# Simultaneous Localization and Active Phenomenon Inference (SLAPI)

**Olivier L. Georgeon** OGEORGEON@UNIV-CATHOLYON.FR
*UR Confluence, Sciences et Humanités (EA 1598) - Lyon Catholic University, France*
**Titouan Knockaert** TITOUAN.KNOCKAERT@GMAIL.COM
*Université Claude Bernard Lyon 1, LIRIS CNRS UMR5205, F-69622 Villeurbanne, France*

**Editor:** Editor's name

## Abstract

This is the abstract for this article.
**Keywords:** List of keywords

## 1. Introduction

The problem of getting mobile robots to autonomously learn the position of surrounding objects, recognize them, and keep track of their relative displacements is considered by many to be a key prerequisite of truly autonomous robots. Within this framework, the SLAM problem (Simultaneous Localization and Mapping) has been formalized and studied since the 1990s: constructing and updating a map of an unknown environment while simultaneously keeping track of the robot's position within it (e.g., Taketomi et al., 2017). SLAM algorithms are tailored to the available resources: odometric sensors, sensors of the environment, computational capacities, as well as the landmarks' properties, quantity, and dynamics, and the usage intended for the robot.

When displacements are imprecise and odometric data is not available, when landmarks are not directly identifiable, and below a certain level of scarcity and noise in the sensory data relative to the environment's complexity, it becomes difficult to perform SLAM accurately enough to use the robot for tasks involving complex navigation (Gay et al., 2021). For such robots, we propose the SLAPI problem: Simultaneous Localization and Active Phenomenon Inference. In contrast with SLAM, SLAPI does not aim at constructing a map to use for navigation. Instead, it aims at organizing behavior spatially in the vicinity of objects to design robots that exhibit intrinsic motivation (e.g., Oudeyer et al., 2007) such as playfulness and curiosity as they discover and interact with unknown objects. Possible applications may not include delivery tasks but may include entertainment and games, similar to playing with pets.

SLAPI makes no assumption that landmarks can be directly and passively uniquely identified through sensors. The robot must rather actively interact with objects, possibly from different angles and through different modalities of control loops, to categorize and recognize objects, and possibly use them as landmarks. We call this process *active phenomenon inference*, drawing from the work of Friston et al. (2021) on active inference. Here the term *phenomenon* refers to the knowledge of physical objects actively constructed by the robot from its point of view and "as the robot experiences the object through interaction" (Thórisson, 2021).

We designed a proof-of-concept algorithm to illustrate the SLAPI problem. We demonstrated it in a robot mounted on omnidirectional wheels and endowed with an echo-localization sensor, photosensitive sensors, and an inertial measurement unit, but no camera and lidar. As the robot circles around an object, it constructs the phenomenon corresponding to this object under the form of the set of the spatially-localized control loops that the object affords to the robot. New elements can be subsequently added to this set as the robot improves its knowledge of the object. Results show that the robot drew out a few cries of amusement and endearment from some human observers, which encourages us to keep improving this range of algorithms.

Moreover, we believe that the study of SLAPI problems can shed some light on how animals construct knowledge of objects through sensorimotor interactions, while keeping this knowledge grounded in experience. It can also provide an angle of attack to the more general AI problem of self-motivated open-ended learning in the real world.

## 2. The status of input data

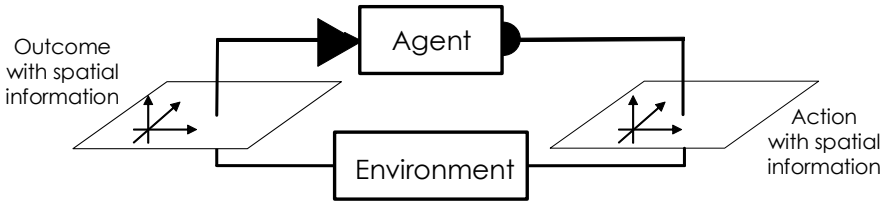


Figure 1: The interaction cycle. Black bullet: the cycle begins with the agent sending an action containing spatial information to the environment (right). Black arrowhead: the cycle ends with the agent receiving the outcome containing spatial information (left).

