Definition and modelisation of the DigSLAM Problem

V. Radwan

LIRMM

Contents

Definition

The DigSLAM problem is a SLAM (Simultaneous Localization And Mapping) problem

Aim:

Localize, i.e. get knowledge over the state $\boldsymbol{x}(\cdot)$

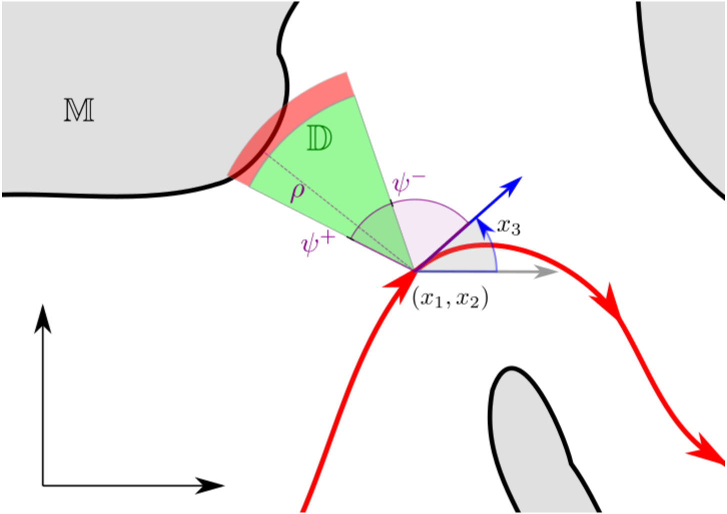
Map, i.e. get knowledge over the map $\mathbb{M}$

Tools:

Range only sensor (ultrasonic rangefinder): gives the range of the closest obstacle within scope (cone)

Proprioceptive sensors: gives an enclosure of the derivative of the state $\dot{\boldsymbol{x}}(\cdot)$

Illustration of one observation



Variables and parameters

- $\boldsymbol{x}(\cdot)$: trajectory/state of the robot
- $\mathbb{M}$: 2D Cartesian occupancy map of the cave
- $\mathbb{D}(t_i)$: 2D area guaranteed free from obstacles at observation i expressed in Cartesian system,
- $(\rho, \theta)(t_i)$: polar coordinates of detected obstacle at observation i
- $[\psi](t_i)$: opening angle of the sensor at observation i
- $([\rho], [\![\psi]\!])(t_i)$ the information given by the sensor at observation i
- $[\boldsymbol{u}](\cdot)$: approximation of the control $\boldsymbol{u}(\cdot)$ (proprioceptive sensors)
- $\mathbb{X}_0$: set of initial conditions

Contents

The DigSLAM Problem

Modeling as a dynamical system:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) & \\ (\rho(t_i), \theta(t_i)) \in \mathbf{g}_{\boldsymbol{x}(t_i)}(\mathbb{M}) & \forall t_i \in \mathbb{T} \\ \boldsymbol{x}(0) \in \mathbb{X}_0 & \end{cases}$$

where:

- $\mathbb{T}$ is the set of timestamps of observations
- $\mathbf{f}$ is the evolution function
- $\mathbf{g}_{\boldsymbol{x}}$ is the observation function:
  a change of basis from Cartesian to polar centered in $\boldsymbol{x}$

Constraint Network

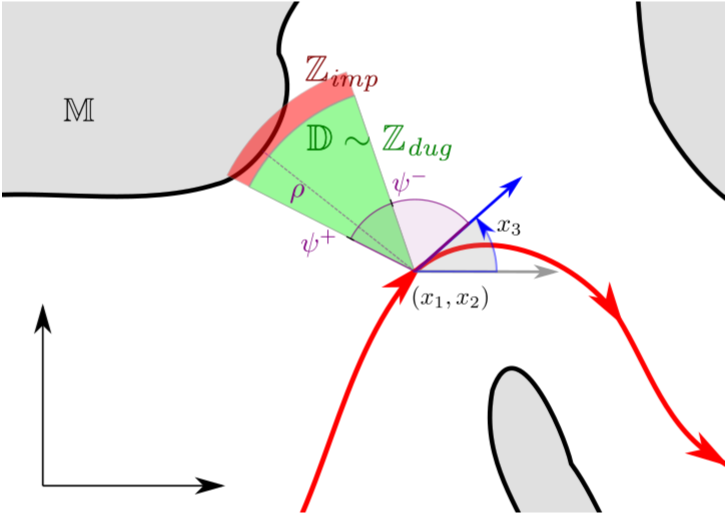
Modeling as a CSP (Constraint Satisfaction Problem):

