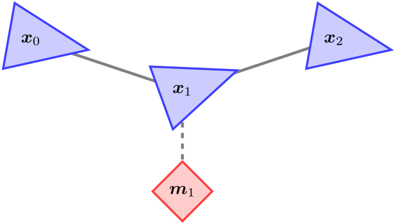
GraphSLAM

GraphSLAM: SLAM $\Rightarrow$ graph



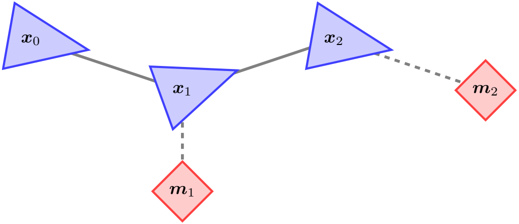
GraphSLAM

GraphSLAM: SLAM $\Rightarrow$ graph



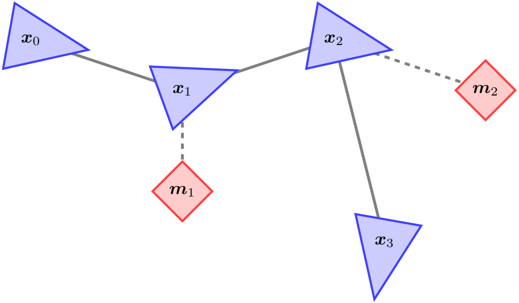
GraphSLAM

GraphSLAM: SLAM $\Rightarrow$ graph



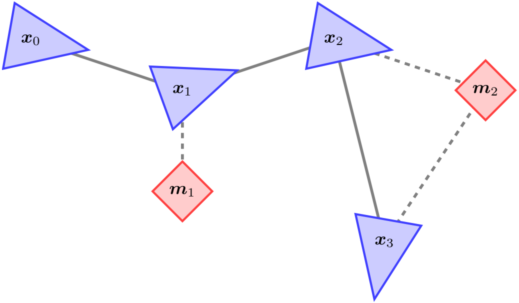
GraphSLAM

GraphSLAM: SLAM ⇒ graph



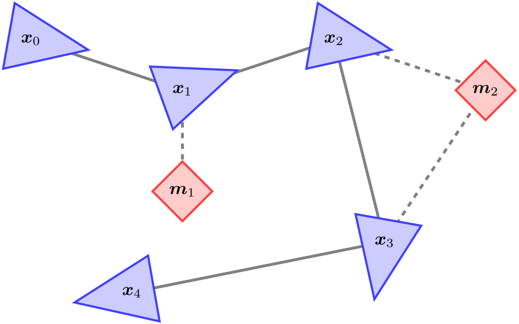
GraphSLAM

GraphSLAM: SLAM $\Rightarrow$ graph



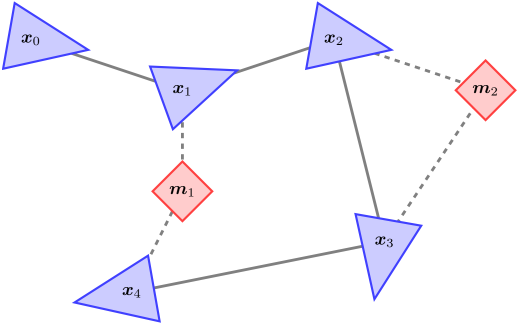
GraphSLAM

GraphSLAM: SLAM $\Rightarrow$ graph



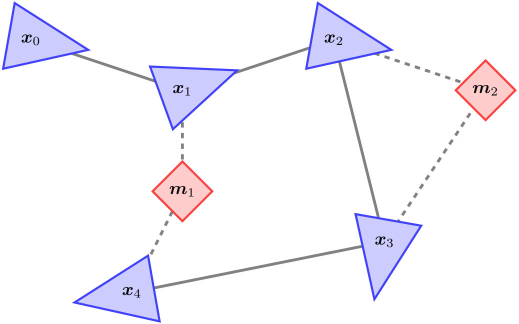
GraphSLAM

GraphSLAM: SLAM $\Rightarrow$ graph



GraphSLAM

GraphSLAM: SLAM $\Rightarrow$ graph



Nodes are: Robot poses, or
Landmarks positions

Edges are: Odometry, or
Measurements

Goal: Find the graph that more accurately represents the data.

Mathematical Formulation of GraphSLAM

SLAM posterior probability:

$$\boxed{p(\boldsymbol{x}_{0:k}, \boldsymbol{m} | \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k})}$$

Mathematical Formulation of GraphSLAM

SLAM posterior probability:

$$\boxed{p(\overbrace{\boldsymbol{x}_{0:k}}^{\text{pose}}, \overbrace{\boldsymbol{m}}^{\text{map}} | \underbrace{\boldsymbol{z}_{1:k}}_{\text{measurements}}, \underbrace{\boldsymbol{u}_{1:k}}_{\text{odometry}})}$$

Mathematical Formulation of GraphSLAM

SLAM posterior probability:

pose map

$$\boxed{p(\boldsymbol{x}_{0:k}, \boldsymbol{m} | \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k})}$$

measurements odometry

Using models in $-\log(p(\boldsymbol{y}|\boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k}))$:

$$\boxed{F(\boldsymbol{y}) := \sum \boldsymbol{e}_{ij}(\boldsymbol{y})^T \boldsymbol{\Omega}_{ij} \boldsymbol{e}_{ij}(\boldsymbol{y})}$$

We need the maximum of this equation.