Let's talk about Figure 1.

## 3. The experimental setup

Let's talk about Figure 2. The data exchanged between the PC and the robot is as follows:

**PC to Robot** Action code, focus position (x, y), estimated speed (x,y).

**Robot to PC** Outcome code, echo distance, head direction, yaw, azimuth, duration.

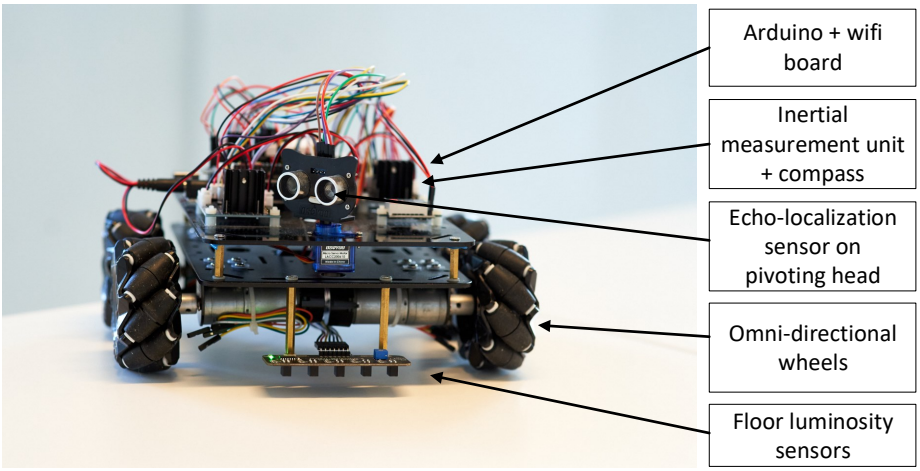


Figure 2: The robot

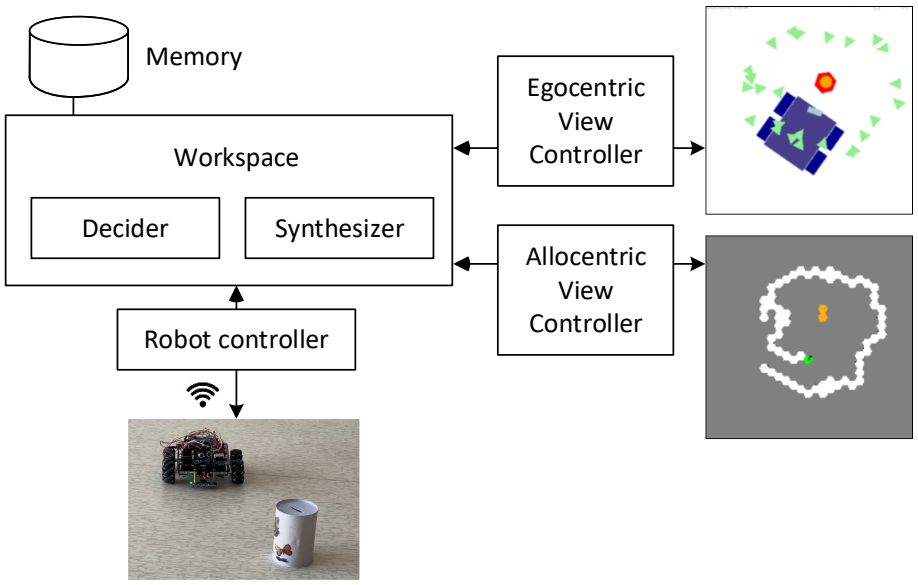


Figure 3: The software architecture. The *Workspace* plays the role of the Model in a regular Model-View-Controller architecture. It contains the *Decider* which selects the next intended interaction to send to the robot through wifi (bottom), and the *Synthesizer* which infers the phenomena. The *Memory* plays the role of the database. It contains the egocentric memory and the allocentric memory which are accessed through the *Workspace* and the *View Controllers* to display on screen (right).

