Politecnico di Torino

Modeling and control of cyberphysical systems

Project I: distributed localization with CPSs

Simone Gallo
s276217

Francesco Menon
s277870

Angelo Pettinelli
s269291

Esmeraldi Xuna
s277995

June 15, 2021

Contents

1 Introduction

In this project, we simulate an indoor localization/tracking system through a wireless sensor network (WSN). The sensor acquire the received signal strength (RSS) on a signal broadcast by a target to be located.

1.1 Aims of the project

1. Simulation of a localization/tracking problem in WSNs
2. Implementation of a localization/tracking distributed algorithm
3. Analysis of the results

1.2 Physical setting

- Environment: square room of $100\,\mathrm{m}^2$
- Grid: $p = 100$ square cells of $1\,\mathrm{m}^2$
- Reference points: centers of the cells
- RSS model: indoor empirical model defined by the IEEE 802.15.4 standard

$$RSS(d) = \begin{cases} P_t - 40.2 - 20\log d + \eta & \text{, if } d \leq 8\,\mathrm{m} \\ P_t - 58.5 - 33\log d + \eta & \text{, if } d > 8\,\mathrm{m} \end{cases} \tag{1}$$

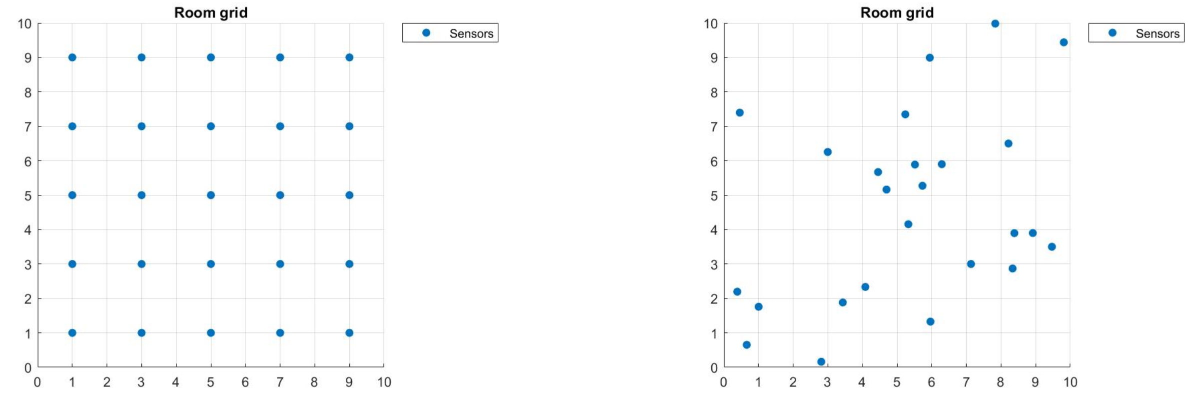
  where $P_t = 25$, η is a Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2)$, $\sigma = 0.5$.

1.3 WSN

- $n = 25$ sensors
- Deployment:
  - *uniformly* at random positions; each sensor is connected with sensors at distance $\leq r$.
  - *grid* topology: the sensors are deployed on a grid 5×5; sensors are connected to 4 closest sensors (3 or 2 on the boundaries)

1.4 Data preprocessing

To generate the data for the execution of the project, the script build_data.m must be executed first. It generates the file data_cps.m which stores all the data used by the other scripts. The user can choose whether to generate a random topology for the sensor network or a grid mesh. For completeness, we analyzed both the situations.