Mathematical Formulation of GraphSLAM

SLAM posterior probability:

pose map

$$\boxed{p(\boldsymbol{x}_{0:k}, \boldsymbol{m} | \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k})}$$

measurements odometry

Using models in $-\log(p(\boldsymbol{y}|\boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k}))$:

For all edges in graph

information matrix

$$\boxed{F(\boldsymbol{y}) := \sum \boldsymbol{e}_{ij}(\boldsymbol{y})^T \boldsymbol{\Omega}_{ij} \boldsymbol{e}_{ij}(\boldsymbol{y})}$$

error function $[\boldsymbol{x}_{0:k}, \boldsymbol{m}]$

We need the maximum of this equation.

Mathematical Formulation of GraphSLAM

- $\mathrm{g}^2\mathrm{o}$: General Graph Optimization framework

Mathematical Formulation of GraphSLAM

- g^{2}o: General Graph Optimization framework

- Gauss-Newton Algorithm:
  1. Taylor Approximation:

$$F(\breve{\boldsymbol{y}} + \boldsymbol{\Delta y}) = k + 2\boldsymbol{b}\boldsymbol{\Delta y} + \boldsymbol{\Delta y}^T \boldsymbol{H}\boldsymbol{\Delta y}$$

- $\boldsymbol{b}$: Total system information vector
- $\boldsymbol{H}$: Total system information matrix

Mathematical Formulation of GraphSLAM

- g^{2}o: General Graph Optimization framework
- Gauss-Newton Algorithm:
  1. Taylor Approximation:

$$F(\breve{y} + \boldsymbol{\Delta y}) = k + 2\boldsymbol{b}\boldsymbol{\Delta y} + \boldsymbol{\Delta y}^T \boldsymbol{H} \boldsymbol{\Delta y}$$

  2. Minimization:

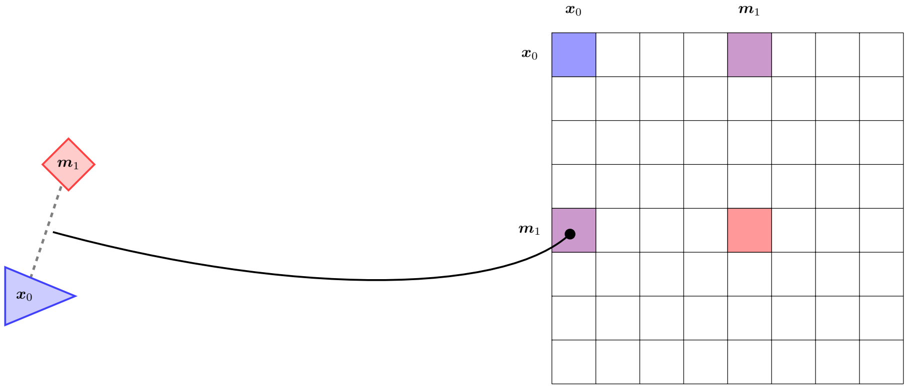
$$\boldsymbol{H}\boldsymbol{\Delta y}^* = -\boldsymbol{b}$$

  3. Update:

$$\boldsymbol{y}^* = \breve{y} + \boldsymbol{\Delta y}^*$$

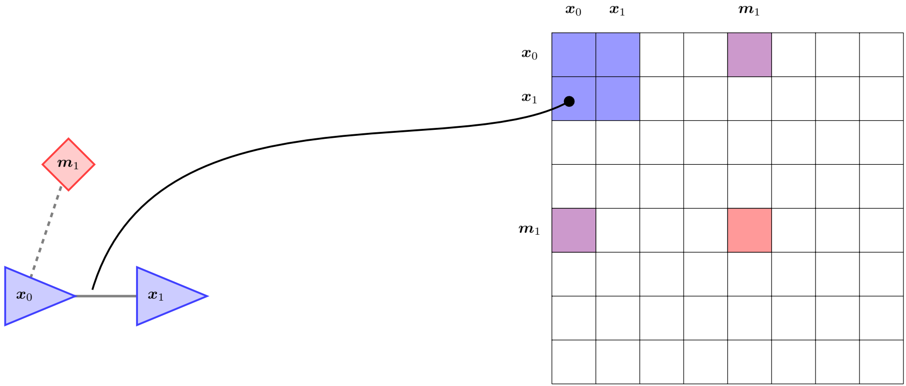
- $\boldsymbol{b}$: Total system information vector
- $\boldsymbol{H}$: Total system information matrix

Information matrix $\boldsymbol{H}$ is intrinsically ***sparse***.

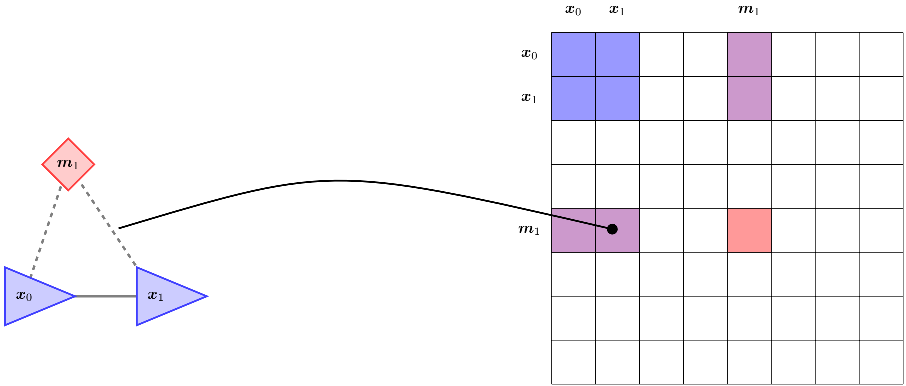


Structure of the Linearized System

Information matrix $\boldsymbol{H}$ is intrinsically ***sparse***.

