5.
Tests and results

This chapter presents results obtained using the proposed method on simulated and real data. First, it is analyzed the scan matching optimization; after that, the Scan Matching Motion Model is detailed presented, then some 2D results are shown using DP-SLAM; and finally a simulated and a real 3D map example are presented.

5.1. Scan Matching

5.1.1. Optimization Parameters Influence

Because this analysis uses simulated data, the truth robot displacement values, Δx, Δy, $\Delta\theta$; are known exactly; this will help us to test the optimization parameters influence on the scan matching process.

As shown in Section 4.4.2, the optimization using genetic algorithm depends on the size of the population and the number of generations. Figure 5.1 and Figure 5.2 present the error displacement obtained by scan matching, showing the population size influence in DE optimization (the number of generation is kept fixed, in this case *generations* = 50), this experiment has 366 simulated robot poses and were acquired by simulating an standard deviation error of 15 mm.

Notice that in Figure 5.1, small population leads to non-convergence and consequently the estimated displacement surfers from hug errors, as shown in this case for population size of 20 and 40.

In the other hand Figure 5.2, shows the same experiment using a larger population; here the peaks have been significantly reduced in size and number (see the scale).

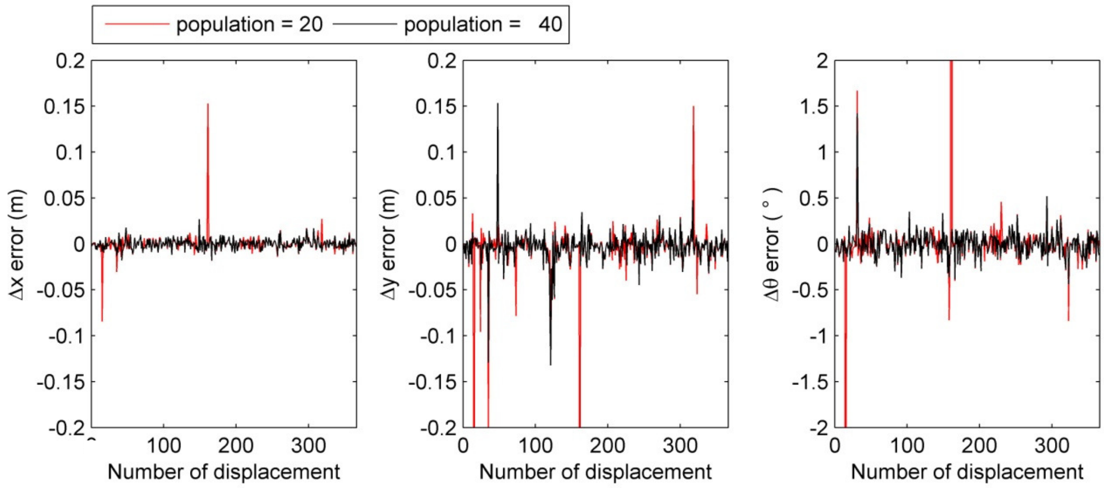

Figure 5.1: Error in displacement, Δx, Δy, $\Delta\theta$, influenced by population size (20 and 40) in DE optimization.

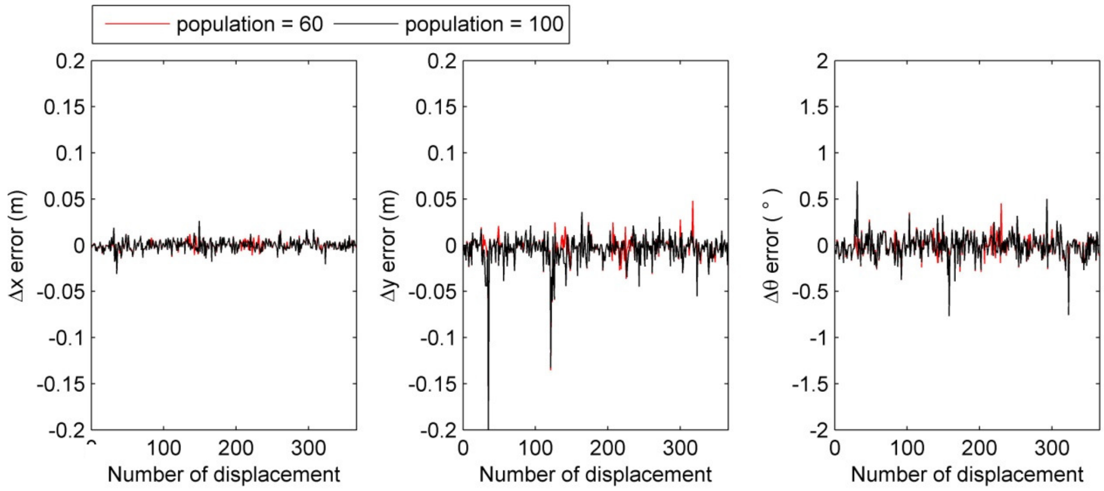

Figure 5.2: Error in displacement, Δx, Δy, $\Delta\theta$, influenced by population size (60 and 100) in DE optimization.

Figure 5.3 and Figure 5.4 show the error displacement, when the population size is fixed to 100 and the generation number is varied. The effect of generation increase is to improve the precision in displacement estimation. As seen in Figure 5.3 (for 15 and 30 generations), although convergence has been guaranteed, there are not enough generations to significantly improve the estimation.

On the other hand, Figure 5.4, shows a refined estimation by increasing the generation number (50 and 75 generations).

