CuikSLAM with Unknown Correspondence – Preliminary Results*

Bartłomiej Jacek Kubica[1] and Wojciech Szynkiewicz[1]

[1] Warsaw University of Technology, Institute of Control and Computation Engineering, ul. Nowowiejska 15/19, 00–665 Warsaw, Poland, email: bkubica@elka.pw.edu.pl

Key words: SLAM, kinematic problem, unknown correspondence, interval computations, constraint satisfaction

Abstract. This paper considers the problem of simultaneous localization and mapping of a mobile robot. The kinematic approach of CuikSLAM is adopted applying constrained satisfaction and interval methods. The novelty is that we do not assume the landmark identification problem to be solved.

1 Introduction

SLAM, i.e. simultaneous localization and mapping problem (called also CML = concurrent mapping and localization) is a crucial task in mobile robotics. It is the problem of constructing the map of an unknown environment of the robot and localizing the robot in the partially created map.

Several approaches and several – often incompatible – formulations of the problem (see e.g. [4]) have been introduced. Main approaches may be classified as follows:

- *evidence grid approach*, where the map is divided into a grid of small cells, each of which can be occupied or free; this approach is the oldest, but some recent methods adopt it, too (e.g. [3]),
- *stochastic maps*, which are probably most commonly used; SLAM is formulated as a probabilistic estimation problem and several nonlinear variants of Kalman filter are used to estimate the features (see e.g. [4]),
- *kinematic approach* considered in our paper.

We adopt here the kinematic approach of [9], where observations from the sensors are used to generate constraints on robot poses and positions of landmarks. We solve the resulting constraint satisfaction problem to determine the set of possible locations of all objects.

It is a common assumption, however, that we can uniquely identify all of the observed landmarks. Other words, the so-called *correspondence* (between observations and landmarks; this name is used e.g. in [3], [8]) is precisely known. Obviously, this assumption is very optimistic and often may be wrong.

*This work is supported by Warsaw University of Technology Research Program grant.

2 CuikSLAM

Paper [9] describes the kinematic approach to SLAM. Authors call the algorithm presented here CuikSLAM, from the name Cuik-0 of a solver they use (which takes the name form the institute name, probably).

Coordinates (x, y, θ) of subsequent robot poses and coordinates (x, y) of subsequent landmarks are the variables. Measurements of the odometer give us some information on the distance between robot positions and other sensors (e.g. sonars) give us information on the distance between robot and landmark positions.

When the robot closes a loop, we obtain a cycle in the kinematic graph. Also when a landmark can be observed from two robot poses, a cycle can be obtained (see Figure 1).

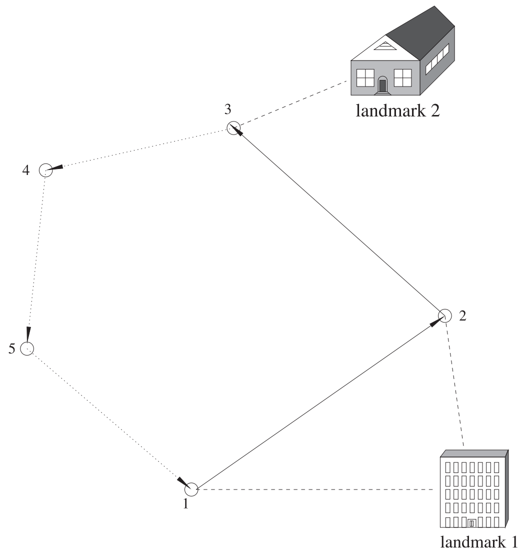

Figure 1. A Kinematic graph with cycles.

In the original paper [9] authors associate a matrix equation with each of the cycles: the product of homogeneous transformation matrices of each edge is equal to the identity matrix. Our approach is slightly different: we associate constraints with each of the edges as it will be described later. Such an approach seems much simpler to implement.

When a cycle in the kinematic graph is encountered, coordinates of the same position may be calculated in (at least) two ways – following two possible paths in the graph. For example on Figure 1 coordinates of the second robot position may be calculated otherwise from the odometer – with the use of distance from position one – or from the sonar – with the use of distance to the landmark (which position was already observed earlier).