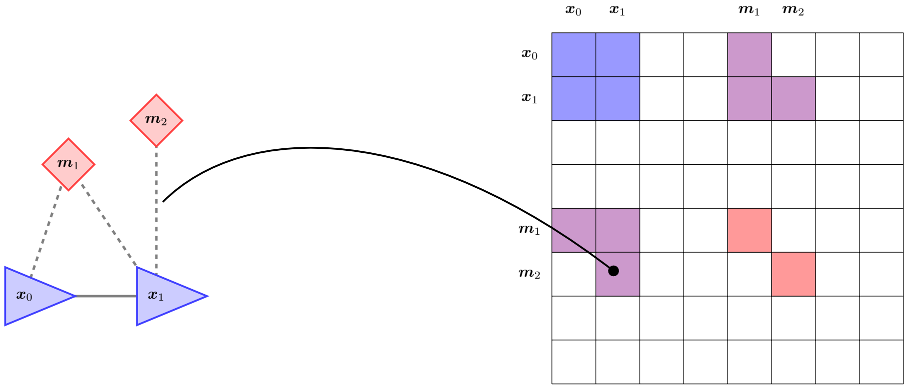


Information matrix $\boldsymbol{H}$ is intrinsically ***sparse***.



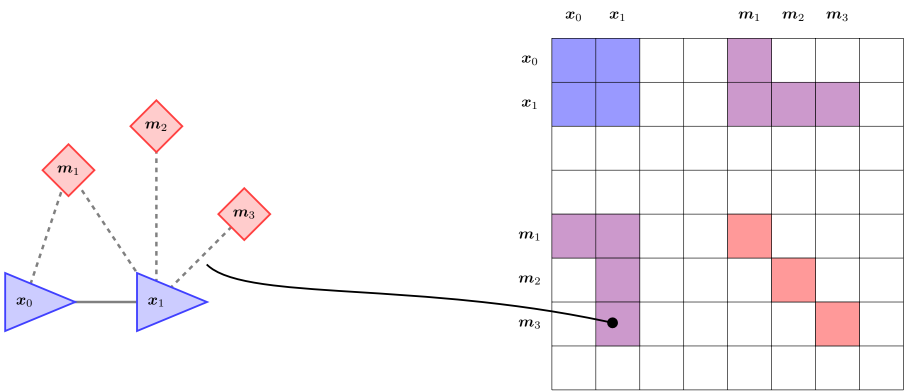
Structure of the Linearized System

Information matrix $\boldsymbol{H}$ is intrinsically ***sparse***.



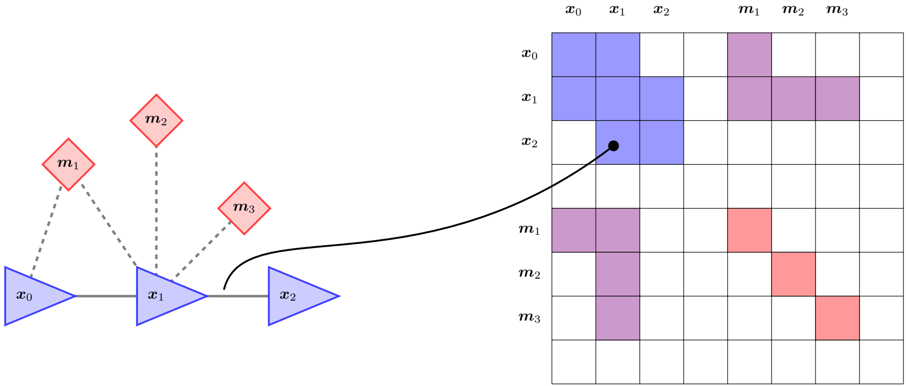
Structure of the Linearized System

Information matrix $\boldsymbol{H}$ is intrinsically ***sparse***.



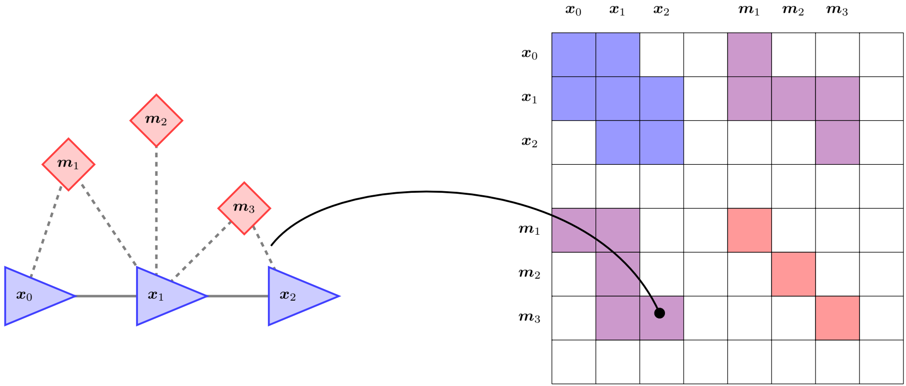
Structure of the Linearized System

Information matrix $\boldsymbol{H}$ is intrinsically ***sparse***.



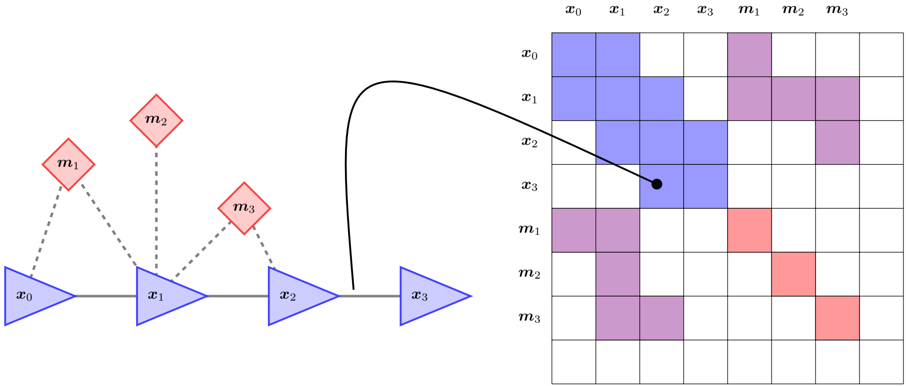
Structure of the Linearized System

Information matrix $\boldsymbol{H}$ is intrinsically ***sparse***.