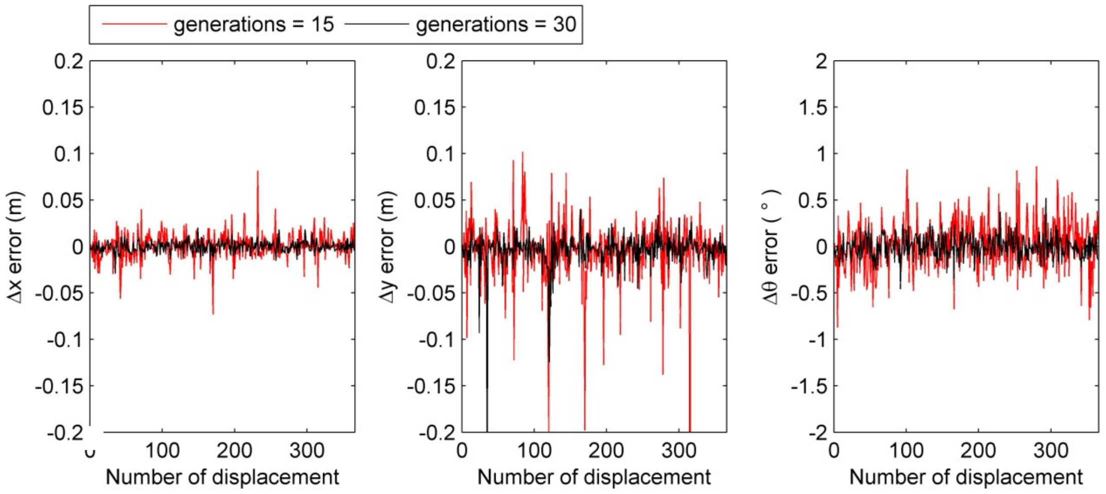

Figure 5.3: Error in displacement, Δx, Δy, $\Delta\theta$, influenced by number of generations (15 and 30) in DE optimization.

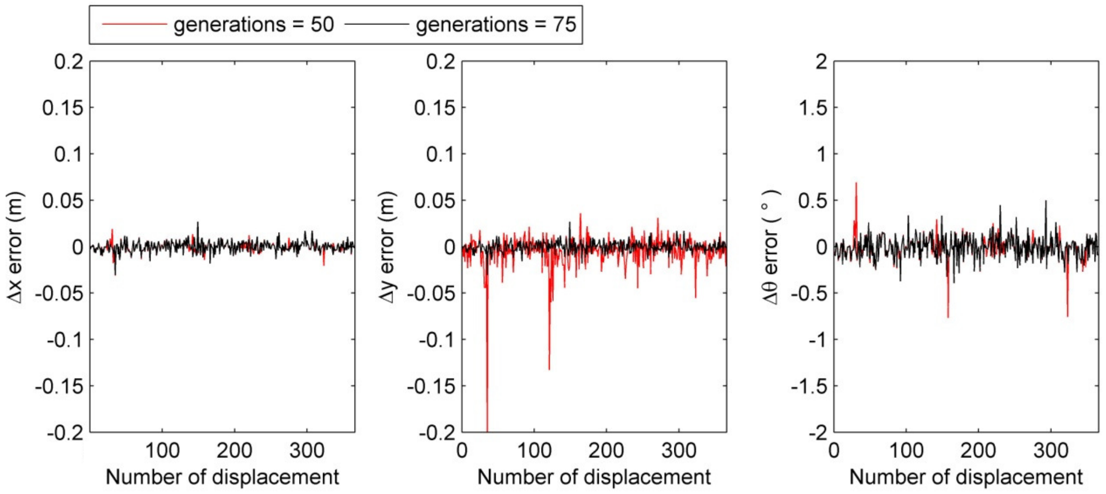

Figure 5.4: Error in displacement, Δx, Δy, $\Delta\theta$, influenced by number of generations (50 and 75) in DE optimization.

5.1.2. LRF Error Influence

Another important scan matching parameter is the LRF error. Using the same simulated environment and simulated robot poses, error in laser range is introduced by adding a random value picked from a Gaussian distribution with zero mean and a variable standard deviation (stdv). In this case, the standard deviations are between 15mm and 80mm.

Figure 5.5 and Figure 5.6 show the LRF error influence in DE optimization. This simulation uses a fixed population size, 100, and a fixed number of generations, 50, showing that the greater the laser range error the worse the estimated displacement. Note however that DE optimization is robust, even if the error displacements increase.

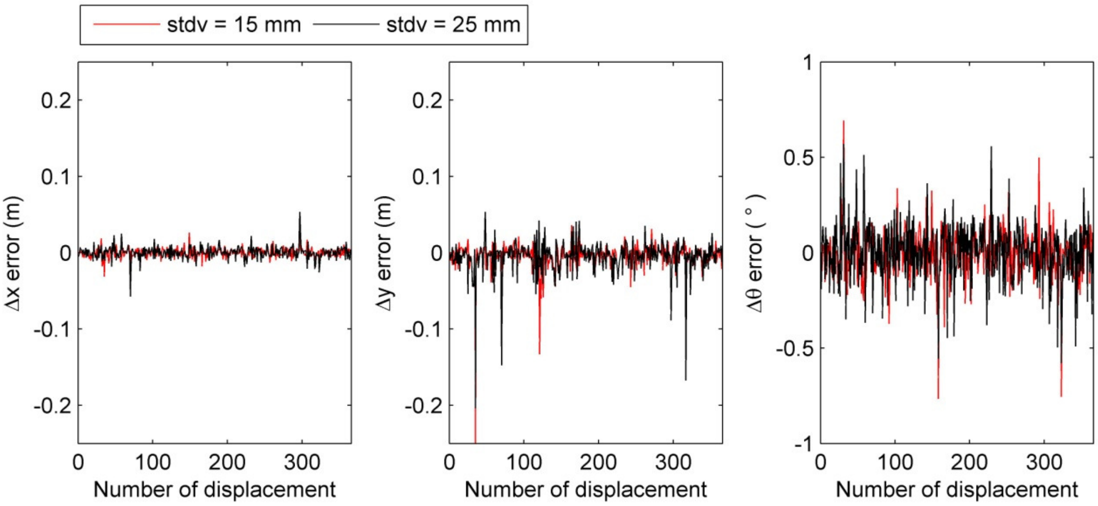

Figure 5.5: Displacement error, Δx, Δy, $\Delta\theta$, influenced by LRF error (stdv= 15mm and stdv=25mm) in DE optimization.

