# Numerical Techniques for Robot Path Planning: Artificial Potential Fields and Proper Generalized Decomposition

Pau Hernando Marmol

Facultat de Matemàtiques i Informàtica
UB

June 2024

# Table of Contents

# Table of Contents

# Artificial Potential Field

A **gradient system** or **potential field** on an open set $\Omega \subset \mathbb{R}^n$ is a system of differential equations of the form

$$\dot{q} = -\nabla U(q), \quad q \in \Omega,$$

where $U : \Omega \to \mathbb{R}$ is a $\mathcal{C}^2(\Omega)$ potential function and

$$\nabla U = \left( \frac{\partial U}{\partial x_1}, ..., \frac{\partial U}{\partial x_n} \right)$$

is the gradient vector field, $\nabla U : \Omega \to \mathbb{R}^n$, of $U$.

# Artificial Potential Field

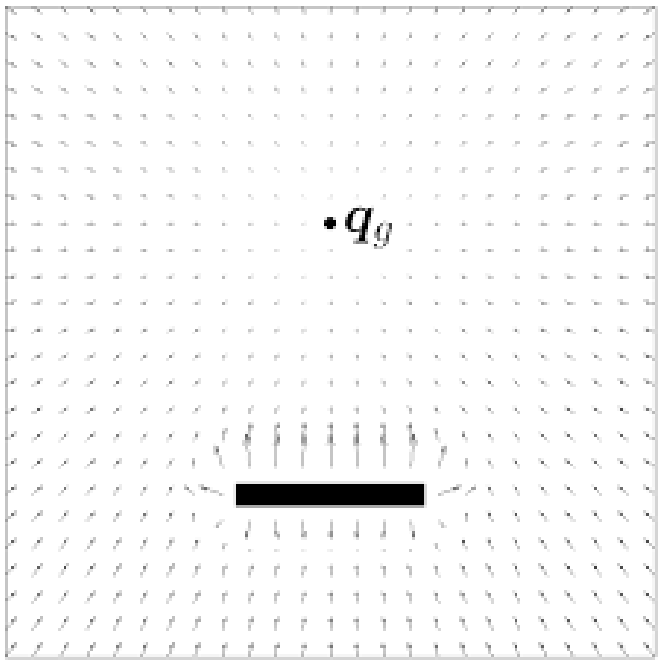
The potential field $U$ is constructed as the sum of two elementary potential functions:

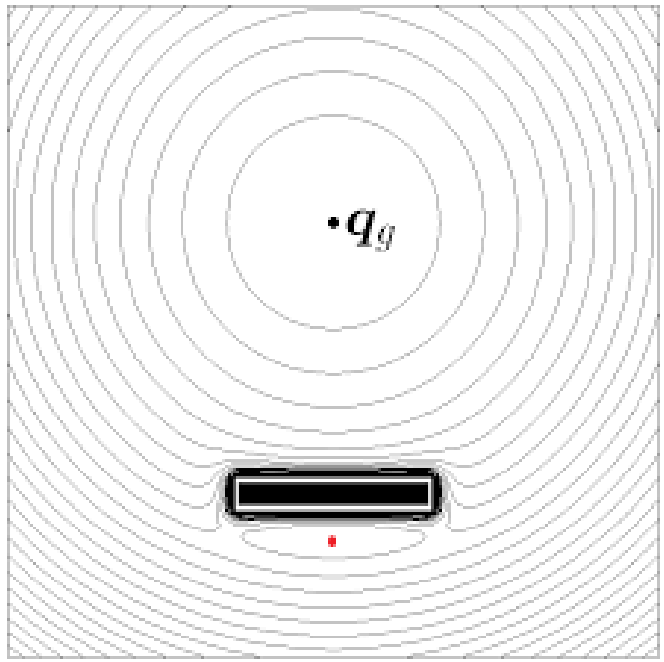
$$U(q) = U_{att}(q) + U_{rep}(q), \tag{1}$$

where $U_{att}$ is the attractive potential associated with the goal configuration $q_{goal}$ and $U_{rep}$ is the repulsive potential, associated with the $C - obstacle$ region.

# Artificial Potential Field

The major problem with the APF is its lack of completeness, since the robot can get stuck at a local minima, and even if there exists a path, the algorithm shall not find it. This happens if there exists a region where the attractive and repulsive forces cancel each other out, and therefore the resultant force is 0.