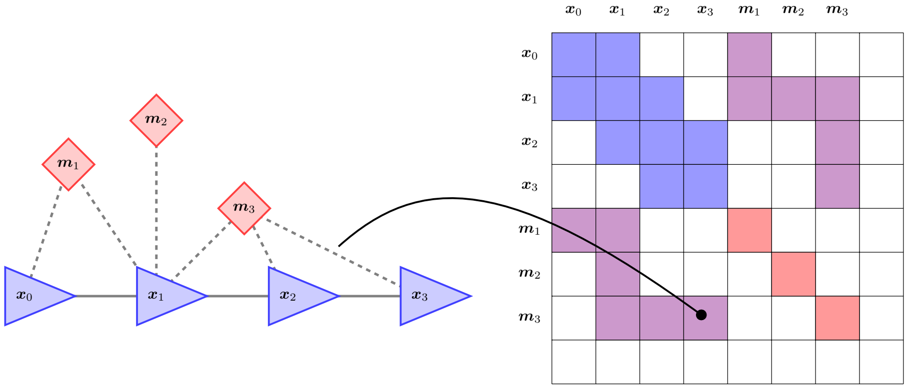
Structure of the Linearized System

Information matrix $\boldsymbol{H}$ is intrinsically ***sparse***.



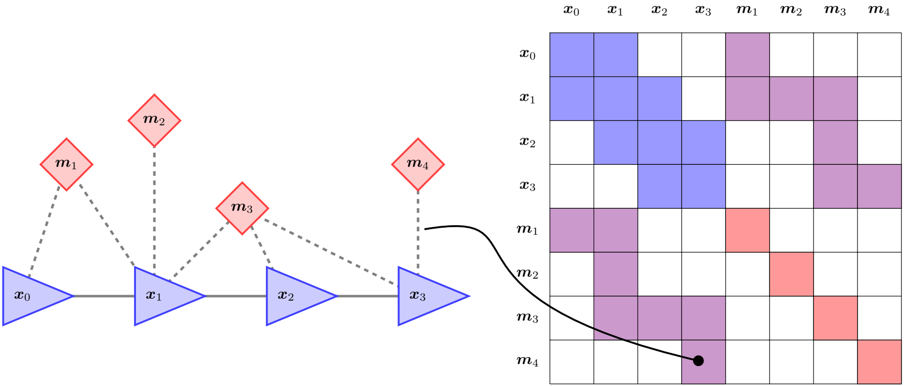
Structure of the Linearized System

Information matrix $\boldsymbol{H}$ is intrinsically ***sparse***.



Structure of the Linearized System

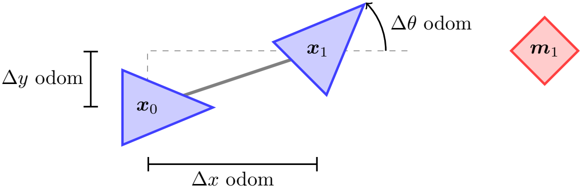
Information matrix $\boldsymbol{H}$ is intrinsically ***sparse***.



g^2o Protocol

Graph Element	Notation
Pose Node	`VERTEX_SE2 id x y a`
Landmark Node	`VERTEX_XY id x y`
Fix a Node	`FIX id`

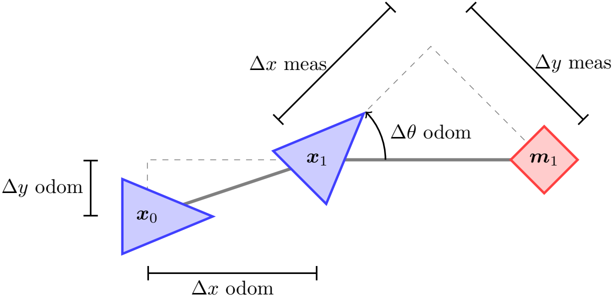
g^2o Protocol



Graph Element	Notation
Pose Node	`VERTEX_SE2 id x y a`
Landmark Node	`VERTEX_XY id x y`
Fix a Node	`FIX id`
Odometry Edge	`EDGE_SE2 id1 id2 dx dy da ipxx ipxy ipxa ipyy ipya ipaa`

One must specify the information matrix of edges.

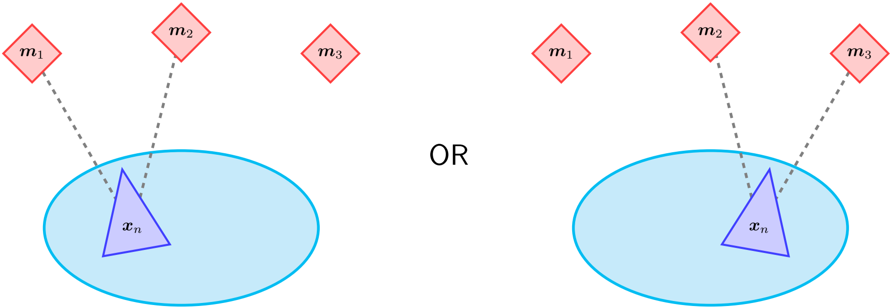
g^2o Protocol



Graph Element	Notation
Pose Node	`VERTEX_SE2 id x y a`
Landmark Node	`VERTEX_XY id x y`
Fix a Node	`FIX id`
Odometry Edge	`EDGE_SE2 id1 id2 dx dy da ipxx ipxy ipxa ipyy ipya ipaa`
Measurement Edge	`EDGE_SE2_XY id1 id2 dx dy ilxx ilxy ilyy`

One must specify the information matrix of edges.