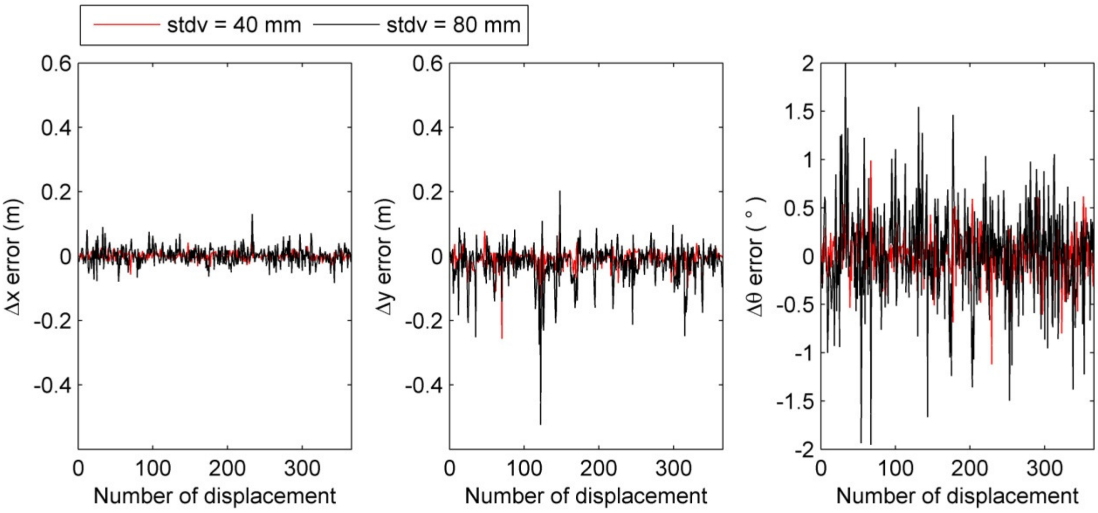

Figure 5.6: Displacement error, Δx, Δy, $\Delta\theta$, influenced by LRF error (stdv= 40mm and stdv=80mm) in DE optimization.

As explained in the introduction of this work, although technology offers increasingly accurate position sensors, even small measurement errors can accumulate and compromise the localization accuracy. This becomes evident when programming a robot to return to its original position after traveling a long distance, based only on its sensor readings. In this case, our position “sensor” is the scan matching process.

Figure 5.7 shows the accumulated error in robot position, where (robot poses are marked in capital letters according to Figure 5.8). This error represents the Euclidean distance between the estimated and true robot simulated robot position. Note that the error inevitably grows as the robot pose number increases.

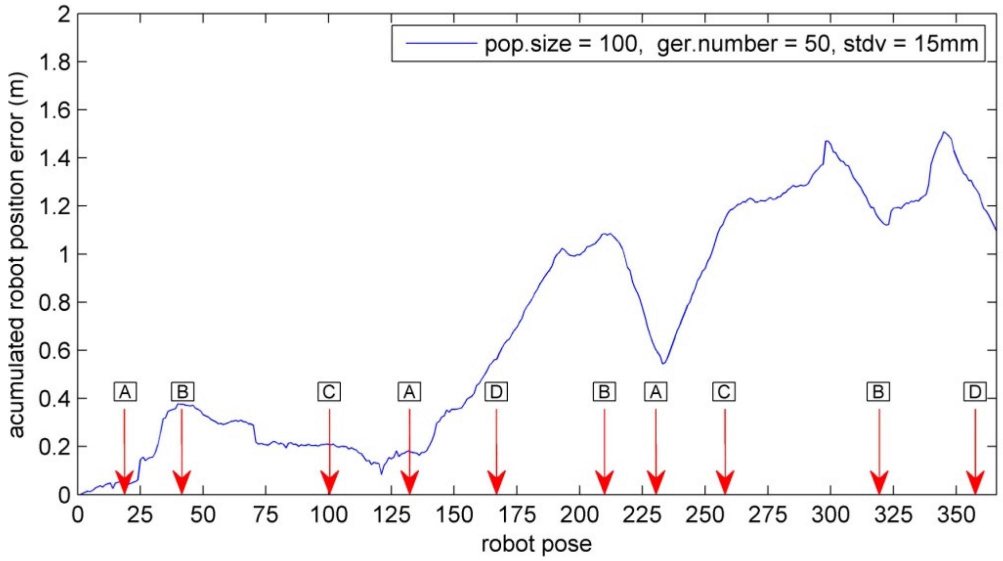

Figure 5.7: Accumulated error in robot position due to imperfect scan matching.

Figure 5.8 shows the simulated environment used for all theses scan matching experiments. It also shows the true (red line) and estimated (blue line) robot path acquired by scan matching. Figure 5.9 shows the map for the estimated robot path, showing the growing misalignments due to accumulated small errors in scan matching process.