# Table of Contents

# Harmonic Functions

An **harmonic function** $\phi \in \mathcal{C}^2$ on a domain $\Omega \in \mathbb{R}^n$ is a function which satisfies Laplace's equation:

$$\Delta\phi = \nabla^2\phi = \sum_{i=1}^{n} \frac{\partial^2 \phi}{\partial x_i^2} = 0 \tag{2}$$

# Properties of Harmonic Functions

There are a few properties of a harmonic functions that make them ideal for constructing artificial potential fields for obstacle avoidance:

- **Principle of superposition** if $\phi_1$ and $\phi_2$ are harmonic, then a linear combination of $\phi_1$ and $\phi_2$ is also harmonic and a solution of Laplace equation

# Properties of Harmonic Functions

There are a few properties of a harmonic functions that make them ideal for constructing artificial potential fields for obstacle avoidance:

- **Principle of superposition** if $\phi_1$ and $\phi_2$ are harmonic, then a linear combination of $\phi_1$ and $\phi_2$ is also harmonic and a solution of Laplace equation
- **The Mean-Value Property** Suppose $\Omega$ is connected, $u$ is real valued and harmonic on $\Omega$. Then

$$u(x) = \fint_{\partial B(x,r)} u(y) \quad dy \tag{3}$$

$$u(x) = \fint_{B(x,r)} u(y) \quad dy \tag{4}$$

for $\forall x \in \Omega$ and $r > 0$ such that $B(x, r) \subset \Omega$.

# Properties of Harmonic Functions

There are a few properties of a harmonic functions that make them ideal for constructing artificial potential fields for obstacle avoidance:

- **Principle of superposition** if $\phi_1$ and $\phi_2$ are harmonic, then a linear combination of $\phi_1$ and $\phi_2$ is also harmonic and a solution of Laplace equation
- **The Mean-Value Property** Suppose $\Omega$ is connected, $u$ is real valued and harmonic on $\Omega$. Then

$$u(x) = \fint_{\partial B(x,r)} u(y) \quad dy \tag{3}$$

$$u(x) = \fint_{B(x,r)} u(y) \quad dy \tag{4}$$

  for $\forall x \in \Omega$ and $r > 0$ such that $B(x, r) \subset \Omega$.
- **The Maximum Principle** Suppose $\Omega$ is connected, $u$ is real valued and harmonic on $\Omega$, and $u$ has a maximum or a minimum in $\Omega$. Then $u$ is constant.

# Boundary Conditions

The different kinds of contour conditions imposed to Laplace's equation have a critical importance in the solution of the equation and the quality of the trajectory that will follow the robot.

**Dirichlet**

$$\phi|_{\partial\Omega} = c, \quad c \in \mathbb{R}$$

**Neumann**

$$\nabla\phi|_{\partial\Omega} = c, \quad c \in \mathbb{R}$$

# Potential Flow Theory

A localized fluid source (or a sink) can be modeled by a **Dirac term** ($\delta$) added to the right hand side of the Laplace equation. Assuming a unit amount of fluid injected at point $S$ during a unit of time and the same unit withdrawn at point $T$, the velocity potential is now solution of the Poisson equation:

$$-\nabla^2\phi = \delta_S - \delta_T, \tag{5}$$

where $\delta_S$ means the fluid source and $\delta_T$ the target sink.

# Table of Contents

# Poisson equation

Let us consider the **Poisson problem** posed in a domain $\Omega$, an relatively compact subset of $\mathbb{R}^d, d \geq 1$ supplemented with homogeneous Dirichlet boundary conditions:

$$\begin{aligned} -\Delta u(x) &= f(x), \quad \forall x \in \Omega \\ u(x) &= 0, \quad \forall x \in \partial\Omega \end{aligned} \tag{6}$$

with $f \in \mathcal{C}^0(\overline{\Omega}), \quad \overline{\Omega} = \partial\Omega \cup \Omega.$

# Weak formulation

Let's consider the **Lebesgue space**

$$L^n(\Omega) = \{u : \int_\Omega |u(x)|^n dx < \infty\}$$

and $L^n_{loc}(\Omega) = \{\phi : \mathbb{R}^n \to \mathbb{R} | \quad \phi \in L^1(K) \text{ for all compact sets } K \subset \Omega\}$.

# Weak formulation

## Definition

A function $u \in L^1_{loc}(\mathbb{R}^n)$ is **weakly differentiable** with respect to $x_i$ if there exists a function $g_i \in L^1_{loc}(\mathbb{R}^n)$ such that

$$\int_{\mathbb{R}^n} u\partial_i\phi \quad dx = -\int_{\mathbb{R}^n} g_i\phi \quad dx, \forall \phi \in \mathcal{C}^\infty_c(\mathbb{R}^n)$$

where
$\mathcal{C}^\infty_c(\mathbb{R}^n) := \{\phi : \mathbb{R}^n \to \mathbb{R} | \quad \phi \in \mathcal{C}^\infty(\mathbb{R}^n), \text{and } \phi \text{ has compact support}\}$.
The function $g_i$ is called the weak ith-partial derivative of $u$ and is denoted by $\partial_i u$.

# Weak formulation

Let's cosider the **Sobolev space**

$$H^1(\Omega) = \{u \in L^2(\Omega) : Du \in L^2(\Omega)\},$$

The homogeneous Dirichlet condition is embedded in the function space of the solution: $u$ vanishing on the boundary $\partial\Omega$ yields that we should seek $u$ in $H_0^1(\Omega)$, the **compact support subspace** of $H^1(\Omega)$.