Correspondence Problem



- In the real world, **correspondence** between measurements and landmarks is **unknown**.
- **Data association**: assignment of measurements to landmarks.

Correspondence Test

Premise: Compute the likelihood of two landmarks being the same.

$$\pi_{i=j} := \det(2\pi\boldsymbol{\Sigma}_{\boldsymbol{\Delta}_{i,j}})^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\boldsymbol{\mu}_{\boldsymbol{\Delta}_{i,j}}^{T}\boldsymbol{\Sigma}_{\boldsymbol{\Delta}_{i,j}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\Delta}_{i,j}}\right\}$$

Correspondence Test

Premise: Compute the likelihood of two landmarks being the same.

$$\pi_{i=j} := \det(2\pi \boldsymbol{\Sigma}_{\boldsymbol{\Delta}_{i,j}})^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\boldsymbol{\mu}_{\boldsymbol{\Delta}_{i,j}}^{T}\boldsymbol{\Sigma}_{\boldsymbol{\Delta}_{i,j}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\Delta}_{i,j}}\right\}$$

We define threshold χ.

If $\pi_{j=k} \geq \chi$:

- Merge landmarks i and j

Else:

- Leave them separated

Correspondence Test

Premise: Compute the likelihood of two landmarks being the same.

$$\pi_{i=j} := \det(2\pi\boldsymbol{\Sigma}_{\boldsymbol{\Delta}_{i,j}})^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\boldsymbol{\mu}_{\boldsymbol{\Delta}_{i,j}}^T \boldsymbol{\Sigma}_{\boldsymbol{\Delta}_{i,j}}^{-1} \boldsymbol{\mu}_{\boldsymbol{\Delta}_{i,j}}\right\}$$

We define threshold χ.

If $\pi_{j=k} \geq \chi$:

- Merge landmarks i and j

Else:

- Leave them separated

Problem: Matrix $\boldsymbol{\Sigma}_{\boldsymbol{\Delta}_{i,j}}$ is expensive to compute.

Distant Test

- Merges landmark according to the distance.
- Avoids the computation of $\boldsymbol{\Sigma}_{\boldsymbol{\Delta}_{i,j}}$ for *easy* cases.

Distant Test

- Merges landmark according to the distance.
- Avoids the computation of $\boldsymbol{\Sigma}_{\boldsymbol{\Delta}_{i,j}}$ for *easy* cases.

Maximum distance at which associations can still be made:

$$\boldsymbol{\mu}_{\boldsymbol{\delta}_{i,j}} = \sqrt{-2\sigma^2_{\boldsymbol{\delta}_{i,j}} \log(\sqrt{2\pi\sigma^2_{\boldsymbol{\delta}_{i,j}}}\chi)}$$

Distant Test

- Merges landmark according to the distance.
- Avoids the computation of $\boldsymbol{\Sigma}_{\boldsymbol{\Delta}_{i,j}}$ for *easy* cases.

Maximum distance at which associations can still be made:

$$\boldsymbol{\mu}_{\boldsymbol{\delta}_{i,j}} = \sqrt{-2\sigma^2_{\boldsymbol{\delta}_{i,j}} \log(\sqrt{2\pi\sigma^2_{\boldsymbol{\delta}_{i,j}}}\chi)}$$

Maximum distance, independent of $\sigma_{\boldsymbol{\delta}_{i,j}}$:

$$\boldsymbol{\mu}_{\boldsymbol{\delta}_{i,j}} = \frac{1}{\sqrt{2\pi}e\chi}$$

Distant Test

- Merges landmark according to the distance.
- Avoids the computation of $\mathbf{\Sigma}_{\mathbf{\Delta}_{i,j}}$ for *easy* cases.

Maximum distance at which associations can still be made:

$$\boldsymbol{\mu}_{\boldsymbol{\delta}_{i,j}} = \sqrt{-2\sigma^2_{\boldsymbol{\delta}_{i,j}} \log(\sqrt{2\pi\sigma^2_{\boldsymbol{\delta}_{i,j}}}\chi)}$$

Maximum distance, independent of $\sigma_{\boldsymbol{\delta}_{i,j}}$:

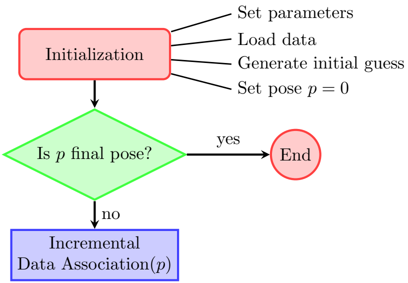
$$\boldsymbol{\mu}_{\boldsymbol{\delta}_{i,j}} = \frac{1}{\sqrt{2\pi}e\chi}$$

The computation saved with this method may be low.