Each edge of the kinematic graph corresponds to a relation between two positions, i.e. to some constraints on suitable variables. For example the edge in Figure 2 corresponds to the following constraints:

$$x_2 - x_1 = d \cdot \sin(\theta) \ ,$$
$$y_2 - y_1 = d \cdot \cos(\theta) \ .$$

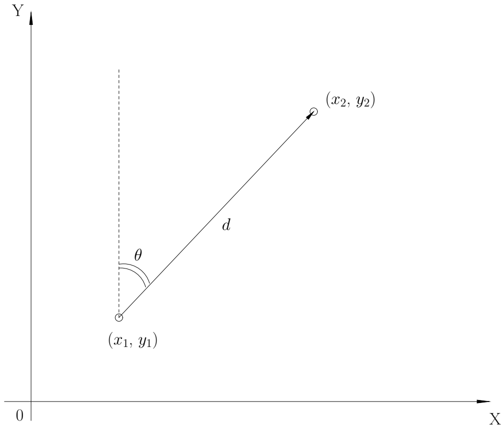

Figure 2. How do the positions relate.

If we treat the overall problem as a *Constraint Satisfaction Problem*, the above equations can be simply transformed into *narrowing operators* (see e.g. [1]) for variables x_1, x_2, y_1, y_2 and θ.

Before we return to the SLAM problem let us recall notions of the CSP and intervals.

3 Basics of interval computations

3.1 Interval arithmetic

Now, we shall define some basic notions of intervals and their arithmetic. We follow a wide literature here, like the books [5], [6], to name just some examples.

We define the (closed) interval $[\underline{x}, \overline{x}]$ as a set $\{x \in \mathbb{R} \mid \underline{x} \leq x \leq \overline{x}\}$.

Following [7], we use boldface lowercase letters to denote interval variables, e.g. $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ and $\mathbb{IR}$ denotes the set of all real intervals.

We design arithmetic operations on intervals so that the following condition was fulfilled: if we have $\odot \in \{+, -, \cdot, /\}$, $a \in \boldsymbol{a}$, $b \in \boldsymbol{b}$, then $a \odot b \in \boldsymbol{a} \odot \boldsymbol{b}$. We omit the actual formulae for arithmetic operations; they can be found elsewhere, e.g. ([5], [6]).

Now, let us define a notion to set links between real and interval functions.

Definition 3.1. A function $\mathsf{f}\colon \mathbb{IR} \to \mathbb{IR}$ is an *inclusion function* of $f\colon \mathbb{R} \to \mathbb{R}$, if for every interval $\boldsymbol{x}$ within the domain of f the following condition is satisfied:

$$\{f(x) \mid x \in \boldsymbol{x}\} \subseteq \mathsf{f}(\boldsymbol{x}) \ . \tag{1}$$

The definition is analogous for functions $f\colon \mathbb{R}^n \to \mathbb{R}^m$.

When computing interval operations, we can round the lower bound downward and the upper bound upward. This will result in an interval that will be a bit overestimated, but will be *guaranteed to contain the true result of the real–number operation.*

4 Constraint Satisfaction Problems

Literature (e.g. [1]) gives us some interesting definitions of CSPs and some underlying philosophy. Let us define a CSP as a set of equations and inequalities we want to solve with respect to a set of variables. Each of the variables has its *domain*, where we seek the solutions.

If the variables take only integer values (or more generally: discrete ones), we can find all solutions precisely. For real-valued variables, we can find solutions only approximately. Intervals are very appropriate to describe such solutions and the reliability of interval computations helps to make sure no solutions are lost.

There are a few approaches to solving a CPS. We use here the one based on *narrowing operators* ([2]).

Definition 4.1. Let ρ be an n-ary relation (constraint) over $\mathbb{R}$. The function $f\colon \mathbb{IR}^n \to \mathbb{IR}^n$ is a *narrowing operator* of ρ, iff for each $\boldsymbol{x} \in \mathbb{IR}^n$ the following conditions hold:

$$f(\boldsymbol{x}) \subseteq \boldsymbol{x} \text{ (contractance)}, \tag{2}$$

$$\boldsymbol{x} \cap \rho \subseteq f(\boldsymbol{x}) \text{ (correctness)}, \tag{3}$$

$$\boldsymbol{x} \subseteq \boldsymbol{y} \text{ implies } f(\boldsymbol{x}) \subseteq f(\boldsymbol{y}) \text{ (monotonicity)}, \tag{4}$$

$$f(f(\boldsymbol{x})) = f(\boldsymbol{x}) \text{ (idempotence)}. \tag{5}$$

In general, the set $\mathbb{IR}$ may be replaced by another family of subsets of $\mathbb{R}$.