# Weak formulation

## Definition

The **weak formulation** for (6) search $u \in H_0^1(\Omega)$ satisfying:

$$\int_\Omega \nabla u \cdot \nabla v \quad dx = \int_\Omega fv \quad dx \quad , \forall v \in H_0^1(\Omega), \quad f \in H_0^1(\Omega)'. \qquad (7)$$

where $H_0^1(\Omega)'$ is the dual space of $H_0^1(\Omega)$

# Weak formulation

More generally, we can define the problem as finding $u$ satisfying

$$a(u, v) = L(v) \quad , \forall v \in \mathcal{V} \tag{8}$$

with $a(\cdot, \cdot)$ a continuous bilinear form on $\mathcal{V} \times \mathcal{V}$ and $L(\cdot)$ a continuous linear form on $\mathcal{V}$.
In our previous case (7), the bilinear form corresponds to

$$\begin{aligned} a: \quad & \mathcal{V} \times \mathcal{W} \to \mathbb{R} \\ & (u, v) \mapsto \int_\Omega \nabla u \cdot \nabla v \quad dx \end{aligned}$$

and the linear form to

$$\begin{aligned} L: \quad & \mathcal{V} \to \mathbb{R} \\ & v \mapsto \int_\Omega f v \quad dx \end{aligned}$$

# Weak formulation

## Riesz Representation theorem

For a continuous linear function $\phi$ on a Hilbert space $H$, there exists a unique $u \in H$ such that $\phi(v) = \langle u, v \rangle, \quad \forall v \in H$.
Furthermore, $\|u\|_H = \|\phi\|_H$

# Table of Contents

# The PGD

From now on, we consider the weak formulation of the Poisson equation

$$\int_{\Omega_x \times \Omega_y} u^* \cdot (\Delta u(x,y) - f) \quad dx \cdot dy = 0 \tag{9}$$

Now, our main goal is to obtain a Proper Generalized Decomposition approximate solution in the separated form

$$u(x,y) = \sum_{i=1}^{N} X_i(x) \cdot Y_i(y) \tag{10}$$

# Enrichment step

In a particular enrichment step n, the PGD approximation $u^{n,p}$ obtained at iteration $p$ reads as

$$u^{n,p}(x,y) = u^{n-1}(x,y) + X_n^p(x) \cdot Y_n^p(y) \tag{11}$$

The topping criterion is associated is

$$\epsilon(n) = \frac{\|X_n(x) \cdot Y_n(y)\|}{\|X_1(x) \cdot Y_1(y)\|}. \tag{12}$$

# Alternating direction step

We will break down a two-dimensional problem searching at each step the solution for a single one-dimensional direction (alternating between the $x - \textit{direction}$ and the $y - \textit{direction}$). The workflow is as follows:

$$Y_n^0 \longrightarrow X_n^1 \longrightarrow Y_n^1 \longrightarrow X_n^2 \longrightarrow Y_n^2 \longrightarrow \ldots \longrightarrow X_n^p \longrightarrow Y_n^p$$

The topping criterion is associated is

$$\frac{\|X_n^p(x) \cdot Y_n^p(y) - X_n^{p-1}(x) \cdot Y_n^{p-1}(y)\|}{\|X_n^{p-1}(x) \cdot Y_n^{p-1}(y)\|} < \epsilon \tag{13}$$

# Pseudocode

**Algorithm 1:** PGD Algorithm

**while** *solution do not converge* **do**
    Initiate the alternating values $Y_n^0$ and $X_n^1$;
    **while** *stopping criterion is not satisfied* **do**
        Compute the alternating direction strategy
        to get $Y_n^p$ and $X_n^p$
    **end**
    Add the enrichment step to the solution
**end**

# Source term

For a sought source and target points, $s$ and $t$, we will create the source term $f$ by means of Gaussian model with mean and a variance. Then, we obtain a separated representation of $f$ in the form

$$f(x,y) = \sum_{j=1}^{\mathcal{F}} F_k^x(x) \cdot F_j^y(y). \tag{14}$$

using the PGD approximation.

