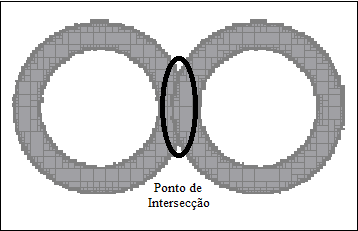
**Step 1**: Study how the SIVIA algorithm works and how insert new equations.



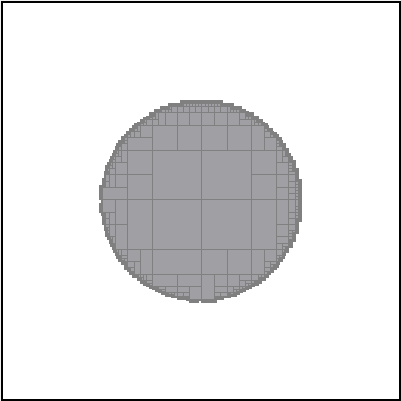


Figure 2 - test 2

Figure 1 - test 1

The SIVIA algorithm and the Qt framework were studied. Examples of the tests can be seen in the Figures 1 and 2.

**Step 2**: Use the SIVIA algorithm to localize the robot with the sensors information.

To use the SIVIA algorithm for localization were necessary simulated data for the tests. An example can be seen in the Figure 3, the parameters used are in the Tables 1 and 2.

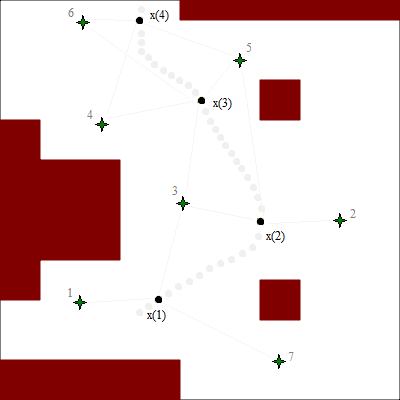


Figure 3 - map

In the Figure 3 has the representation the landmarks like green stars, obstacles in dark red color and robot positions are black circles.

Table 1 - Parameters

|  |  |
| --- | --- |
| Parameters | |
| Size | 20m x 20m |
| Landmarks | 7 |
| Range landmarks | 8m |
| Error | 0.3m |
| Range laser | 10m |

Table 2 - Landmarks localization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Landmarks localization (x,y) | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| (4,5) | (17,9) | (9,10) | (5,14) | (12,17) | (4,19) | (14,2) |

Table 3 - Robot positions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Robot positions | | | | |
|  | x(1) | x(2) | x(3) | x(4) |
| (x, y, th) | (8, 5, 25) | (13, 9, 120) | (10, 15, 135) | (7, 19, 90) |

The Figures 4 and 5 show the results with the algorithm using the data described above. In each position x(i) was found one box set, in grey, who represents the possibles robot position. The black point represents the real robot position. The error used to computed the result showed in the Figure 4 was 0.3m, in the Figure 5 (a) was 0.5m and (b) was 1m. Thereby is possible to see the increased uncertainties about the robot position given by reduction of accuracy.

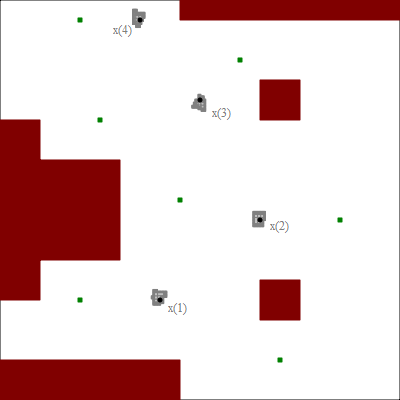


Figure 4 – Results with error 0.3m

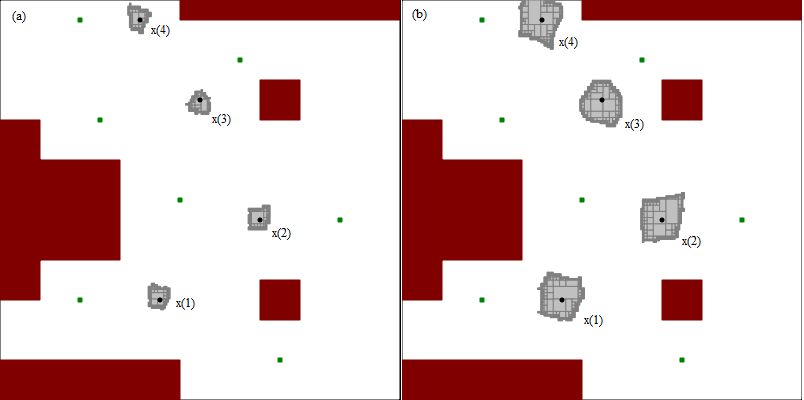


Figure 5 – Results with error 0.5m in (a) and 1m in (b)

**Step 3**: To insert particles in the boxes and to calculate your probabilities, highlighting best particle.

In the step 3, particles were distributed in all the boxes created, in order to estimate the robot position. In Figures 6, 7 and 8 are shown the algorithm results with the particles. Now, besides previous information, the figures have the representation of the particles in red, the best particle in blue and the real robot position in yellow.