The main algorithm to solve a CSP may be written as follows ([1]):

```
Chaotic_Iteration_of_Narrowing_Operators (D, F)
{ D is the domain and F is the set of narrowing operators }
begin
X := D; { D is the domain }
G := F; { F is the set of narrowing operators }
while (G ≠ ∅) do
begin
    choose g from G;
    X' := g(X);
    G := G ∪ {f ∈ F | f depends on a variable that g modifies };
    X := X';
end { of while };
end { of Chaotic_Iteration_of_Narrowing_Operators }
```

The above algorithm computes a fixed point of the set of narrowing operators. To compute the set of specific approximations of all solutions we should include a domain subdivision into it. Nevertheless, in our problem we assume a single solution and subdivision is not necessary.

5 Seeking the correspondence

As we have mentioned earlier, the word *correspondence* in SLAM community means the relation between a measurement made by the robot and a landmark associated with this measurement. Most papers, including [9] assume this relation to be unique, i.e. the robot can always correctly identify the landmark it can “see” with the sensors.

The main idea of relaxing this assumption comes form a pun: *correspondence* in other communities means a multifunction, i.e. a function $f\colon X \multimap Y$, associating subsets of Y to all elements of X. So, we assume that "the correspondence is a correspondence" and associate with each observation a set of possible landmarks it can refer to.

Discussion. A mainstream approach to solving the correspondence problem is the maximal likelihood method. It may be quite efficient if the landmarks are easy to distinguish but may lead to dramatic mapping errors in the case of misinterpreting an observation. Our approach seems to be more reliable, which is necessary when treating the SLAM problem as a Constraint Satisfaction Problem and using interval methods.

If correspondence is unique, it is simple to create the kinematic graph and to update it incrementally, while adding new robot positions (as the robot moves on) and observing new landmarks. When the correspondence is uncertain, so is the kinematic graph.

How to deal with this problem ? In our approach, we do not use a single kinematic graph, but a collection of kinematic graphs, according to several possibilities of identity of observed objects. For example when a new object is observed and the robot identifies it as either landmark 1 or 2 or some other, not observed earlier, we replace each of currently considered kinematic graphs by three new graphs, corresponding to these three possibilities.

Obviously, this leads to a large number of kinematic graphs that may grow quickly at each step. Nevertheless, many of the graphs can simply be discarded as CSPs associated with them are obviously contradictory.

For example, consider a situation when a robot observes an object at angle 90° on the right in distance of one meter (with the errors about a few centimeters). Then it goes straight 20 meters and sees on the right a similar object. Image processing may not allow to assure it is not the same landmark but the resulting CSP will be quickly proved to have no solution.

This example is very simple, but the preliminary experiments we have done suggest, the approach behaves well and several wrong kinematic graphs are quickly discarded.

6 Numerical experiments

The experiments were done only in simulation.

The example is taken from [9] and slightly simplified. The author of the cited article uses a Nomad Scout robot and assumes its trajectory is a circle.

Hence, we assume the robot moves straight and rotates without changing the position. This may be true for several platforms of mobile robots. The environment is similar to the one described in [9] (a square room of diameter 30m) and is presented in Figure 3. As in [9] we assume the odometer error is ± 25 cm and $\pm 5°$. Sonar errors are 10 times smaller.

Contrary to [9] we assume the robot does not distinguish the two corridors nor the two doors. It does notice the closest landmark and can distinguish if it is a door or a corridor. Also, it does not know a priori the number of landmarks it will encounter.