# Results

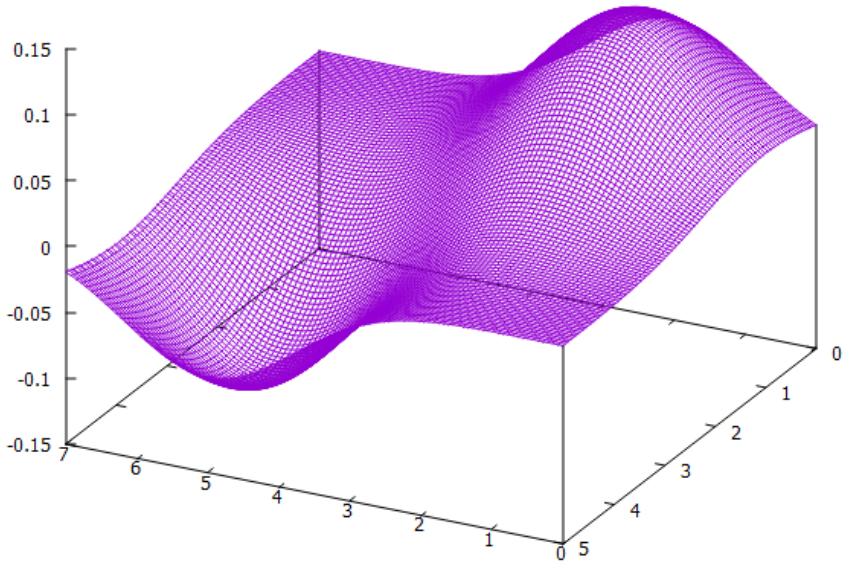
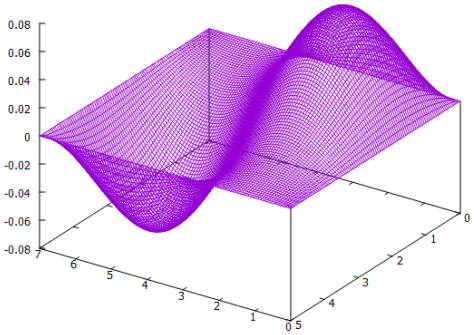
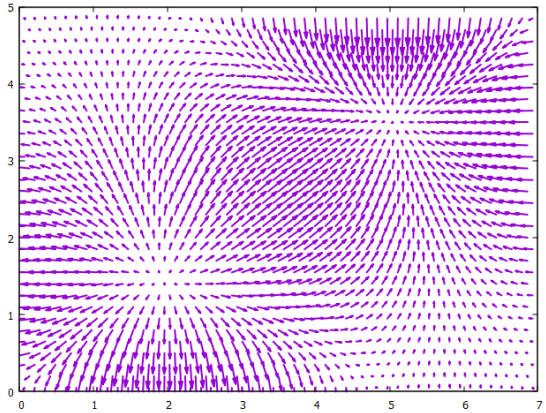
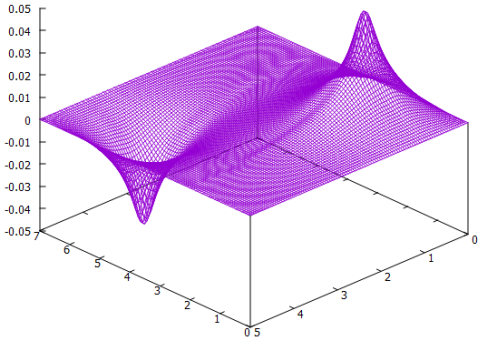


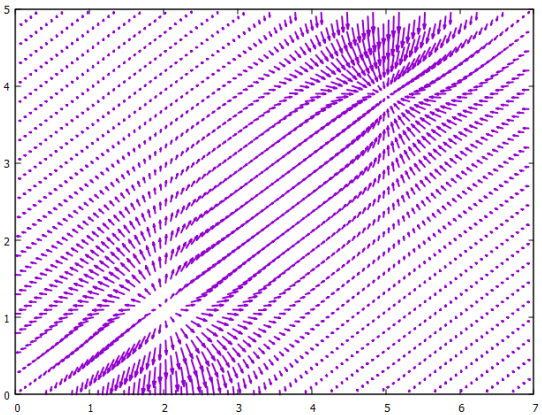

Figure: Separated form estimation of source term function $f$ for source point $S_p = (2, 1)$ and a target point $T_p = (5, 4)$

# Results


0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
7
6
5
4
3
2
1
0
0
1
2
3
4
5


5
4
3
2
1
0
0
1
2
3
4
5
6
7


0.05
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
-0.05
7
6
5
4
3
2
1
0
0
1
2
3
4
5


5
4
3
2
1
0
0
1
2
3
4
5
6
7

# Results

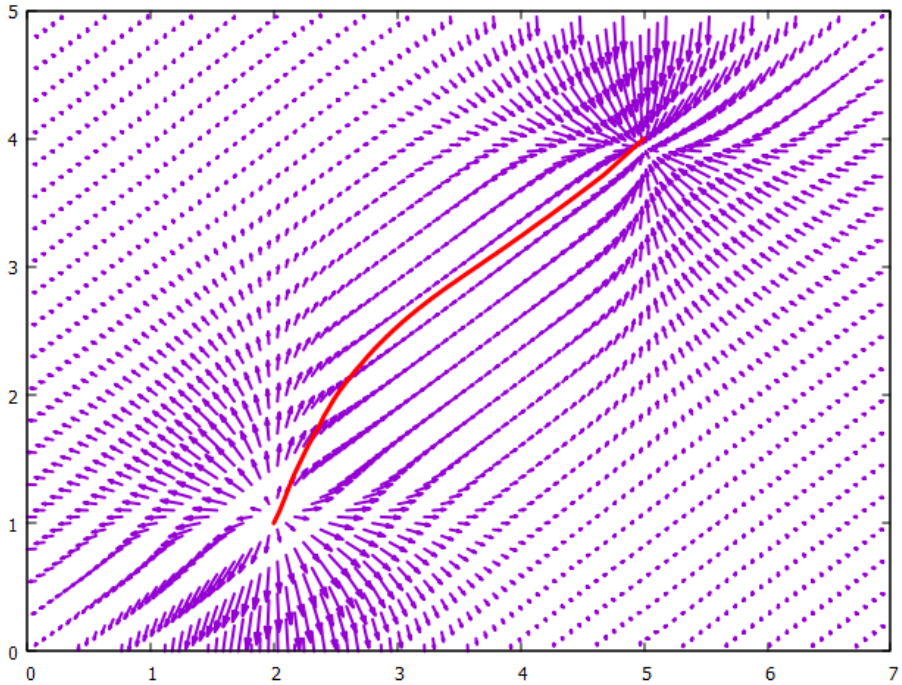



Figure: Interpolated path from source point $S_p = (2, 1)$, target point $T_p = (5, 4)$