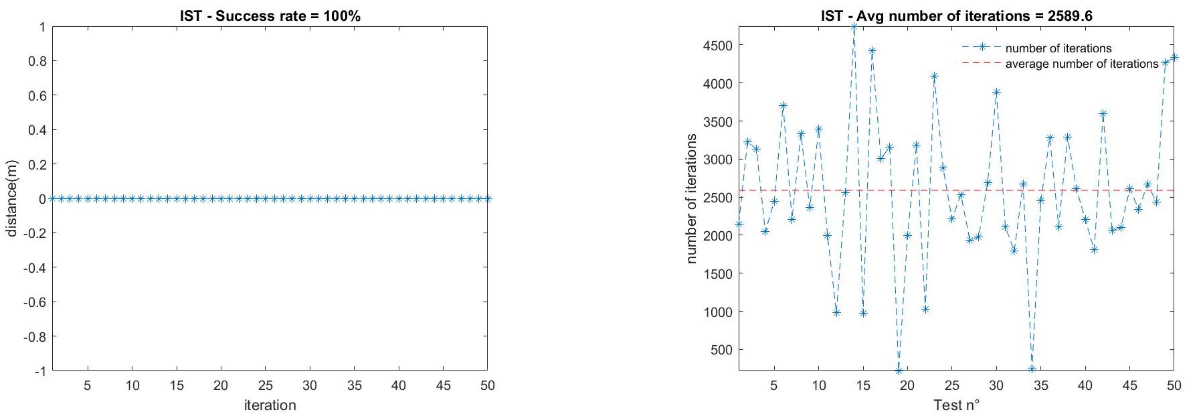


(a) Grid topology (b) Uniform topology

Figure 1.1: Room topology

2 Localization

2.1 IST

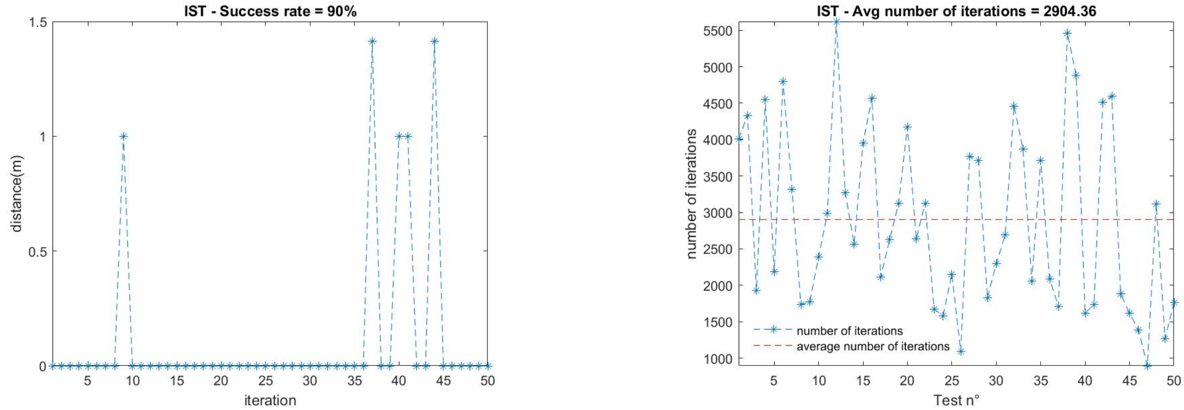


(a) Success rate of 100% (b) Average number of iterations

Figure 2.1: IST algorithm performed on grid topology ($\lambda = 10^{-5}$, $\tau = 0.7$, $\epsilon = 10^{-5}$)

We obtained a better accuracy with the grid topology because the room is uniformly covered by the sensors. By contrast, the uniform topology leaves some uncovered spaces, making the localization inaccurate: on average, the estimated target is a cell adjacent to the correct one.

The average number of iterations is quite the same. In the uniform topology, the presence of errors increases the number of interations to converge.