Table 4 - Position of best particle

|  |  |  |  |
| --- | --- | --- | --- |
| Results referring to Figure 6 | | | |
| x(1) | **x** | **y** | **th** |
| Best particle | 7.94 | 5.15 | 24 |
| Real | 8 | 5 | 25 |
| x(2) | **x** | **y** | **th** |
| Best particle | 12.93 | 9.22 | 124 |
| Real | 13 | 9 | 120 |
| x(3) | **x** | **y** | **th** |
| Best particle | 9.95 | 14.81 | 133 |
| Real | 10 | 15 | 135 |
| x(4) | **x** | **y** | **th** |
| Best particle | 7 | 19.04 | 89 |
| Real | 7 | 19 | 90 |

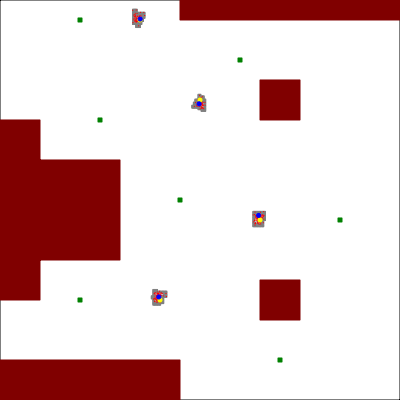


Figure 6 - Using particles

Table 5 - Positions of the best particle

|  |  |  |  |
| --- | --- | --- | --- |
| Results referring to Figure 7 | | | |
| x(1) | **x** | **y** | **th** |
| Best particle | 8.4 | 5 | 25 |
| Real | 8 | 5 | 25 |
| x(2) | **x** | **y** | **th** |
| Best particle | 13.1 | 8.89 | 121 |
| Real | 13 | 9 | 120 |
| x(3) | **x** | **y** | **th** |
| Best particle | 9.95 | 15.03 | 135 |
| Real | 10 | 15 | 135 |
| x(4) | **x** | **y** | **th** |
| Best particle | 6.97 | 18.77 | 90 |
| Real | 7 | 19 | 90 |

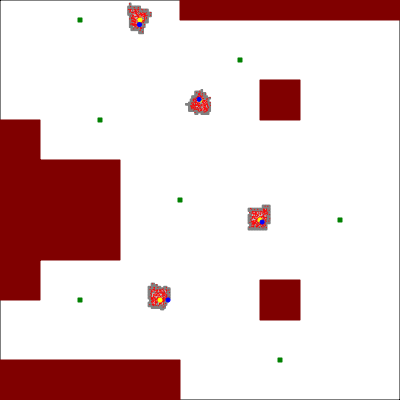


Figure 7 - Using particles

|  |  |  |  |
| --- | --- | --- | --- |
| Results referring to Figure 8 | | | |
| x(1) | **x** | **y** | **th** |
| Best particle | 7.85 | 5.22 | 14 |
| Real | 8 | 5 | 25 |
| x(2) | **x** | **y** | **th** |
| Best particle | 13.11 | 9.02 | 118 |
| Real | 13 | 9 | 120 |
| x(3) | **x** | **y** | **th** |
| Best particle | 10.11 | 15.2 | 136 |
| Real | 10 | 15 | 135 |
| x(4) | **x** | **y** | **th** |
| Best particle | 7.28 | 19.1 | 95 |
| Real | 7 | 19 | 90 |

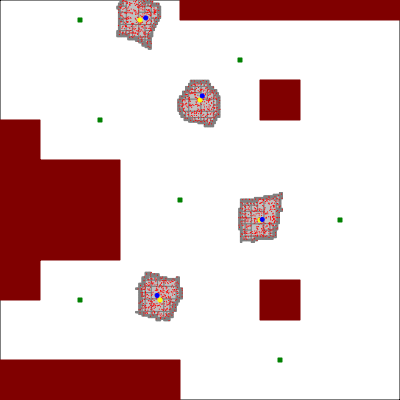


Figure 8 - Using particles