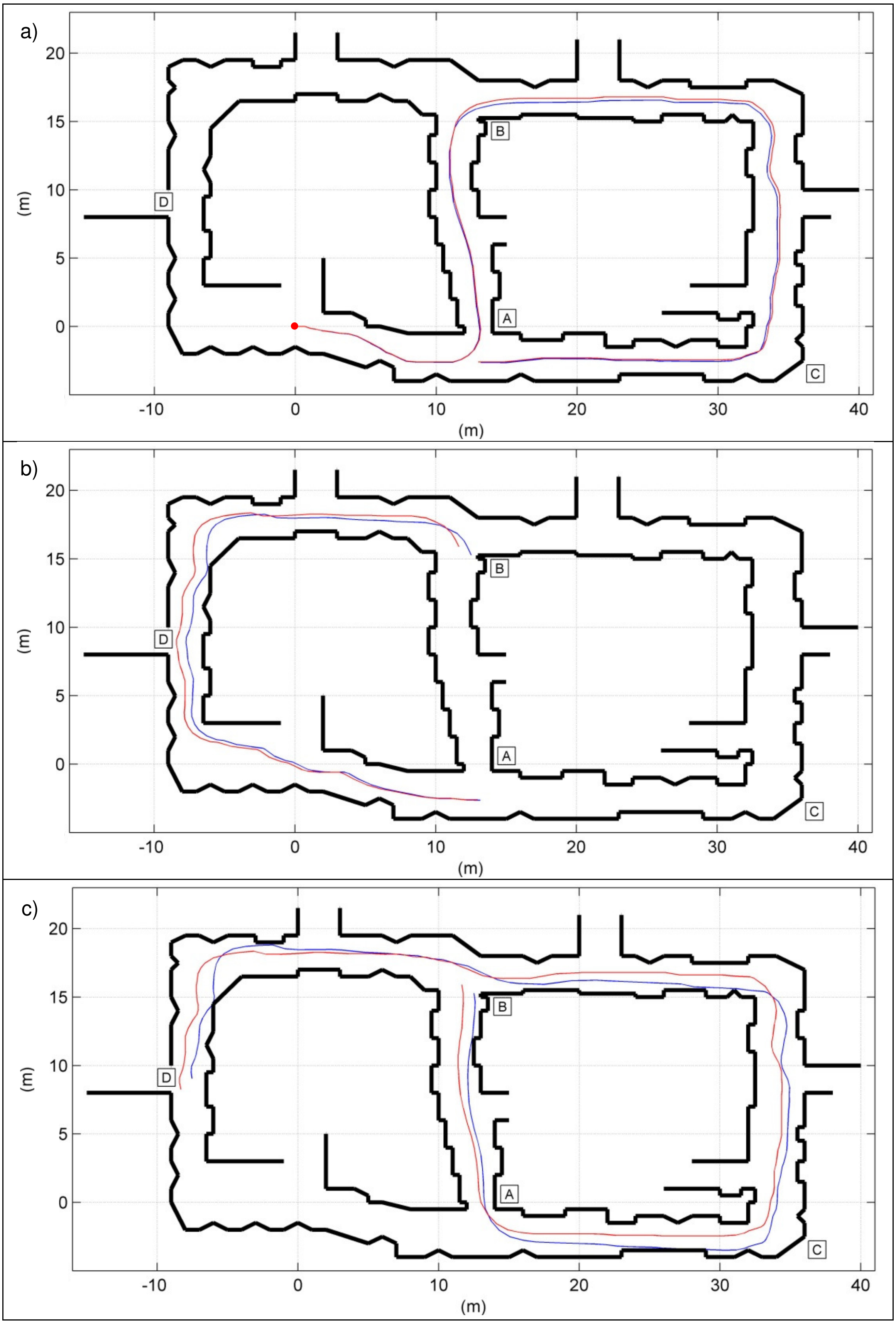

Figure 5.8: Robot trajectories, showing the true path (red line) and estimated path (blue line). a) Robot starts from zero position (red dot), goes through positions A, B, C and A. b) Robot traveling A, D, B. c) Robot completes the course through B, A, C, B and D.

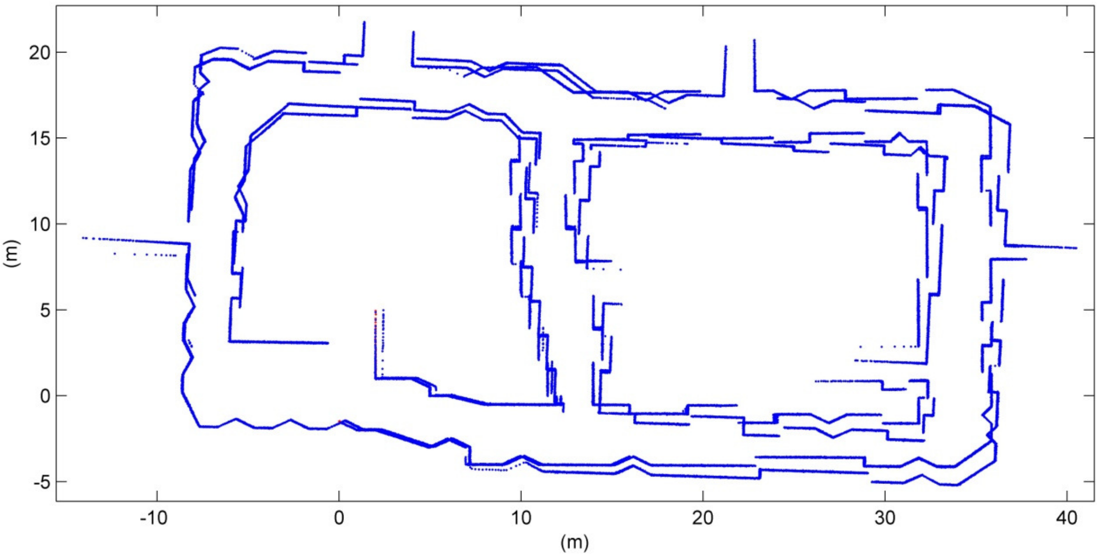

Figure 5.9: Map acquired using scan matching process in the simulated environment.