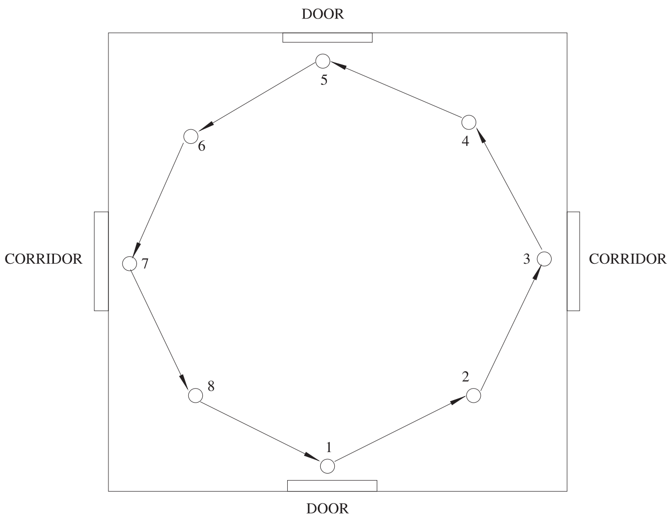

Figure 3. An example of the robot trajectory.

6.1 Results

Table 1 presents the results for the case with unique correspondence and Tables 2–6 – results for the case when it is uncertain. The first column in each table indicates description is described in this row – a certain pose of the robot (denoted by "r" with the pose number) or position of an observed landmark (denoted by "l" with number).

After the nine steps and closing the loop, the robot ended with 5 kinematic graphs. Two of them correctly prompted to 4 landmarks, three other computed 5 of them.

Moreover, the two 4-landmark-solutions differed only in labeling of the landmarks (and so were two of the three other ones). As a relatively simple algorithm could filter such equivalent solutions, there were effectively three of them.

Table 1. Results for the case with unique correspondence

description	$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta}$
r 1	[15.000000, 15.000000]	[5.000000, 5.000000]	[1.963495, 1.963496]
r 2	[21.649003, 22.625890]	[7.254080, 8.692127]	[1.090830, 1.265364]
r 3	[23.859385, 26.376647]	[13.576115, 16.498558]	[0.218166, 0.567233]
r 4	[18.991749, 25.398227]	[19.523065, 24.426118]	[-0.654499, -0.130899]
r 5	[11.003392, 19.871620]	[20.255855, 29.368930]	[-1.527164, -0.829031]
r 6	[3.007425, 14.807419]	[14.964192, 28.134633]	[-2.399828, -1.527163]
r 7	[-2.677651, 15.851102]	[6.968225, 20.138666]	[-3.272493, -2.225294]
r 8	[3.696705, 14.846075]	[1.353586, 12.142699]	[-4.145157, -2.923426]
r 9	[11.692672, 18.307328]	[3.758019, 5.050190]	[-5.017822, -3.621558]
l 1	[14.956149, 15.043851]	[-0.025001, 0.025190]	—
l 2	[28.765282, 31.127910]	[13.490822, 16.465664]	—
l 3	[10.010507, 20.577533]	[25.039668, 33.802960]	—
l 4	[-6.799463, 11.729289]	[4.374615, 22.449843]	—

Table 2. Results for the case with uncertain correspondence: solution 1

description	x	y	θ
r 1	[15.000000, 15.000000]	[5.000000, 5.000000]	[1.963495, 1.963496]
r 2	[21.649003, 22.625890]	[7.254080, 8.692127]	[1.090830, 1.265364]
r 3	[23.859385, 26.376647]	[13.576115, 16.498558]	[0.218166, 0.567233]
r 4	[18.403791, 25.943700]	[19.523065, 24.426118]	[-0.654499, -0.130899]
r 5	[10.415435, 20.417093]	[20.255855, 29.368930]	[-1.527164, -0.829031]
r 6	[2.419468, 15.352891]	[14.964192, 28.134633]	[-2.399828, -1.527163]
r 7	[-2.677651, 16.396574]	[6.968225, 20.138666]	[-3.272493, -2.225294]
r 8	[3.696705, 14.846075]	[1.353586, 12.142699]	[-4.145157, -2.923426]
r 9	[11.692672, 18.307328]	[3.758019, 5.050190]	[-5.017822, -3.621558]
l 1	[14.956149, 15.043851]	[-0.025001, 0.025190]	—
l 2	[28.765282, 31.127910]	[13.490822, 16.465664]	—
l 3	[28.022690, 31.828016]	[12.563177, 17.273132]	—
l 4	[9.422550, 21.123005]	[25.039668, 33.802960]	—
l 5	[-6.799463, 12.274762]	[4.374615, 22.449843]	—