$$\left\{\begin{array}{l}
\textbf{Variables: } \boldsymbol{x}(\cdot),\ \mathbb{M},\ \mathbb{D}(t_i),\ \rho(t_i),\ [\psi](t_i) \\
\textbf{Constraints:} \\
\quad 1.\quad \dot{\boldsymbol{x}}(t) = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \qquad (\mathcal{L}_{evo}) \\
\quad 2.\quad \mathbb{Z}_{imp}(t_i) \cap \mathbf{g}_{\boldsymbol{x}(t_i)}(\mathbb{M}) \neq \emptyset \qquad \forall t_i \in \mathbb{T} \quad (\mathcal{L}_{imp}) \\
\quad 3.\quad \mathbb{Z}_{dug}(t_i) = \mathbf{g}_{\boldsymbol{x}(t_i)}(\mathbb{D}(t_i)) \qquad \forall t_i \in \mathbb{T} \quad (\mathcal{L}_{\mathbf{g}=}) \\
\quad 4.\quad \mathbb{D}(t_i) \cap \mathbb{M} = \emptyset \qquad \forall t_i \in \mathbb{T} \quad (\mathcal{L}_{\cap=\emptyset}) \\
\quad 5.\quad \boldsymbol{x}(0) \in \mathbb{X}_0 \\
\textbf{Domains: } [\boldsymbol{x}](\cdot),\ [\mathbb{M}],\ [\mathbb{D}](t_i),\ [\rho](t_i),\ [\![\psi]\!](t_i)
\end{array}\right.$$

where:

- $\mathbb{Z}_{imp} = (\rho, [\psi])$ is the impact zone in polar coordinates
- $\mathbb{Z}_{dug} = ([0, \rho], [\psi])$ is the dug zone in polar coordinates

Illustration of the problem



Domains of variables

Variables		Domains	
Real number	$x \in \mathbb{R}$	Interval	$[x] \in \mathbb{IR}$
Vector	$\boldsymbol{x} \in \mathbb{R}^n$	Box	$[\boldsymbol{x}] \in \mathbb{IR}^n$
Trajectory	$\boldsymbol{x}(\cdot) \in \mathcal{F}(\mathbb{R}, \mathbb{R}^n)$	Tube	$[\boldsymbol{x}](\cdot) \in \mathbb{I}\mathcal{F}(\mathbb{R}, \mathbb{R}^n)$
Set	$\mathbb{X} \in \mathcal{P}(\mathbb{R}^n)$	Thick set	$[\mathbb{X}] \in \mathbb{I}\mathcal{P}(\mathbb{R}^n)$

$[\mathbb{X}] = [\mathbb{X}^-, \mathbb{X}^+]$ s.t. $\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+$

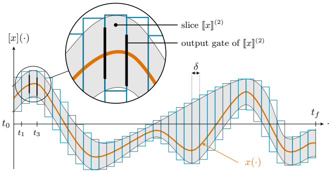


Figure: Tube

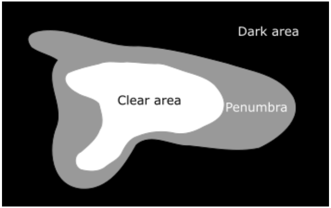


Figure: Thick set

Contents

Contractor for $\mathcal{L}_{evo}$

$$\mathcal{L}_{evo}: \quad \dot{\boldsymbol{x}}(t) = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$$

$\mathcal{C}_{evo}$ contracts $[\boldsymbol{x}]$ with guaranteed integration

Example of methods to compute guaranteed integration: Picard iterations, Lohner method

Contractor for $\mathcal{L}_{imp}$

$$\mathcal{L}_{imp}: \quad \mathbb{Z}_{imp}(t_i) \cap \mathbf{g}_{\boldsymbol{x}(t_i)}(\mathbb{M}) \neq \emptyset$$

$$\mathcal{C}_{imp}\begin{pmatrix} [\boldsymbol{x}] \\ [\mathbb{M}] \\ [\mathbb{Z}_{imp}] \end{pmatrix} = \begin{pmatrix} \left[proj_{\boldsymbol{x}}\left\{([\boldsymbol{x}] \times \mathbb{M}^{+}) \cap \mathbf{g}^{-1}(\mathbb{Z}^{+}_{imp})\right\}\right] \\ \left[\mathbb{M}^{-}, proj_{\mathbb{M}}\left\{([\boldsymbol{x}] \times \mathbb{M}^{+}) \cap \mathbf{g}^{-1}(\mathbb{Z}^{+}_{imp})\right\}\right] \\ [\mathbb{Z}_{imp}] \end{pmatrix}$$

Basic operations: Thick set inversion ($\mathbf{g}^{-1}(\mathbb{Z}^{+}_{imp})$), thick set intersection