# Table of Contents

# Robot Operating System

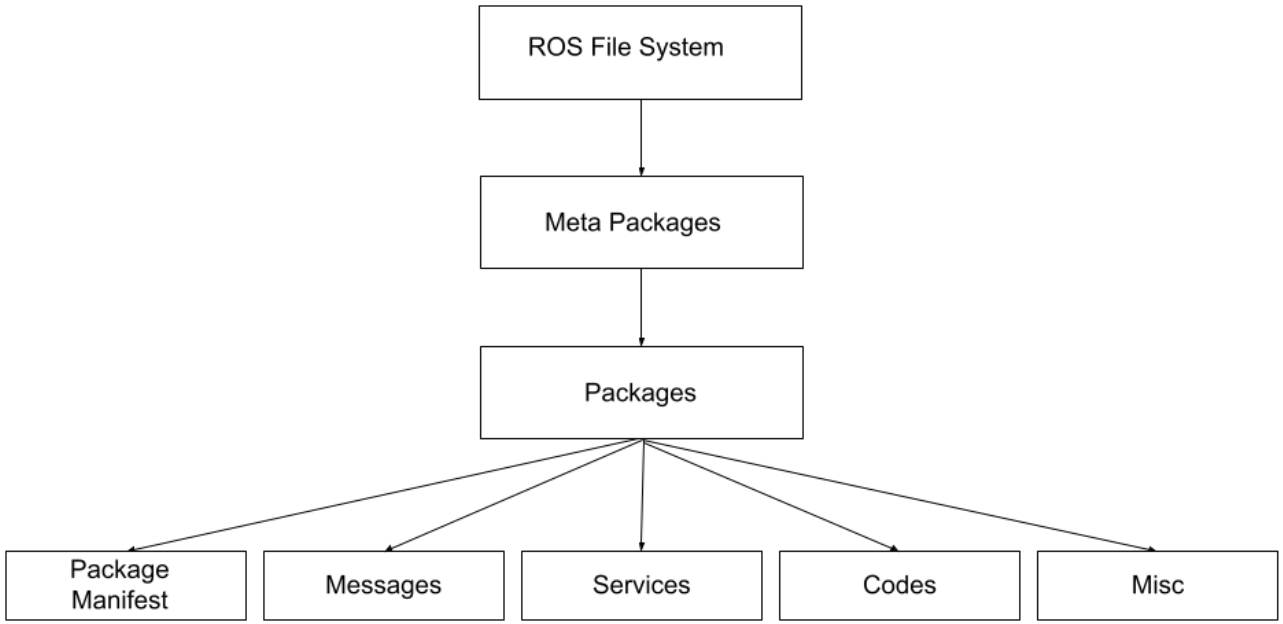



Figure: Graph representing the ROS file system hierarchy

# Computation Graph

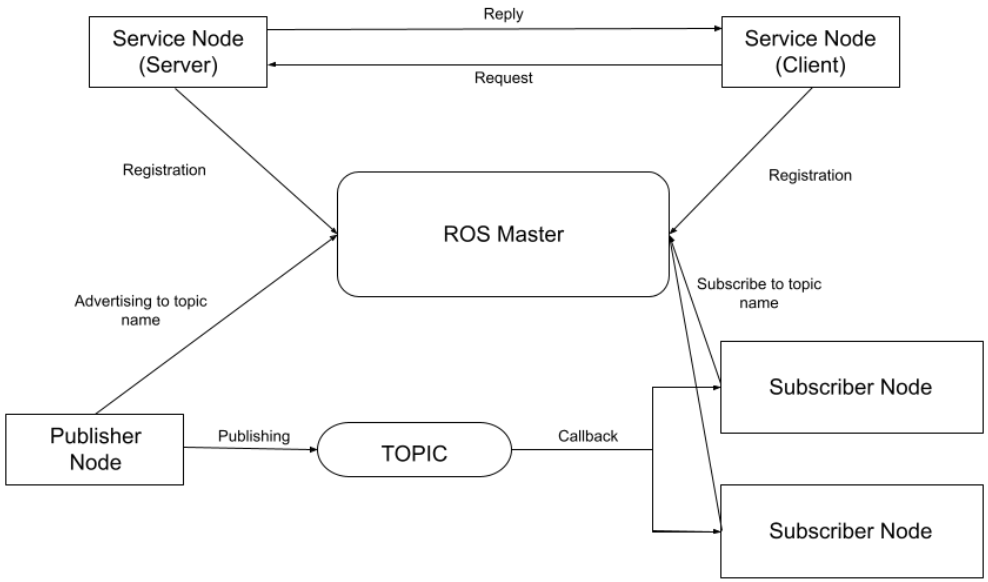