## 4. The software architecture and algorithm

## 5. Equations

The jmlr class loads the amsmath package, so you can use any of the commands and environments defined there. (See the amsmath documentation for further details.[1])

Unnumbered single-lined equations should be displayed using `\[` and `\]`. For example:

$$E = mc^2$$

Numbered single-line equations should be displayed using the `equation` environment. For example:

$$\cos^2\theta + \sin^2\theta \equiv 1 \tag{1}$$

This can be referenced using `\label` and `\equationref`. For example, Equation (1).

Multi-lined numbered equations should be displayed using the `align` environment.[2] For example:

$$f(x) = x^2 + x \tag{2}$$
$$f'(x) = 2x + 1 \tag{3}$$

Unnumbered multi-lined equations should be displayed using the `align*` environment. For example:

$$\begin{aligned} f(x) &= (x+1)(x-1) \\ &= x^2 - 1 \end{aligned}$$

If you want to mix numbered with unnumbered lines use the align environment and suppress unwanted line numbers with `\nonumber`. For example:

$$\begin{aligned} y &= x^2 + 3x - 2x + 1 \\ &= x^2 + x + 1 \end{aligned} \tag{4}$$

An equation that is too long to fit on a single line can be displayed using the `split` environment. Text can be embedded in an equation using `\text` or `\intertext` (as used in Theorem 1). See the amsmath documentation for further details.

### 5.1. Operator Names

Predefined operator names are listed in Table 1. For additional operators, either use `\operatorname`, for example $\operatorname{var}(X)$ or declare it with `\DeclareMathOperator`, for example

```
\DeclareMathOperator{\var}{var}
```

and then use this new command. If you want limits that go above and below the operator (like `\sum`) use the starred versions (`\operatorname*` or `\DeclareMathOperator*`).

1. Either `texdoc amsmath` or http://www.ctan.org/pkg/amsmath
2. For reasons why you shouldn't use the obsolete `eqnarray` environment, see Lars Madsen, *Avoid eqnarray!* TUGboat 33(1):21–25, 2012.

Table 1: Predefined Operator Names (taken from amsmath documentation)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| \arccos | arccos | \deg | deg | \lg | lg | \projlim | proj lim |
| \arcsin | arcsin | \det | det | \lim | lim | \sec | sec |
| \arctan | arctan | \dim | dim | \liminf | lim inf | \sin | sin |
| \arg | arg | \exp | exp | \limsup | lim sup | \sinh | sinh |
| \cos | cos | \gcd | gcd | \ln | ln | \sup | sup |
| \cosh | cosh | \hom | hom | \log | log | \tan | tan |
| \cot | cot | \inf | inf | \max | max | \tanh | tanh |
| \coth | coth | \injlim | inj lim | \min | min | | |
| \csc | csc | \ker | ker | \Pr | Pr | | |

\varlimsup $\overline{\lim}$ \varinjlim $\varinjlim$

\varliminf $\underline{\lim}$ \varprojlim $\varprojlim$

## 6. Vectors and Sets

Vectors should be typeset using \vec. For example $\vec{x}$. The jmlr class also provides \set to typeset a set. For example $\mathcal{S}$.

### 6.1. Tables

Table 2: Data exchanged between the PC and the robot through wifi

| | |
|---|---|
| PC to Robot | Action code, focus position (x, y), estimated speed (x,y) |
| Robot to PC | Outcome code, echo distance, head direction, yaw, azimuth, duration |

### 6.2. Figures

Figures should go in the figure environment. Within this environment, use \floatconts to correctly position the caption and center the image. Use \includegraphics for external graphics files but omit the file extension. Do not use \epsfig or \psfig. If you want to scale the image, it's better to use a fraction of the line width rather than an explicit length. For example, see Figure ??.