Table 3. Results for the case with uncertain correspondence: solution 2

description	x	y	θ
r 1	[15.000000, 15.000000]	[5.000000, 5.000000]	[1.963495, 1.963496]
r 2	[21.649003, 22.625890]	[7.254080, 8.692127]	[1.090830, 1.265364]
r 3	[23.859385, 26.376647]	[13.576115, 16.498558]	[0.218166, 0.567233]
r 4	[18.991749, 25.398227]	[19.523065, 24.426118]	[-0.654499, -0.130899]
r 5	[11.003392, 19.871620]	[20.255855, 29.368930]	[-1.527164, -0.829031]
r 6	[3.007425, 14.807419]	[14.964192, 28.134633]	[-2.399828, -1.527163]
r 7	[-3.047025, 15.851102]	[6.968225, 20.138666]	[-3.272493, -2.225294]
r 8	[3.696705, 14.846075]	[1.353586, 12.142699]	[-4.145157, -2.923426]
r 9	[11.692672, 18.307328]	[3.758019, 5.050190]	[-5.017822, -3.621558]
l 1	[14.956149, 15.043851]	[-0.025001, 0.025190]	—
l 2	[28.765282, 31.127910]	[13.490822, 16.465664]	—
l 3	[-6.799463, 12.551566]	[2.523877, 22.449843]	—
l 4	[10.010507, 20.577533]	[25.039668, 33.802960]	—
l 5	[-7.952921, 11.729289]	[4.374615, 22.523398]	—

Please note that the two 4-landmark-solution for the case with uncertain correspondence were identical to the solution for unique correspondence.

Unfortunately, the positions also for the case without uncertainty are known only crudely – some of them have uncertainty of range over 17 meters. This is disconcerting, but the reason is that the narrowing operators we used were very simple. In particular, no narrowing of θ's were used. In [9] an advanced kinematic solver was used and the error obtained was below one meter.

What counts for our experiments is that adding the uncertainty of correspondence has not spoiled the results significantly.

Table 4. Results for the case with uncertain correspondence: solution 3

description	$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta}$
r 1	[15.000000, 15.000000]	[5.000000, 5.000000]	[1.963495, 1.963496]
r 2	[21.649003, 22.625890]	[7.254080, 8.692127]	[1.090830, 1.265364]
r 3	[23.859385, 26.376647]	[13.576115, 16.498558]	[0.218166, 0.567233]
r 4	[18.991749, 25.398227]	[19.523065, 24.426118]	[-0.654499, -0.130899]
r 5	[11.003392, 19.871620]	[20.255855, 29.368930]	[-1.527164, -0.829031]
r 6	[3.007425, 14.807419]	[14.964192, 28.134633]	[-2.399828, -1.527163]
r 7	[-2.677651, 15.851102]	[6.968225, 20.138666]	[-3.272493, -2.225294]
r 8	[3.696705, 14.846075]	[1.353586, 12.142699]	[-4.145157, -2.923426]
r 9	[11.692672, 18.307328]	[3.758019, 5.050190]	[-5.017822, -3.621558]
l 1	[14.956149, 15.043851]	[-0.025001, 0.025190]	—
l 2	[28.765282, 31.127910]	[13.490822, 16.465664]	—
l 3	—	—	—
l 4	[10.010507, 20.577533]	[25.039668, 33.802960]	—
l 5	[-6.799463, 11.729289]	[4.374615, 22.449843]	—

Table 5. Results for the case with uncertain correspondence: solution 4

description	$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta}$
r 1	[15.000000, 15.000000]	[5.000000, 5.000000]	[1.963495, 1.963496]
r 2	[21.649003, 22.625890]	[7.254080, 8.692127]	[1.090830, 1.265364]
r 3	[23.859385, 26.376647]	[13.576115, 16.498558]	[0.218166, 0.567233]
r 4	[18.991749, 25.398227]	[19.523065, 24.426118]	[-0.654499, -0.130899]
r 5	[11.003392, 19.871620]	[20.255855, 29.368930]	[-1.527164, -0.829031]
r 6	[3.007425, 14.807419]	[14.964192, 28.134633]	[-2.399828, -1.527163]
r 7	[-2.677651, 15.851102]	[6.968225, 20.138666]	[-3.272493, -2.225294]
r 8	[3.696705, 14.846075]	[1.353586, 12.142699]	[-4.145157, -2.923426]
r 9	[11.692672, 18.307328]	[3.758019, 5.050190]	[-5.017822, -3.621558]
l 1	[14.956149, 15.043851]	[-0.025001, 0.025190]	—
l 2	[28.765282, 31.127910]	[13.490822, 16.465664]	—
l 3	[-6.799463, 11.729289]	[4.374615, 22.449843]	—
l 4	[10.010507, 20.577533]	[25.039668, 33.802960]	—

7 Conclusions

An new way of solving the SLAM problem with unknown correspondence has been proposed and preliminary results show it can be promising.