5.1.3. Scan Matching in Real Data

To analyze the proposed method using real experimental data, it is necessary to have the true robot position in the environment. But this would require expensive motion detector sensors and an external measurement, which are usually not available. Thus, in this work, the analysis made with real data collected from the literature focuses on the study of the convergence on DE optimization. As was seen in Section 4.4.1, the fitness function to minimize has been defined as $\boldsymbol{A}$-*score*($\boldsymbol{p}$), which should be close to zero for a good match.

The performance of the proposed method is evaluated with real data from four experiments from the literature.

The first experiment comes from the second floor of the Duke University Computer Science Department (the second floor of the D-Wing of the LSRC building) [8]. This contains 551 robot poses and uses a LRF with 8m maximum range. This data includes odometry information, which is removed to show the efficiency of the presented approach using a single LRF. Figure 5.10 shows the fitness function value for each of the 550 robot displacements, acquired using the proposed method. The fitness value will rarely be close to zero, because two consecutive will never be the same, unless the robot does not make any movement and the sensors didn't have uncertainties. The fitness values for "D-Wing" experiments show some peaks. Empirically, it is possible to state that values below 30 have a greater probability of convergence in DE optimization.

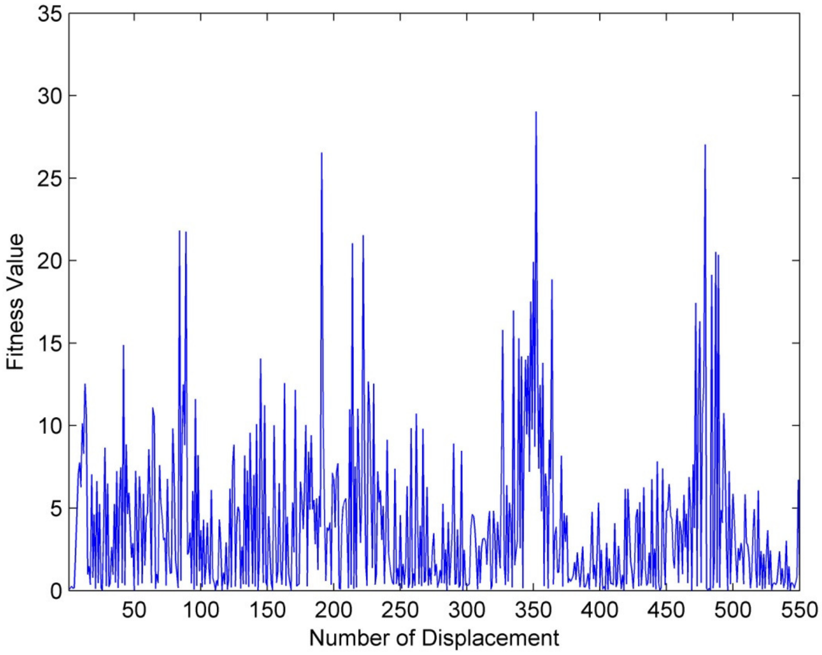

Figure 5.10: Fitness values for "D-Wing" experiment.

Since all fitness values are below 30, this experiment has a greater probability of not having non-convergence problems. This can be confirmed on the resulting map, shown in Figure 5.11. This figure shows the "D-Wing" experiment, which is a closed loop indoor structured environment. It can be observed also the resulting misalignment due to cumulative error in scan matching.

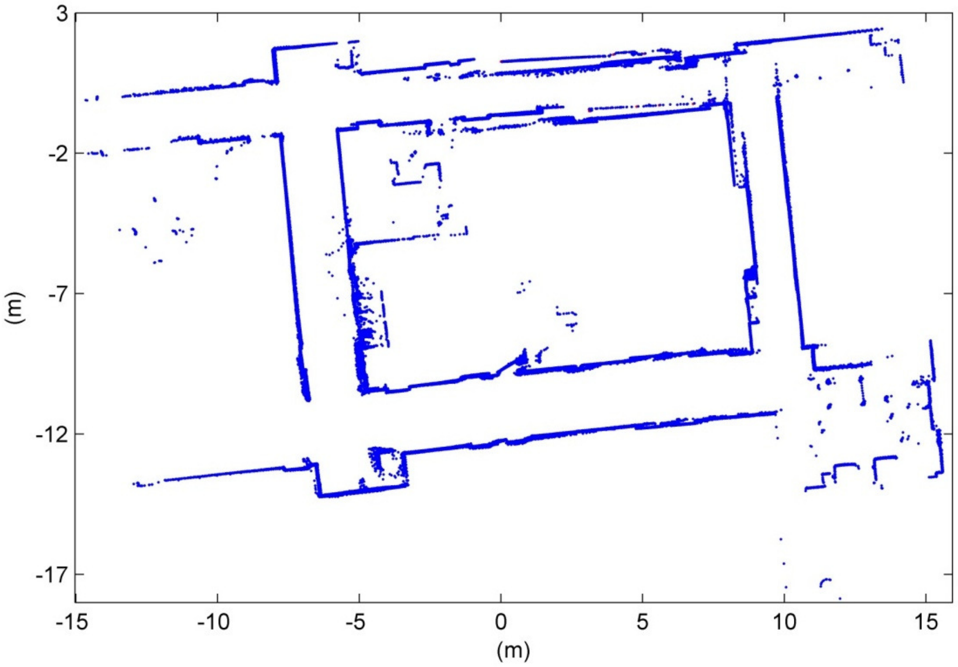

Figure 5.11: “D-Wing” experiment acquired using the proposed scan matching process, without the use of odometry data.