Contractor for $\mathcal{L}_{\mathbf{g}=}$

$$\mathcal{L}_{\mathbf{g}=} : \quad \mathbb{Z}_{dug}(t_i) = \mathbf{g}_{\boldsymbol{x}(t_i)}(\mathbb{D}(t_i))$$

$$\mathcal{C}_{\mathbf{g}=}\begin{pmatrix}[\boldsymbol{x}]\\ [\mathbb{D}]\\ [\mathbb{Z}_{dug}]\end{pmatrix} = \begin{pmatrix}[\boldsymbol{x}]\\ [\mathbb{D}] \sqcap [\mathbf{g}^{-1}_{[\boldsymbol{x}]}](\mathbb{Z}_{dug})\\ [\mathbb{Z}_{dug}] \sqcap [\mathbf{g}_{[\boldsymbol{x}]}](\mathbb{D})\end{pmatrix}$$

Basic operations: Thick set inversion, thick set squared intersection

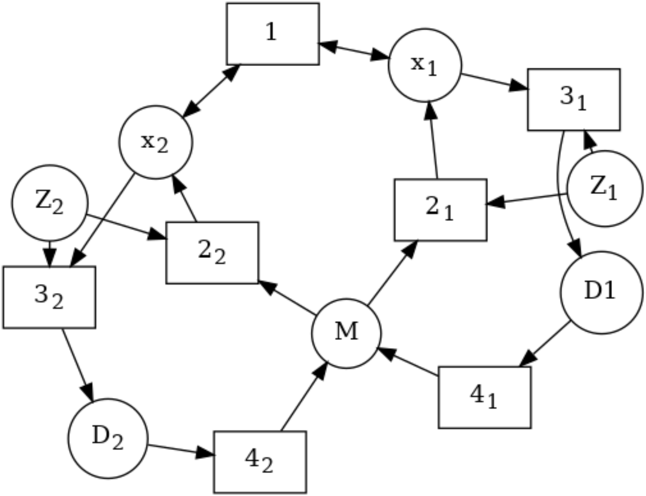
Contractor for $\mathcal{L}_{\cap=\emptyset}$

$$\mathcal{L}_{\cap=\emptyset} : \quad \mathbb{D}(t_i) \cap \mathbb{M} = \emptyset$$

$$\mathcal{C}_{\cap=\emptyset}\begin{pmatrix}[\mathbb{D}]\\ [\mathbb{M}]\end{pmatrix} = \begin{pmatrix}[\mathbb{D}^-, \mathbb{D}^+ \setminus \mathbb{M}^-]\\ [\mathbb{M}^-, \mathbb{M}^+ \setminus \mathbb{D}^-]\end{pmatrix} = \begin{pmatrix}[\mathbb{D}] \sqcap ([\mathbb{D}] \setminus [\mathbb{M}])\\ [\mathbb{M}] \sqcap ([\mathbb{M}] \setminus [\mathbb{D}])\end{pmatrix}$$

Basic operations: thick set squared intersection, thick set difference

Constraint graph with 2 observations



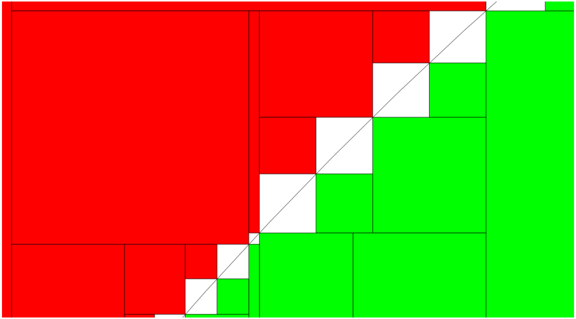
Discussion on the contraint graph

There are loops of contractions on this simple graph.
Hence, heuristics play a big role on the resolution.

Sets

- Let $\mathbb{X}$ be a set
- $\mathbb{X}$ represented as a paving, implemented as a binary tree
- Nodes represent a part of the search space as a box $[\boldsymbol{x}]$, and give them a state
- 3 states are possible:
  - "in" if $[\boldsymbol{x}] \subset \mathbb{X}$
  - "out" if $[\boldsymbol{x}] \subset \overline{\mathbb{X}}$
  - "maybe" otherwise

Visualisation of a set



Only the leaves are represented. The inside leaves are green, the outside ones are red, while the leaves that contains the frontier are white.