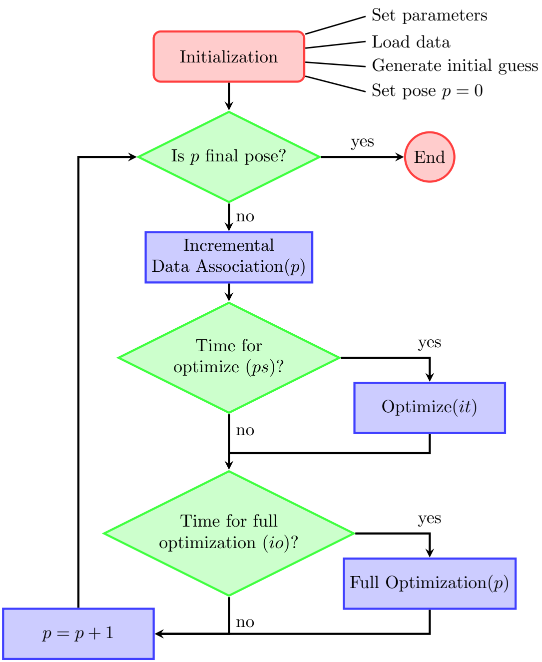
The Final Algorithm

- Initialization
  - Set parameters
  - Load data
  - Generate initial guess
  - Set pose $p = 0$

The Final Algorithm



The Final Algorithm



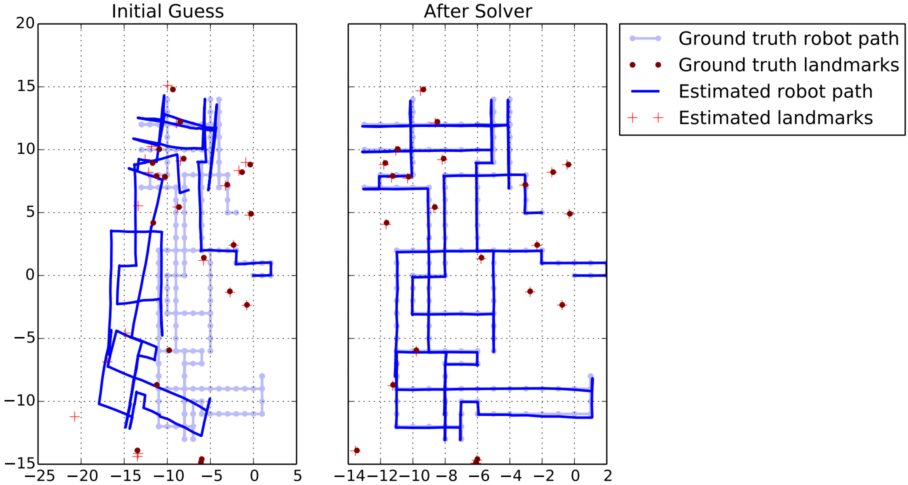
The Final Algorithm

Algorithm 1 Incremental Data Association Function

```
function IncrementalDataAssociation(p)
    for all landmarks l_p observed in p do
        for all previous poses q up to p do
            for all landmarks l_q observed in q do
                if distance(l_p, l_q) < dt then
                    if correspondenceTest(i, j) ≥ χ then
                        optimizer.merge(i, j)
                    end if
                end if
            end for
        end for
    end for
end function
```

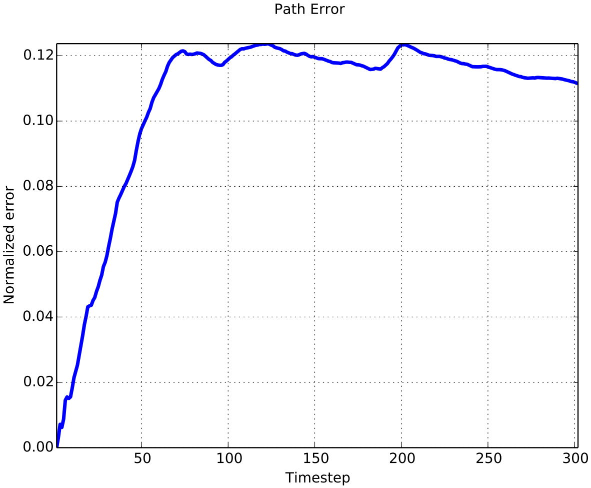
Results - Simulation - Known Data Association

n_p	n_l	i_{op}	i_{oa}	i_{lp}	it	k_w
300	40	1000	1000	1000	20	1



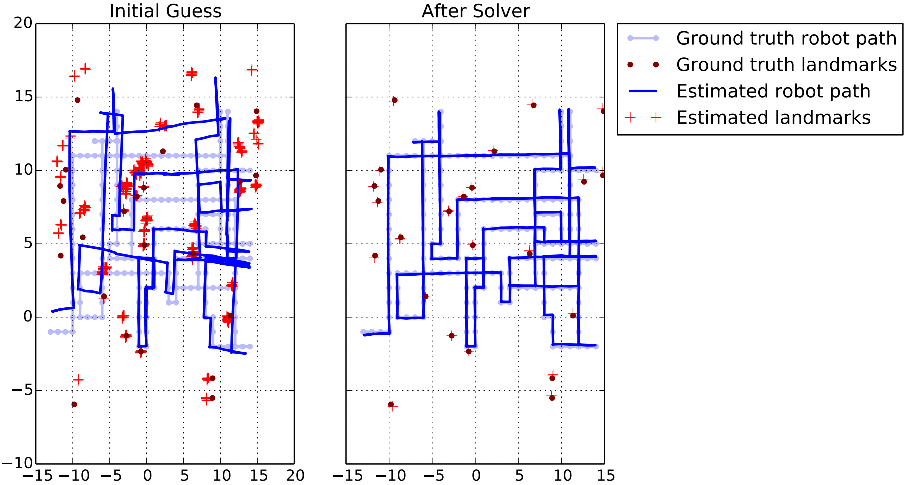
Results - Simulation - Known Data Association