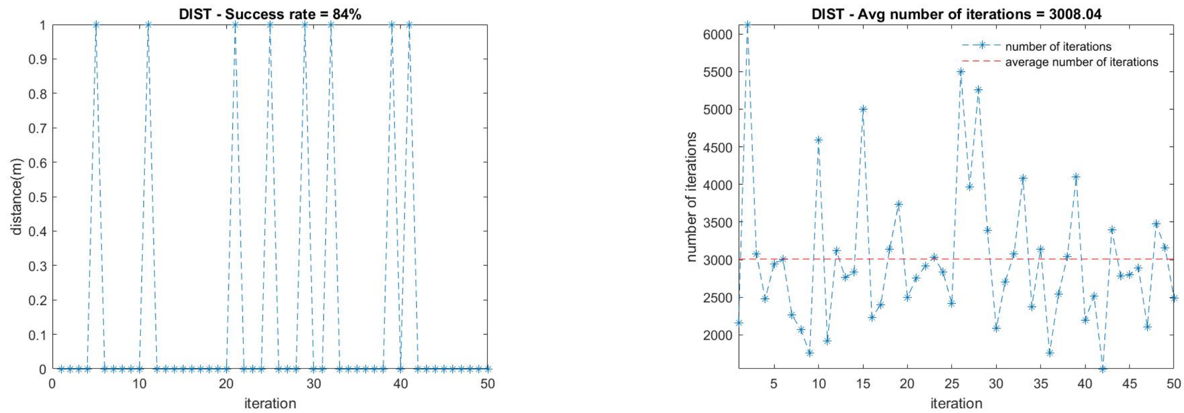
(a) Success rate of 90%

(b) Average number of iterations

Figure 2.2: IST algorithm performed on uniform topology ($\lambda = 10^{-5}$, $\tau = 0.7$, $\epsilon = 10^{-5}$)

2.2 DIST

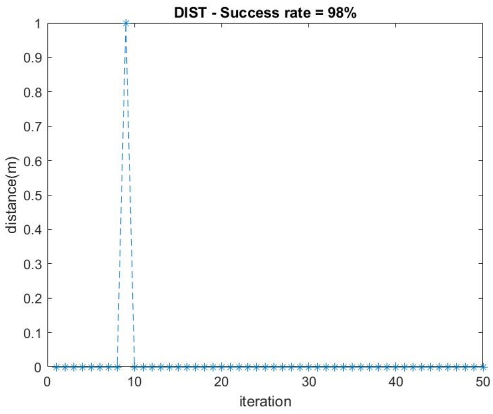
With $\epsilon = 10^{-5}$, the DIST algorithm performed on the grid topology stops , so we reduced it to 10^{-6}, improving the accuracy but, on the other hand, increasing the average number of iterations. Moreover, if we look at the results of the DIST algorithm with $\epsilon = 10^{-5}$, we can notice that sensors have faster convergence time, but since we have to wait for the last sensor to converge, globally the number of iterations is bigger with respect to the IST algorithm.



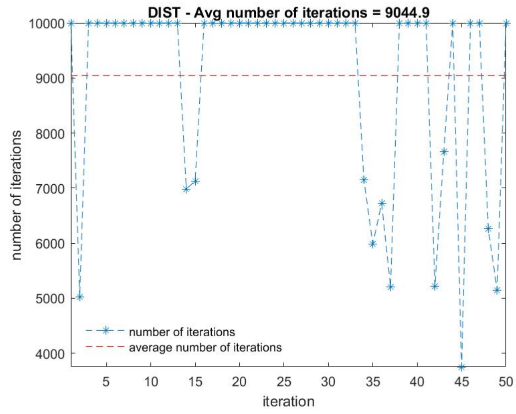
(a) Success rate of 84%

(b) Average number of iterations

Figure 2.3: DIST algorithm performed on grid topology ($\lambda = 10^{-5}$, $\tau = 0.7$, $\epsilon = 10^{-5}$)

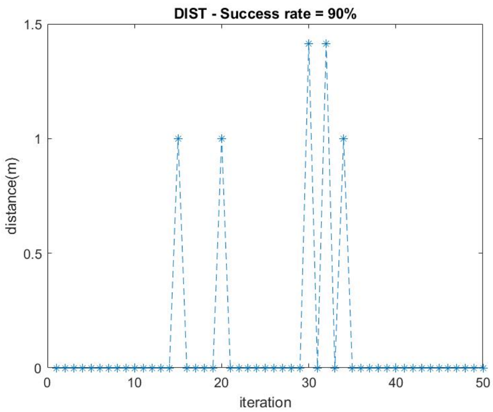


(a) Success rate of 98%

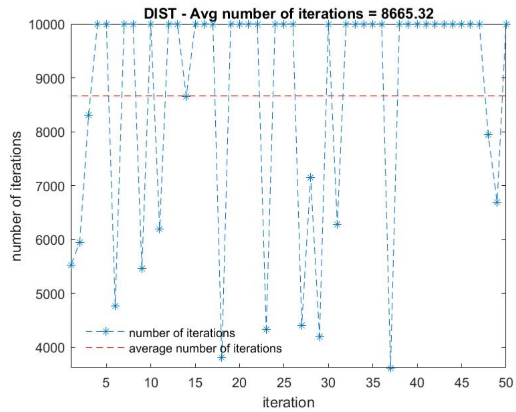


(b) Average number of iterations

Figure 2.4: DIST algorithm performed on grid topology ($\lambda = 10^{-5}$, $\tau = 0.7$, $\epsilon = 10^{-6}$)