Figure: Workflow of ROS main communication processes: Topics and Services

# Navigation Stack

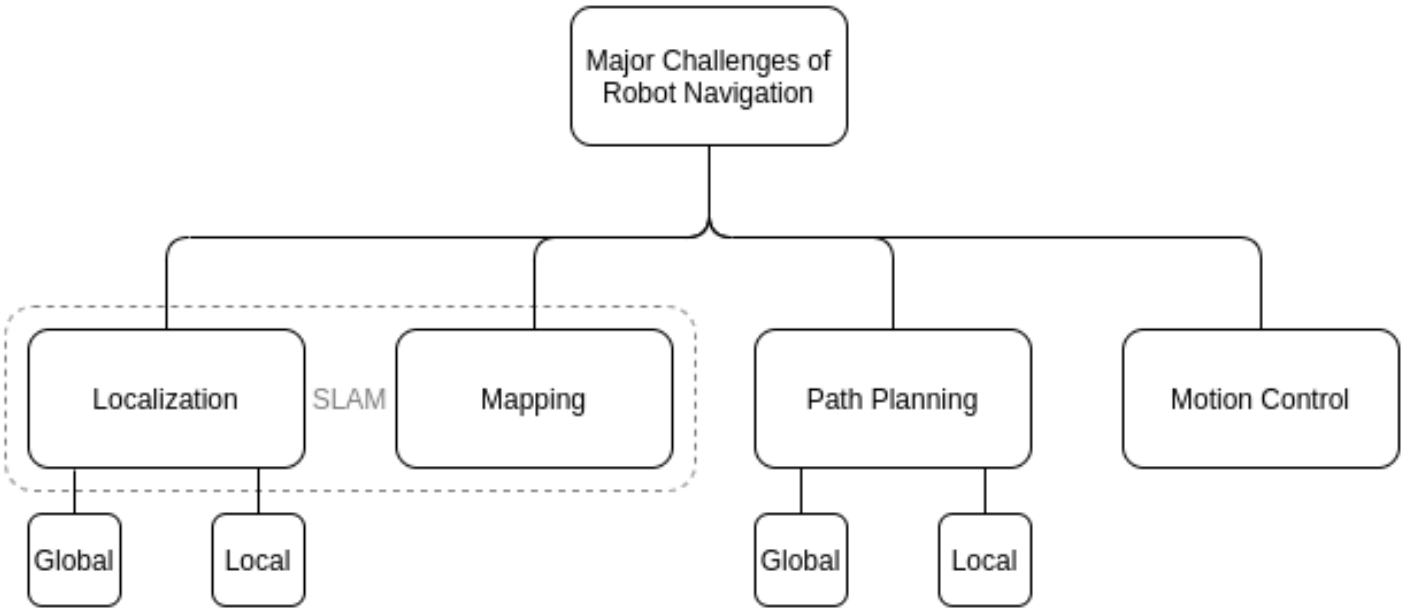



Figure: The navigation stack