n_p	n_l	i_{op}	i_{oa}	i_{lp}	it	k_w
300	40	1000	1000	1000	20	1



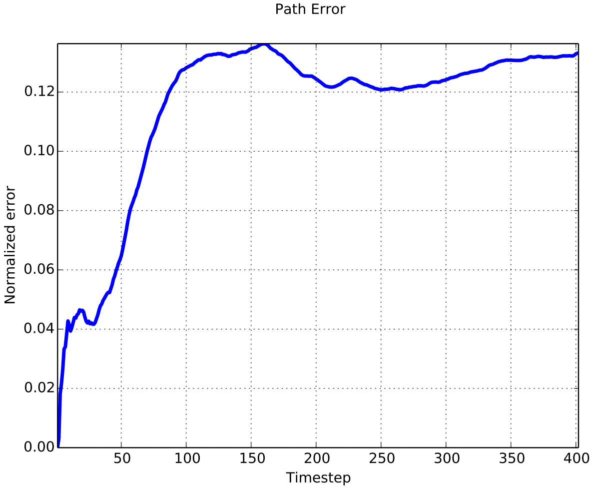
Results - Simulation - Unknown Data Association

n_p	n_l	i_{op}	i_{oa}	i_{lp}	it	k_w	χ	dt	io	ps	t
400	30	1000	10000	1000	20	1	0.1	∞	400	10	$13[s]$



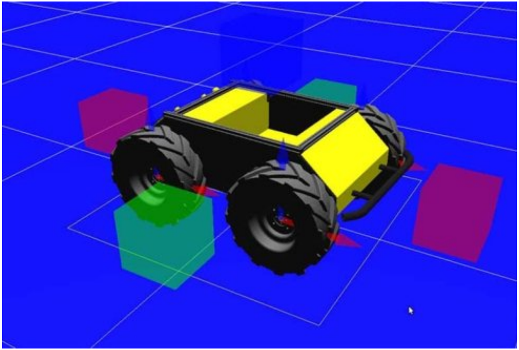
Results - Simulation - Unknown Data Association

n_p	n_l	i_{op}	i_{oa}	i_{lp}	it	k_w	χ	dt	io	ps	t
400	30	1000	10000	1000	20	1	0.1	∞	400	10	$13[s]$



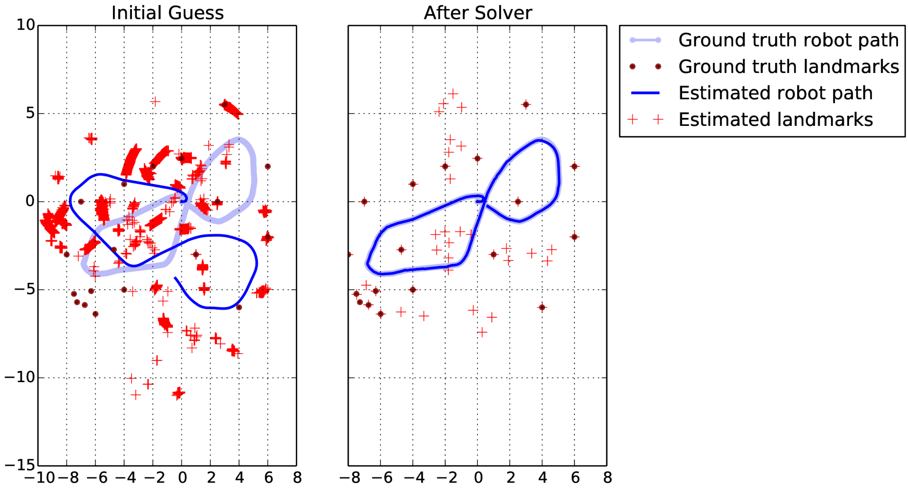
Results - Real Data - Husky a200 ROS Simulation

n_p	n_l	i_{op}	i_{oa}	i_{lp}	it	k_w	χ	dt	io	ps	t
4700	18	10000	10000	1000	10	1	∞	0.5	500	10	$3[min]$



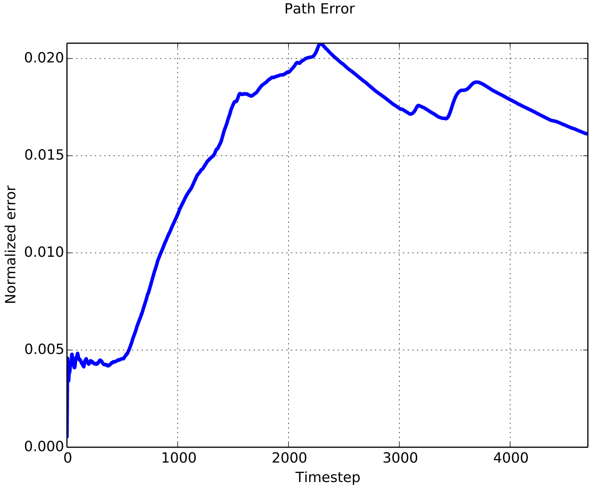
Results - Real Data - Husky a200 ROS Simulation

n_p	n_l	i_{op}	i_{oa}	i_{lp}	it	k_w	χ	dt	io	ps	t
4700	18	10000	10000	1000	10	1	∞	0.5	500	10	$3[min]$



Results - Real Data - Husky a200 ROS Simulation

n_p	n_l	i_{op}	i_{oa}	i_{lp}	it	k_w	χ	dt	io	ps	t
4700	18	10000	10000	1000	10	1	∞	0.5	500	10	$3[min]$