(a) Success rate of 90%



(b) Average number of iterations

Figure 2.5: DIST algorithm performed on uniform topology ($\lambda = 10^{-5}$, $\tau = 0.7$, $\epsilon = 10^{-6}$)

As for the IST algorithm, we obtained a better accuracy with the grid topology but similar number of iterations.

After running multiple times the algorithm on different uniform topologies, we analyzed the convergence time with respect to the spectral radius of the matrix Q. As we can see from Figure 2.6, the number of iterations to achieve convergence grows with the spectral radius of the matrix Q.

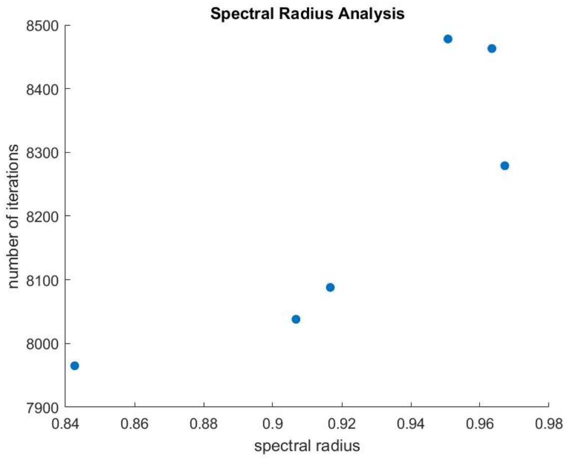
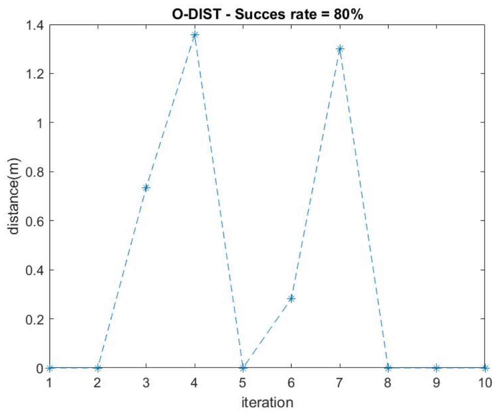


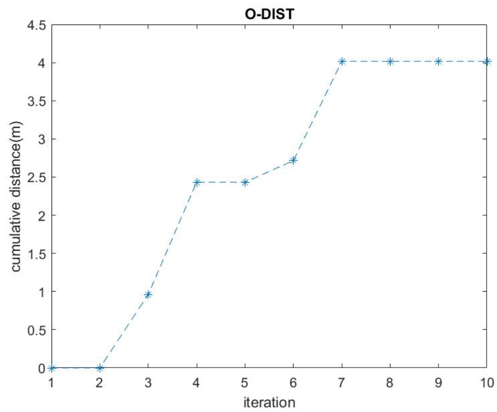
Figure 2.6: Spectral Radius Analysis on uniform topology ($\lambda = 10^{-5}$, $\tau = 0.7$, $\epsilon = 10^{-6}$)

3 Tracking

The tracking task has been performed exclusively on the grid topology because, in general, it produces better results. We have analyzed two different cases: in the first one (Figure 3.1) the target is moving along the diagonal of the room, while in the second one (Figures 3.2 and 3.3), the target is changing it's position randomly at every single step.