# Our implementation

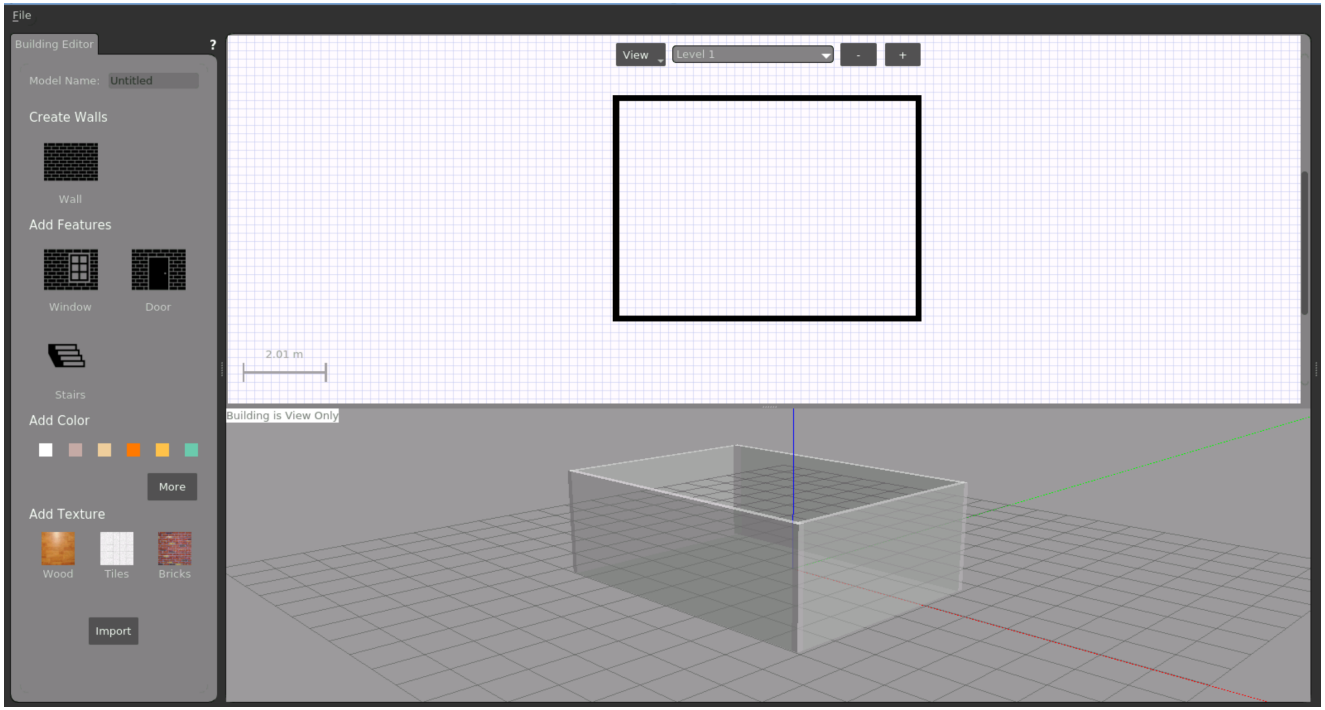



Figure: Gazebo building editor tool interface.

# Our implementation

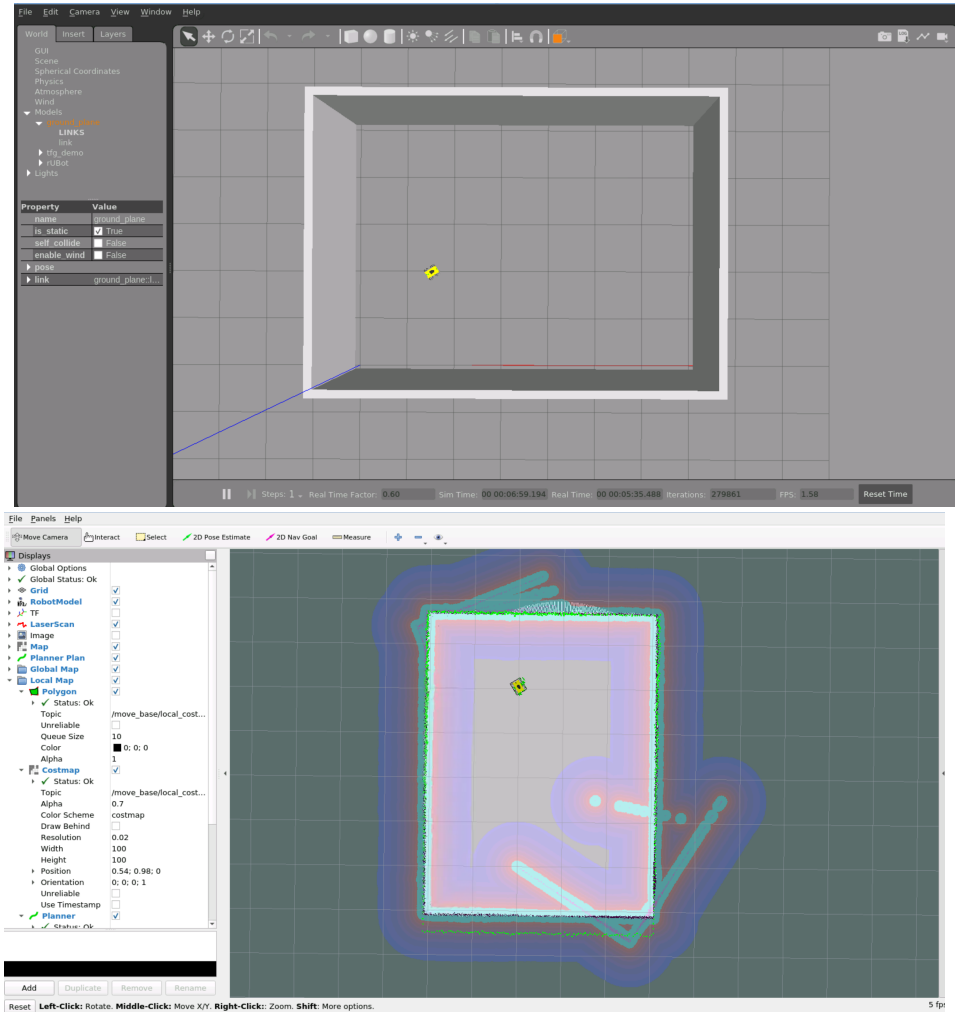



Figure: Gazebo and RVIZ map visualizations.