The second experiment also comes from the Duke University Computer Science Department, a long stretch of the second floor of the C-Wing of LSRC – pharmacology [8]. This contains 1,106 robot poses and uses a LRF with 8m maximum range. Once again the odometry data is removed for tow show that it is not needed in the SLAM process using the presented methods. Figure 5.12 shows the fitness function value for each of the 1105 robot displacements, acquired using the proposed method.

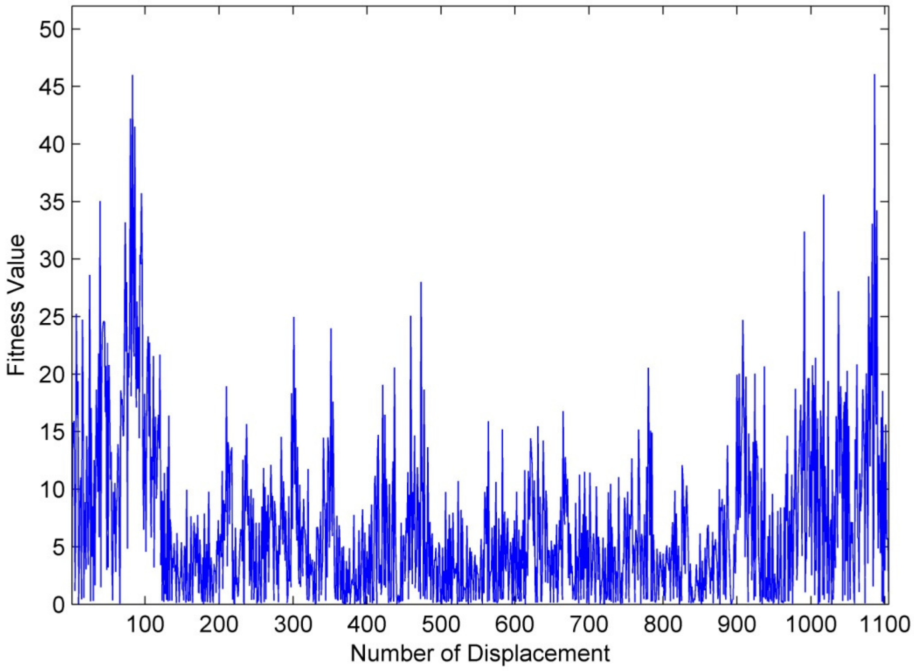

Figure 5.12: Fitness values for “C-Wing” experiment

This figure shows a few peaks (12 in total) larger than the empirically estimated threshold 30. They have a higher probability of resulting in non-convergence in DE optimization. To determine their convergence, they have to be inspected by simple observation.

Figure 5.13 shows the “C-Wing” experiment, which is also a loop closed indoor structured environment. Note also it the misalignment due to cumulative error in scan matching.

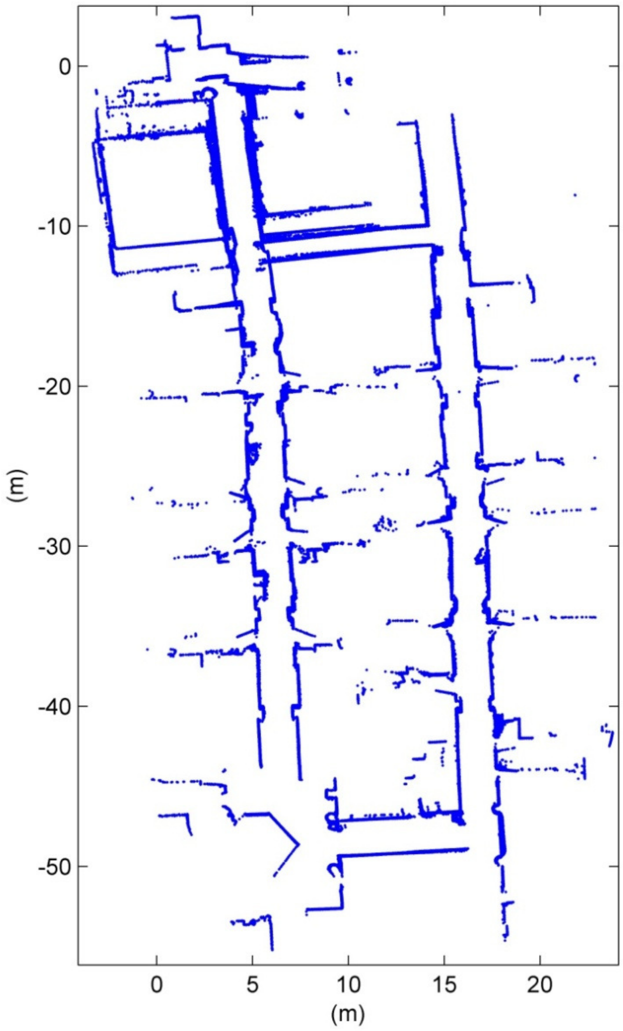

Figure 5.13: "C-Wing" experiment acquired using the proposed scan matching process, without the use of odometry data.

The third experiment comes from the Department of DIIGA at the Engineering University in Ancona [33]. This set contains 9,382 robot poses and uses a LRF of 80m of maximum range. Once again the odometry data is removed. To decrease the computational effort, only 1 robot pose out of every 5 was used in this evaluation, resulting in a number of robot poses of only 1,875. In Figure 5.14 the fitness function value is shown for each of the 1,874 robot displacements, calculated using the proposed method.