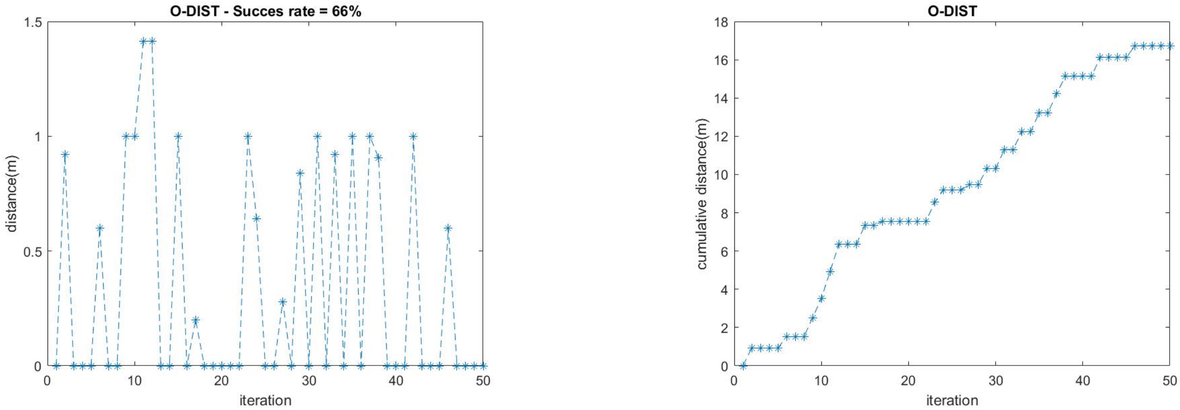


(a) Success rate of 80%



(b) Cumulative distance

Figure 3.1: O-DIST algorithm performed on uniform topology with the target moving along the diagonal of the room ($\epsilon = 10^{-6}$, $T = 10^2$)

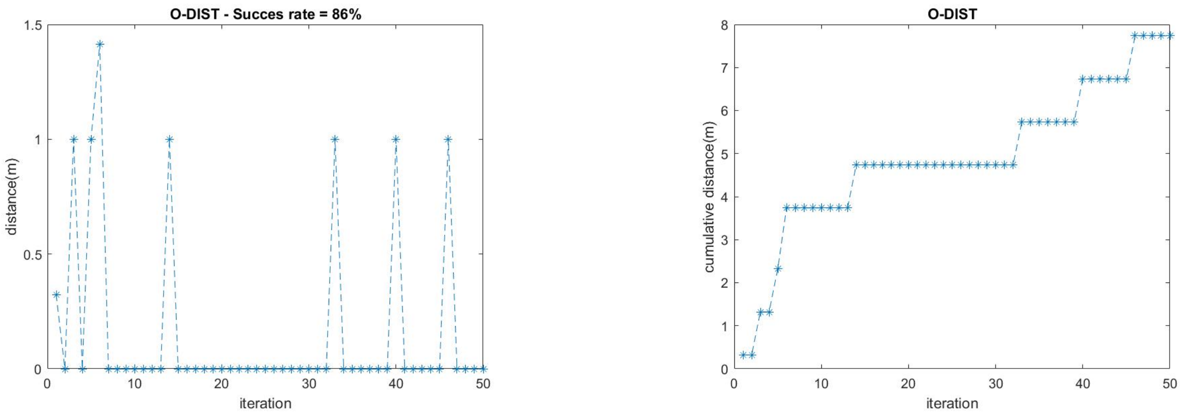


(a) Success rate of 66% (b) Cumulative distance

Figure 3.2: O-DIST algorithm performed on uniform topology with the target moving randomly in the room ($\epsilon = 10^{-6}$, $T = 5 \cdot 10^2$)

With 10^2 maximum number of iterations, the algorithm is not precise, in fact the convergence is not achieved. However, the wrong estimated cell is always the adjacent to the correct one.

To improve the accuracy, we can increase the maximum number of iterations from 10^2 to $5 \cdot 10^2$. As we can see in Figure 3.3, we obtained better results in terms of success rate (from 66% to 86%) and cumulative distance.