Thick sets

- Let $[\mathbb{X}] = [\mathbb{X}^-, \mathbb{X}^+]$ be a thick set
- $[\mathbb{X}]$ represented as a paving, implemented as a binary tree
- Nodes represent a part of the search space as a box $[\boldsymbol{x}]$, and give them a state
- States are determined with 3 Booleans:
  - in = true iff $[\boldsymbol{x}] \cap \mathbb{X}^- \neq \emptyset$
  - out = true iff $[\boldsymbol{x}] \cap \overline{\mathbb{X}^+} \neq \emptyset$
  - maybe = true iff $[\boldsymbol{x}] \cap (\mathbb{X}^+ \setminus \mathbb{X}^-) \neq \emptyset$
- States can also be expressed as words of 3 bits, hence 8 different words

Contents

Contractions on the measures

Possibility to contract the measures $[\rho], [\![\psi]\!]$

Since the measures sets $\mathbb{Z}_{imp}, \mathbb{Z}_{dug}$ are thick boxes, contractors for $\mathcal{L}_{imp}, \mathcal{L}_{\mathbf{g}=}$ can be tuned to contract the measures

Cross-observations constraints

- Small uncertainty is added between consecutive observations
- Constraints linking $x(t_i)$ and $\mathbb{Z}_{\mathsf{dug}}(t_j)$ can be considered, with t_i, t_j close enough
- Involves $\mathbf{g}_{\boldsymbol{x},\boldsymbol{y}}$: a change of basis from polar centered in $\boldsymbol{x}$ to polar centered in $\boldsymbol{y}$

Identifiable obstacles

- Idea: add the angle $\theta(t_i)$ of the obstacle to the modeling
- A box enclosing the obstacle can be built
- Useful for obstacles proved to be detected several times (i.e. stalactite tip)

Desrochers' model

$$\left\{\begin{array}{l}
\textbf{Variables: } \boldsymbol{x}(\cdot),\ \mathbb{M},\ \rho(t_i),\ [\psi](t_i) \\
\textbf{Constraints:} \\
\quad \begin{array}{llll}
1. & \dot{\boldsymbol{x}}(t) = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) & & (\mathcal{L}_{evo}) \\
2. & \mathbb{Z}_{imp}(t_i) \cap \mathbf{g}_{\boldsymbol{x}(t_i)}(\mathbb{M}) \neq \emptyset & \forall t_i \in \mathbb{T} & (\mathcal{L}_{imp}) \\
3. & \mathbb{Z}_{dug}(t_i) \subset \mathbf{g}_{\boldsymbol{x}(t_i)}(\overline{\mathbb{M}}) & \forall t_i \in \mathbb{T} & (\mathcal{L}_{dug}) \\
4. & \boldsymbol{x}(0) \in \mathbb{X}_0 & &
\end{array} \\
\textbf{Domains: } [\boldsymbol{x}](\cdot),\ [\mathbb{M}],\ [\rho](t_i),\ [\![\psi]\!](t_i)
\end{array}\right.$$

Comparison between both models

In short: Desrochers' model links the observations directly to the map, without considering $\mathbb{D}(t_i)$ sets

Pros:

- Less memory space used (fewer thick sets)
- Less computing time (fewer contractors, fewer thick sets)

Cons:

- Some Cartesian data are not computed, like the *unexplored area*
  Possible to obtain with a second thick set $\mathbb{U}$ and other constraints:

$$3'.\quad \mathbf{g}_{\boldsymbol{x}(t_i)}(\mathbb{U}) \subset \overline{\mathbb{Z}_{dug}(t_i)} \qquad \forall t_i \in \mathbb{T} \qquad (\mathcal{L}_{unex})$$

With intermediate variables

Alternative way to compute the $\mathbb{D}$ thick sets:

Decomposition of the constraint $\mathcal{L}_{\mathbf{g}=}$ into:

$$\begin{cases} 3.1. \quad \mathbb{D}(t_i) = \mathcal{R}(x_3(t_i))\,\mathbb{W}(t_i) + (x_1(t_i),\, x_2(t_i))^\intercal & \forall t_i \in \mathbb{T} \\ 3.2. \quad \mathbb{W}(t_i) = \boldsymbol{\pi}(\mathbb{Z}_{dug}(t_i)) & \forall t_i \in \mathbb{T} \end{cases}$$

where:
$\mathcal{R}(x_3(t_i))$ is the rotation matrix of angle $x_3(t_i)$,
$\boldsymbol{\pi}$ is the Cartesian to polar function.

With intermediate variables, pros and cons

Pros:

- Does not require to compute $\mathbb{W}$ when $\boldsymbol{x}(\cdot)$ is contracted
- Use an already built contractor (ctc polar), proved to be minimal

Cons:

- Introduce wrapping effect with the rotation matrix
- This problem can be prevented if 3.2 is modified to consider the heading $x_3(t_i)$
- But, a modification of $x_3(t_i)$ leads to a contraction of $\mathbb{W}(t_i)$

Incrementality and space/time efficiency

- Methods need to be incremental to be consistent with a conrtactor network
- Contractions on a thick set seems to be tedious...
- ...and lead to more bisection thus more boxes
- Bloating in space and time
- Methods to reduce the number of nodes of the thick set and keep the same precision are needed