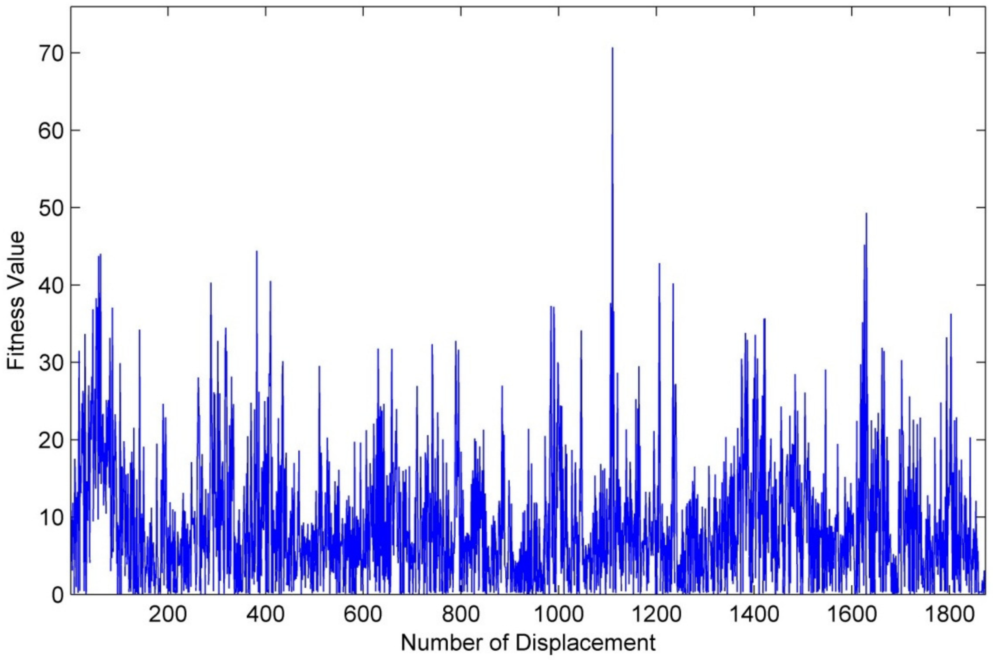

Figure 5.14: Fitness values for “Diiga” experiment.

This figure shows some peaks (50 in total) greater than 30. They have a probability of being a non-convergence in DE optimization.

The Figure 5.15 shows the “Diiga” experiment.

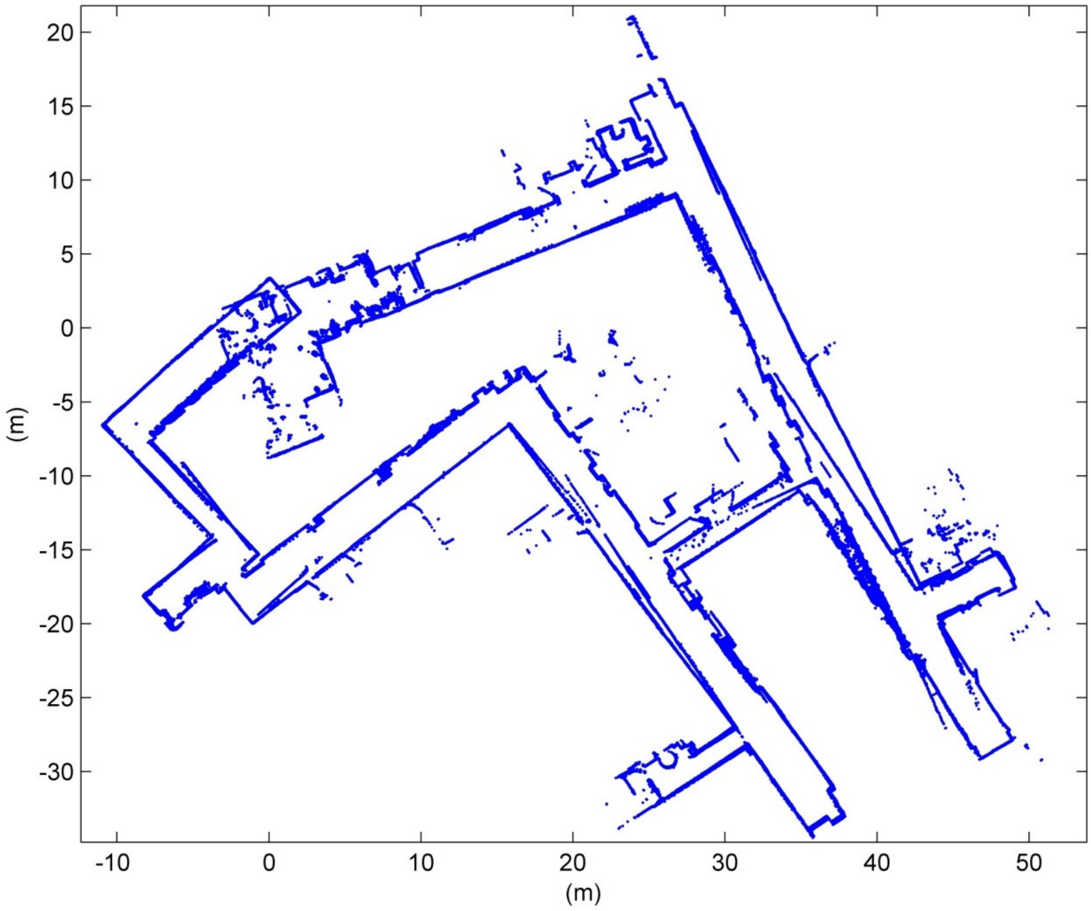

Figure 5.15: “Diiga” experiment acquired using the proposed scan matching process, without the use of odometry data.