As it was written above, the precision of computing robot's and landmarks' localizations was much worse than in [9]. It was probably caused by a far less advanced solver used to solve inverse kinematic problems. It is also unclear whether in [9] landmarks are assumed to have only Cartesian coordinates x and y or an orientation θ, too. In out experiments it was assumed they do not have an orientation (or we cannot measure it), which – obviously – makes the problem more difficult.

More experiments – especially with a more sophisticated kinematic solver and with a real robot – will be performed to verify the usefulness of our approach in practice.

Table 6. Results for the case with uncertain correspondence: solution 5

description	$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta}$
r 1	[15.000000, 15.000000]	[5.000000, 5.000000]	[1.963495, 1.963496]
r 2	[21.649003, 22.625890]	[7.254080, 8.692127]	[1.090830, 1.265364]
r 3	[23.859385, 26.376647]	[13.576115, 16.498558]	[0.218166, 0.567233]
r 4	[18.991749, 25.398227]	[19.523065, 24.426118]	[-0.654499, -0.130899]
r 5	[11.003392, 19.871620]	[20.255855, 29.368930]	[-1.527164, -0.829031]
r 6	[3.007425, 14.807419]	[14.964192, 28.134633]	[-2.399828, -1.527163]
r 7	[-3.047025, 15.851102]	[6.968225, 20.138666]	[-3.272493, -2.225294]
r 8	[3.696705, 14.846075]	[1.353586, 12.142699]	[-4.145157, -2.923426]
r 9	[11.692672, 18.307328]	[3.758019, 5.050190]	[-5.017822, -3.621558]
l 1	[14.956149, 15.043851]	[-0.025001, 0.025190]	—
l 2	[28.765282, 31.127910]	[13.490822, 16.465664]	—
l 3	[-7.952921, 11.729289]	[4.374615, 22.523398]	—
l 4	[10.010507, 20.577533]	[25.039668, 33.802960]	—
l 5	[-6.799463, 12.551566]	[2.523877, 22.449843]	—

Bibliography

[1] K. R. Apt, "*The essence of constraint propagation*", *Theoretical Computer Science*, Vol. 221, No. 1–2 (1999), pp. 179–210.

[2] F. Benhamou, L. Granvilliers, "*Combining Local Consistency, Symbolic Rewriting and Interval Methods*", *LNCS*, Vol. 1138 (1996), pp. 144–159.

[3] T. Duckett, "*A genetic algorithm for simultaneous localization and mapping*", Proceedings of ICRA'03, Vol. 1 (2003), pp. 434–439.

[4] U. Frese, "*A Discussion of Simultaneous Localization and Mapping*", *Autonomous Robots*, Vol. 20, No. 1 (2006), pp. 25–42.

[5] E. Hansen, "*Global Optimization Using Interval Analysis*", Marcel Dekker, New York, 1992.

[6] R. B. Kearfott, "*Rigorous Global Search: Continuous Problems*", Kluwer Academic Publishers, Dordrecht, 1996.

[7] R. B. Kearfott, M. T. Nakao, A. Neumaier, S. M. Rump, S. P. Shary, P. van Hentenryck, "*Standardized notation in interval analysis*", available on the web at `http://www.mat.univie.ac.at/~neum/software/int/notation.ps.gz` .

[8] M. Montemarlo, S. Thrun, D. Koller, B. Wegbreit, "*FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem*", Proceedings of the Eighteenth National Conference on Artificial Intelligence, pp. 593–598.

[9] J. M. Porta, "*CuikSLAM: A Kinematics-based Approach to SLAM*", *IEEE International Conference on Robotics and Automation. IEEE Press* (2005), pp. 2436–2442.