Figure 4: The robot

### 6.3. Algorithms

Enumerated textual algorithms can be displayed using the algorithm environment. Within this environment, use use an enumerate or nested enumerate environments. For example, see Algorithm 1. Note that algorithms float like figures and tables.

Table 3: Actions available to the robot and their possible outcomes

($a$) Actions

| **Action** | **Span** |
|---|---|
| Forward<br>Backward<br>Shift left<br>Shift right | During 1 second or until line detection |
| Turn left | $\pi/4$ |
| Turn right | $-\pi/4$ |
| Head scan | $[-\pi/2, \pi/2]$ |

($b$) Outcomes

| **Outcome** | **Description** |
|---|---|
| Line left<br>Line front<br>Line right | Floor sensors cross luminosity threshold |
| Echo lost focus | No echo where expected |
| Echo left<br>Echo right<br>Echo far left<br>Echo far right<br>Echo far front<br>Echo close front | Direction and range of the nearest echo |
| Default | No line, no echo |

**Algorithm 1:** The Gauss-Seidel Algorithm

1. For $k = 1$ to maximum number of iterations
   (a) For $i = 1$ to $n$
      i. $x_i^{(k)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)}}{a_{ii}}$
      ii. If $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)} < \epsilon\|$, where $\epsilon$ is a specified stopping criteria, stop.

If you'd rather have the same numbering throughout the algorithm but still want the convenient indentation of nested `enumerate` environments, you can use the `enumerate*` environment provided by the jmlr class. For example, see Algorithm 2.

**Algorithm 2:** Moore's Shortest Path

Given a connected graph $G$, where the length of each edge is 1:

1. Set the label of vertex $s$ to 0
2. Set $i = 0$
   3. Locate all unlabelled vertices adjacent to a vertex labelled $i$ and label them $i+1$
   4. If vertex $t$ has been labelled,

      the shortest path can be found by backtracking, and the length is given by the label of $t$.

      otherwise

      increment $i$ and return to step 3

Pseudo code can be displayed using the `algorithm2e` environment. This is defined by the algorithm2e package (which is automatically loaded) so check the algorithm2e documentation for further details.[3] For an example, see Algorithm 3.

**Algorithm 3:** Computing Net Activation

**Input:** $x_1, \ldots, x_n, w_1, \ldots, w_n$
**Output:** $y$, the net activation
$y \leftarrow 0$;
**for** $i \leftarrow 1$ **to** $n$ **do**
  $y \leftarrow y + w_i * x_i$;
**end**

## 7. Description Lists

The jmlr class also provides a description-like environment called `altdescription`. This has an argument that should be the widest label in the list. Compare:

**add** A method that adds two variables.

**differentiate** A method that differentiates a function.

with

**add** A method that adds two variables.

**differentiate** A method that differentiates a function.

3. Either `texdoc algorithm2e` or http://www.ctan.org/pkg/algorithm2e

## 8. Theorems, Lemmas etc

The following theorem-like environments are predefined by the jmlr class: `theorem`, `example`, `lemma`, `proposition`, `remark`, `corollary`, `definition`, `conjecture` and `axiom`. You can use the `proof` environment to display the proof if need be, as in Theorem 1.