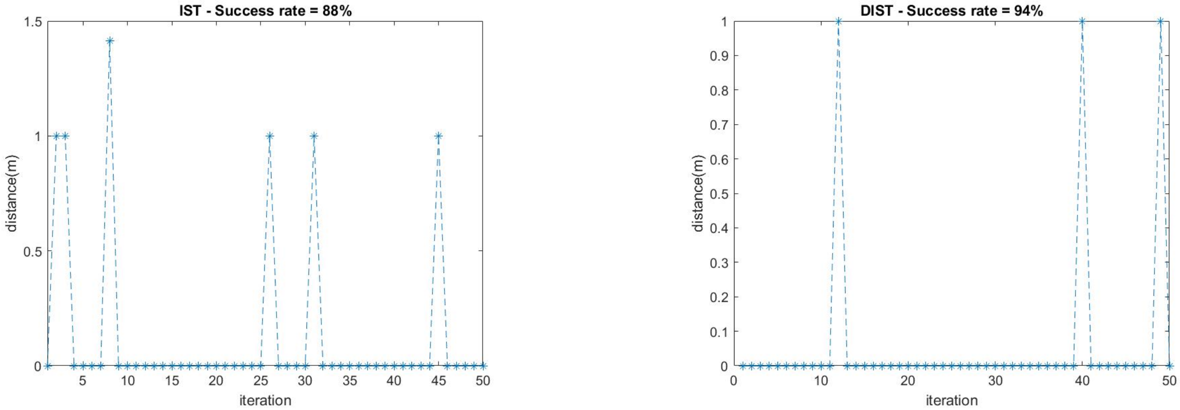
(a) Success rate of 66% (b) Cumulative distance

Figure 3.3: O-DIST algorithm performed on uniform topology with the target moving randomly in the room ($\epsilon = 10^{-6}$, $T = 5 \cdot 10^2$)

4 Extensions

4.1 Changing parameters

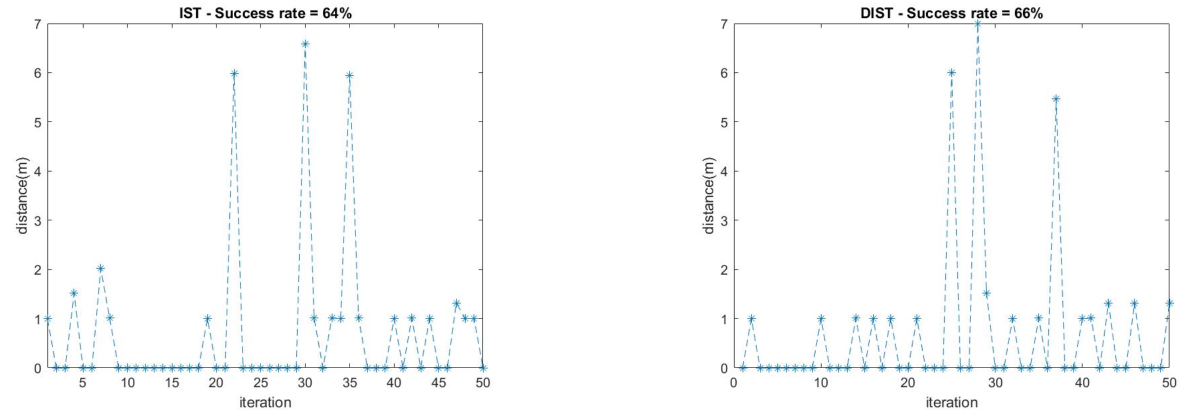
To better investigate the robustness of our algorithms we tried to modify the noise increasing the standard deviation by a factor of 2. We noticed that, increasing the noise, the IST algoritm is less precise with respect to the DIST one, which is more robust preserving approximately the same success rate.