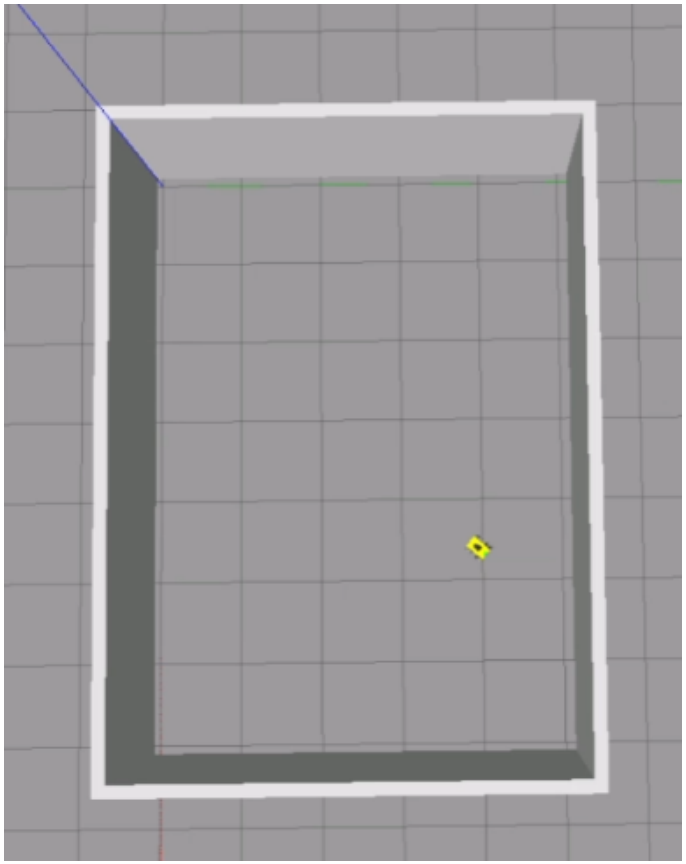
# Our implementation

```
1  #waypoint.yaml file
2  goal1: {"x": -0.5, "y": 0.8, "w": 90}
3  goal2: {"x": -0.5, "y": -0.5, "w": 180}
4  goal3: {"x": -0.5, "y": -1.3, "w": 180}
5
6  ...
```

# Our implementation

```
1        def movebase_client():
2            client = actionlib.SimpleActionClient('move_base',MoveBaseAction)
3            client.wait_for_server()
4            waypoints = []
5            with open(rospy.get_param("~waypoints_file")) as file:
6                waypoints_data = yaml.load(file, Loader=yaml.FullLoader)
7            # Process loaded waypoints
8            for goal_data in waypoints_data.values():
9                goal_pose = create_pose_stamped(goal_data['x'],
                     goal_data['y'], radians(goal_data['w']))
10               waypoints.append(goal_pose)
11           #we send a goal for each waypoint
12           for wp in waypoints:
13               max_attempts = 3
14               for attempt in range(max_attempts):
15                   client.send_goal(wp)
16                   wait = client.wait_for_result(rospy.Duration(100))
17                   if wait:
18                       rospy.loginfo("Goal execution done!")
19                       break # Goal reached successfully, exit loop
20                   else:
21                       rospy.logwarn("Failed to reach goal, retrying...")
```

# Final result



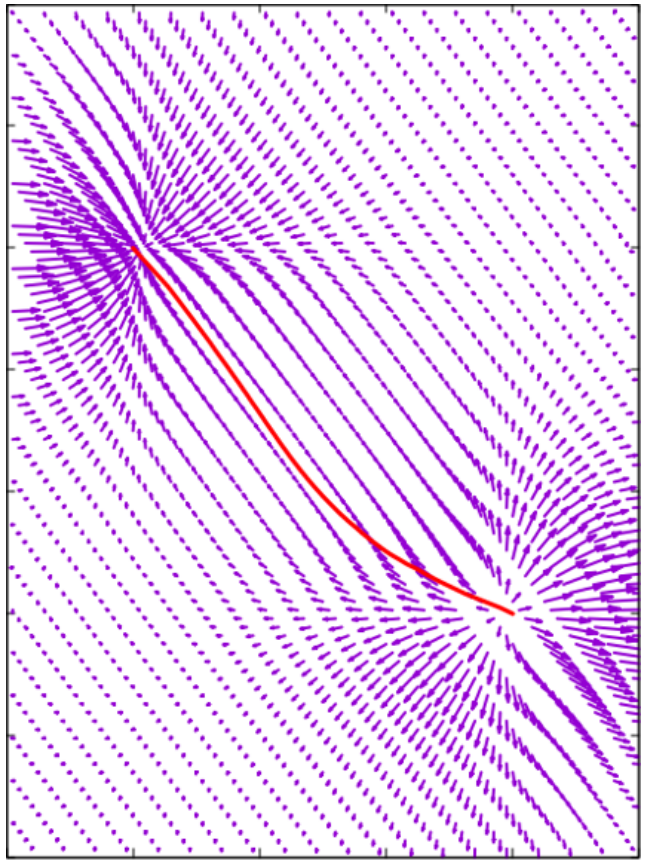


Figure: Comparation of the computed PDG path and the ROS result.

# Next steps

This work provides the perfect basis for future extensions. As for the mathematically oriented part, an interesting future continuation would be integrate dynamic objects to the Poisson equation, as originally intended.

On the other hand, translating all this work to the physical realm by leaving the simulation environment and test it on a real environment can be a good experiment.