**Theorem 1 (Eigenvalue Powers)** *If $\lambda$ is an eigenvalue of $\boldsymbol{B}$ with eigenvector $\boldsymbol{\xi}$, then $\lambda^n$ is an eigenvalue of $\boldsymbol{B}^n$ with eigenvector $\boldsymbol{\xi}$.*
**Proof** *Let $\lambda$ be an eigenvalue of $\boldsymbol{B}$ with eigenvector $\xi$, then*

$$\boldsymbol{B\xi} = \lambda\boldsymbol{\xi}$$

*premultiply by $\boldsymbol{B}$:*

$$\begin{aligned} \boldsymbol{BB\xi} &= \boldsymbol{B}\lambda\boldsymbol{\xi} \\ \Rightarrow \boldsymbol{B}^2\boldsymbol{\xi} &= \lambda\boldsymbol{B\xi} \\ &= \lambda\lambda\boldsymbol{\xi} \qquad \textit{since } \boldsymbol{B\xi} = \lambda\boldsymbol{\xi} \\ &= \lambda^2\boldsymbol{\xi} \end{aligned}$$

*Therefore true for $n = 2$. Now assume true for $n = k$:*

$$\boldsymbol{B}^k\boldsymbol{\xi} = \lambda^k\boldsymbol{\xi}$$

*premultiply by $\boldsymbol{B}$:*

$$\begin{aligned} \boldsymbol{BB}^k\boldsymbol{\xi} &= \boldsymbol{B}\lambda^k\boldsymbol{\xi} \\ \Rightarrow \boldsymbol{B}^{k+1}\boldsymbol{\xi} &= \lambda^k\boldsymbol{B\xi} \\ &= \lambda^k\lambda\boldsymbol{\xi} \qquad \textit{since } \boldsymbol{B\xi} = \lambda\boldsymbol{\xi} \\ &= \lambda^{k+1}\boldsymbol{\xi} \end{aligned}$$

*Therefore true for $n = k + 1$. Therefore, by induction, true for all $n$.* ■

**Lemma 2 (A Sample Lemma)** *This is a lemma.*

**Remark 3 (A Sample Remark)** *This is a remark.*

**Corollary 4 (A Sample Corollary)** *This is a corollary.*

**Definition 5 (A Sample Definition)** *This is a definition.*

**Conjecture 6 (A Sample Conjecture)** *This is a conjecture.*

**Axiom 7 (A Sample Axiom)** *This is an axiom.*

**Example 1 (An Example)** *This is an example.*

## 9. Citations and Bibliography

The jmlr class automatically loads natbib. This sample file has the citations defined in the accompanying BibTeX file `pmlr-sample.bib`. For a parenthetical citation use `\citep`. For example (**?**). For a textual citation use `\citet`. For example **?**. Both commands may take a comma-separated list, for example **??**.

These commands have optional arguments and have a starred version. See the natbib documentation for further details.[4]

The bibliography is displayed using `\bibliography`.

## References

Karl Friston, Rosalyn J. Moran, Yukie Nagai, Tadahiro Taniguchi, Hiroaki Gomi, and Josh Tenenbaum. World model learning and inference. *Neural Networks*, 144:573–590, December 2021. ISSN 0893-6080. doi: 10.1016/j.neunet.2021.09.011. URL https://www.sciencedirect.com/science/article/pii/S0893608021003610.

Simon Gay, Kévin Le Run, Edwige Pissaloux, Katerine Romeo, and Christèle Lecomte. Towards a Predictive Bio-Inspired Navigation Model. *Information*, 12(3):100, March 2021. ISSN 2078-2489. doi: 10.3390/info12030100. URL https://www.mdpi.com/2078-2489/12/3/100. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V. Hafner. Intrinsic Motivation Systems for Autonomous Mental Development. *IEEE Transactions on Evolutionary Computation*, 11(2):265–286, April 2007. ISSN 1089-778X. doi: 10.1109/TEVC.2006.890271. URL http://ieeexplore.ieee.org/document/4141061/.

Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. *Visual SLAM algorithms: a survey from 2010 to 2016*. 2017. Publication Title: IPSJ Transactions on Computer Vision and Applications.

Kristinn R. Thórisson. *The 'Explanation Hypothesis' in General Self-Supervised Learning*. International Workshop in Self-Supervised Learning, 2021.

4. Either `texdoc natbib` or http://www.ctan.org/pkg/natbib