(a) IST (b) DIST

Figure 4.1: Comparison of IST and DIST performance increasing noise

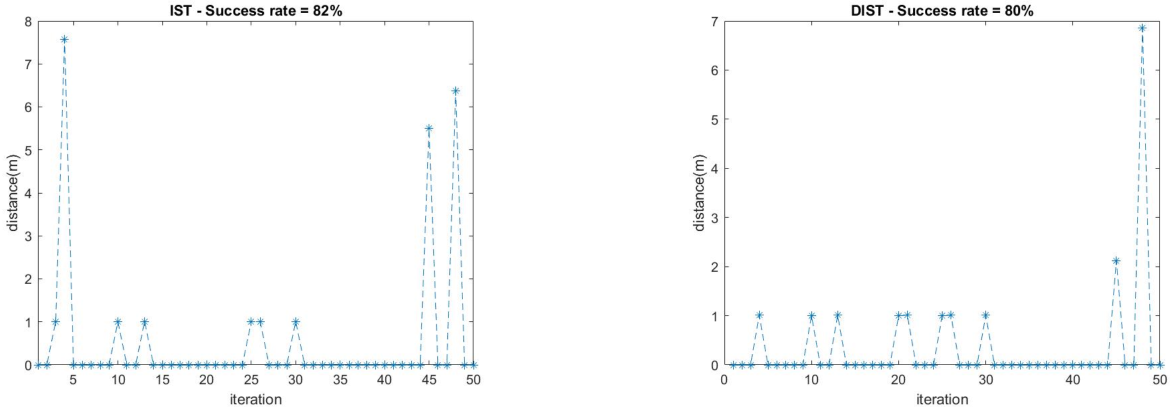
Then, we increased the number of cells from 100 to 200 and both the IST and DIST alghoritm performed with a very low success rate, due to the uniform topology, because the regions uncovered by sensors increase with the number of cells.



(a) IST algorithm on a grid of 200 cells (b) DIST algorithm on a grid of 200 cells

Figure 4.2: Comparison of IST and DIST performance increasing the number of cells

We increased the number of sensors to 40 to obtain a good rate of success. After running the algorithm different times on a uniform topology, we empirically noticed that, to obtain a success rate of at least 80%, the number of sensors should be around $\frac{1}{4}$ or $\frac{1}{5}$ of the number of cells.



(a) IST algorithm on a grid of 200 cells and 40 sensors

(b) DIST algorithm on a grid of 200 cells and 40 sensors

Figure 4.3: Comparison of IST and DIST performance increasing the number of cells

4.2 k-Nearest Neighbors

To implement the k-Nearest Neighbors algorithm we used the function knnsearch from the Machine Learning Toolbox.

We noticed that, on a grid topology it rarely misses. Using a uniform topology, it's success rate is lower, but it is still very accurate (96%). However, it is computationally intensive and the computational cost increases with the number of cells and the number of sensors.

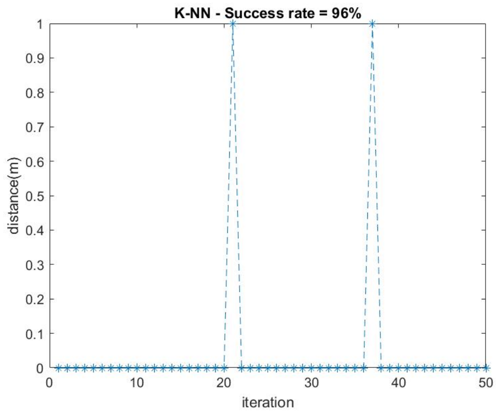


Figure 4.4: k-Nearest Neighbors success rate using a uniform topology

4.4 Presence of broken sensors

We added two broken sensors to our environment to produce a wrong *RSS*. We approached to this situation as an external attack to the network.

We defined

$$B_{aug} = \begin{bmatrix} A & I_n \end{bmatrix} \qquad \text{with} \quad x \in R^p,\, u \in R^n$$
$$z_{aug} = \begin{bmatrix} x & u \end{bmatrix}$$

Even if two sensors are borken, the succes rate is quite good. We also estimated the succes rate for the identification of the two broken sensors, which resulted to be very good (92%).

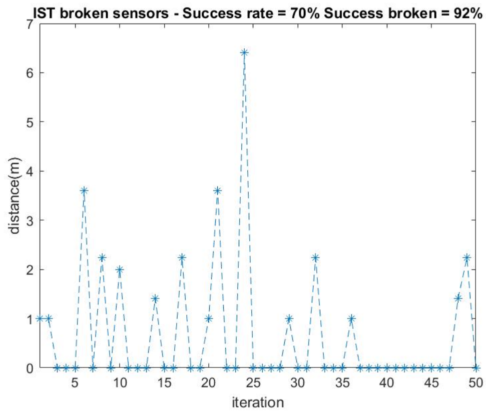


Figure 4.5: k-Nearest Neighbors success rate