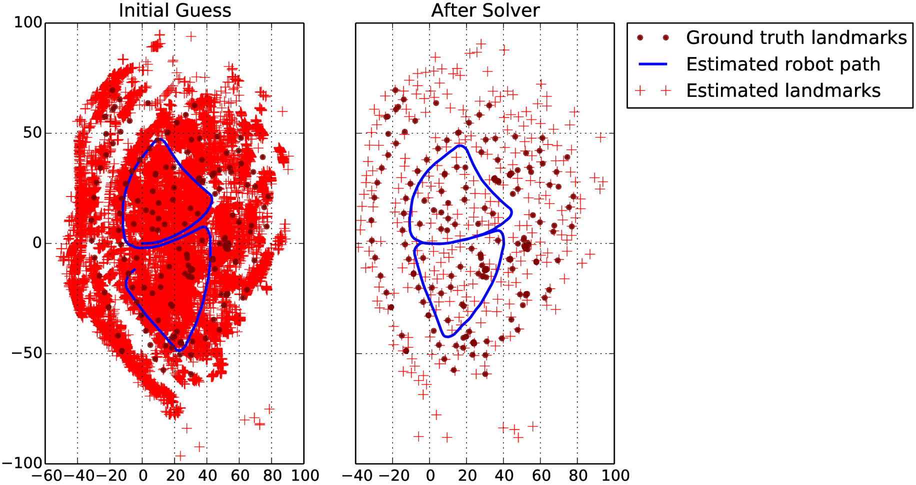


Results - Real Data - Parque O'Higgins

n_p	n_l	i_{op}	i_{oa}	i_{lp}	it	k_w	χ	dt	io	ps	t
8130	174	100000	100000	100	10	1	∞	3	500	10	$4[hrs]$

Results - Real Data - Parque O'Higgins

n_p	n_l	i_{op}	i_{oa}	i_{lp}	it	k_w	χ	dt	io	ps	t
8130	174	100000	100000	100	10	1	∞	3	500	10	$4[hrs]$

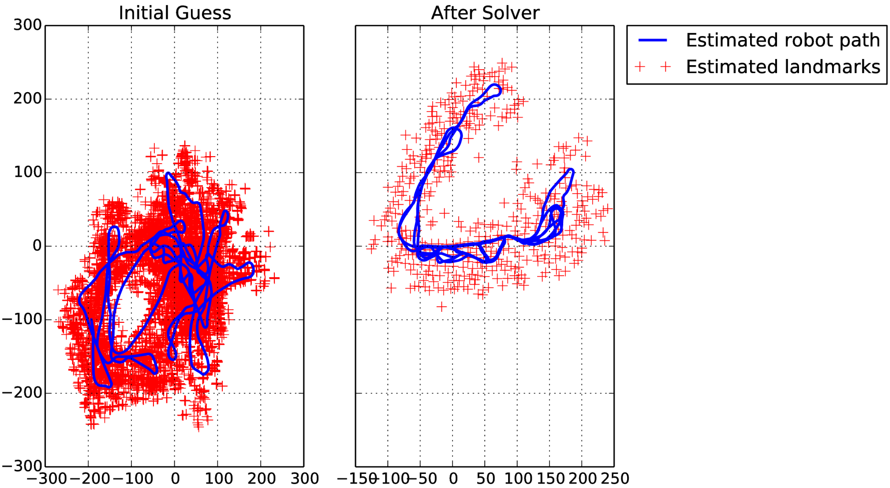


Results - Real Data - Victoria Park

n_p	n_l	i_{op}	i_{oa}	i_{lp}	it	k_w	χ	dt	io	ps	t
61763	-	8000	100000	5	15	1	∞	5	500	10	$10[hrs]$

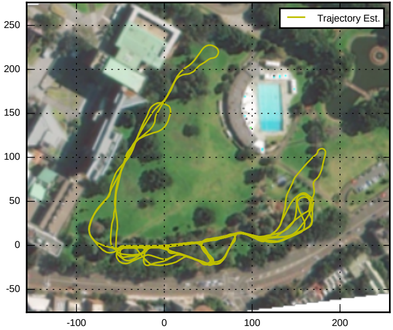
Results - Real Data - Victoria Park

n_p	n_l	i_{op}	i_{oa}	i_{lp}	it	k_w	χ	dt	io	ps	t
61763	-	8000	100000	5	15	1	∞	5	500	10	$10[hrs]$



Results - Real Data - Victoria Park

n_p	n_l	i_{op}	i_{oa}	i_{lp}	it	k_w	χ	dt	io	ps	t
61763	-	8000	100000	5	15	1	∞	5	500	10	$10[hrs]$



GraphSLAM

iSAM

Conclusions

- GraphSLAM was successfully implemented for the 2D scenario using $\mathbf{g}^2\mathbf{o}$.
- A **correspondence test** was implemented to deal with the correspondence problem.
- The **distant test** is used to speed up the algorithm.
- The implementation works in the **known** and **unknown data association**, for **simulated** and **real data**.
- The biggest drawbacks are the **computational time** and the **false positives**.
- **Repository**: `https://github.com/francocurotto/GraphSLAM`.

References

- Thrun, Sebastian, and Michael Montemerlo. "The graph SLAM algorithm with applications to large-scale mapping of urban structures." *The International Journal of Robotics Research* 25.5-6 (2006): 403-429.
- Kümmerle, Rainer, et al. "g 2 o: A general framework for graph optimization." *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011.
- Kaess, Michael, Ananth Ranganathan, and Frank Dellaert. "iSAM: Incremental smoothing and mapping." *Robotics, IEEE Transactions on* 24.6 (2008): 1365-1378.
- Thrun, Sebastian, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

Ingeniería Eléctrica
FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Thesis Defense

GraphSLAM Algorithm Implementation for Solving Simultaneous Localization and Mapping

Author: Franco Curotto
Thesis Adviser: Martin Adams
Commission Members: Marcos Orchard
Jorge Silva

Departamento de Ingeniería Eléctrica
Facultad de Ciencias Físicas y Matemáticas
Universidad de Chile

April 18, 2016