The fourth experiment comes from the Kvarntorp mine, near to Örebro, Sweden [36]. It contains 95 robot poses with 3D scans; again the odometry data was removed.

Figure 5.16 shows the fitness function value for each of the 94 robot displacements, acquired using the proposed method. In this experiment, there are 36 fitness values greater than 30. This means that 37% of the matches are poor matches, consequently this produces a poor map as shown in Figure 5.17.

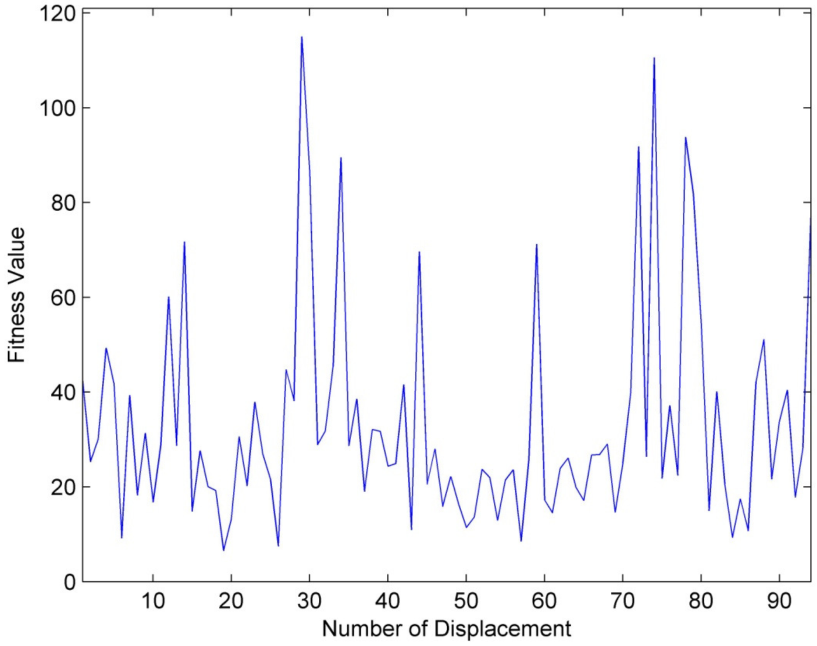

Figure 5.16: Fitness values for “Mine” experiment

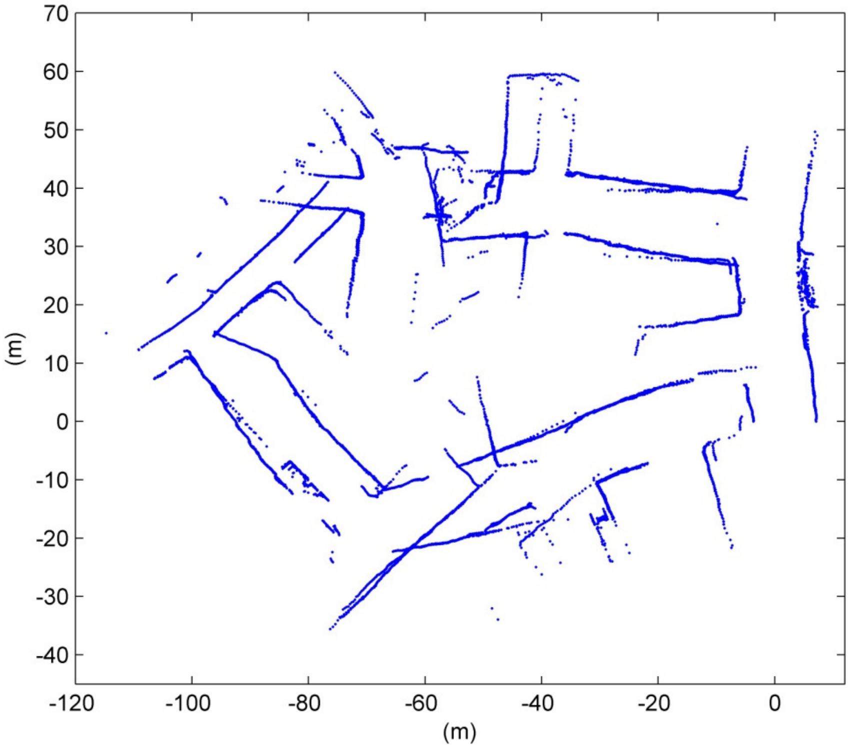

Figure 5.17: Poor map of “Mina” experiment.

This poor map is because scans were taken out of the search space parameters described in Section 4.4.2, Table 4.4. The simple solution that comes out is to increase these parameter ranges. But this is a risky solution, because this could lead to false positives and the map would end up worse.

In this case de adopted solution was to increase the NTD cell size to 4000mm (originally it was defined to 1000mm, as described in Section 2.3.3). This adopted solution has the effect of blurring the NDT representation and consequently facilitates the optimization convergence. However, this scaling increase only helps the translation optimization; rotations, on the other hand, can't be scaled. Thus to solve this problem, misaligned matches are ran individually on DE optimization, increasing or decreasing the rotation search space depending on the case.

Obviously, this is computationally intensive, but it is necessary to allow the proposed method to be successful even in the situations where the matching process is not guarantee.

The resulting map, after increasing the NTD cell size, but before running individually misaligned matches, was already shown in Figure 5.17. The map after running individually misaligned matches and varying the rotation search space is shown in Figure 5.18.

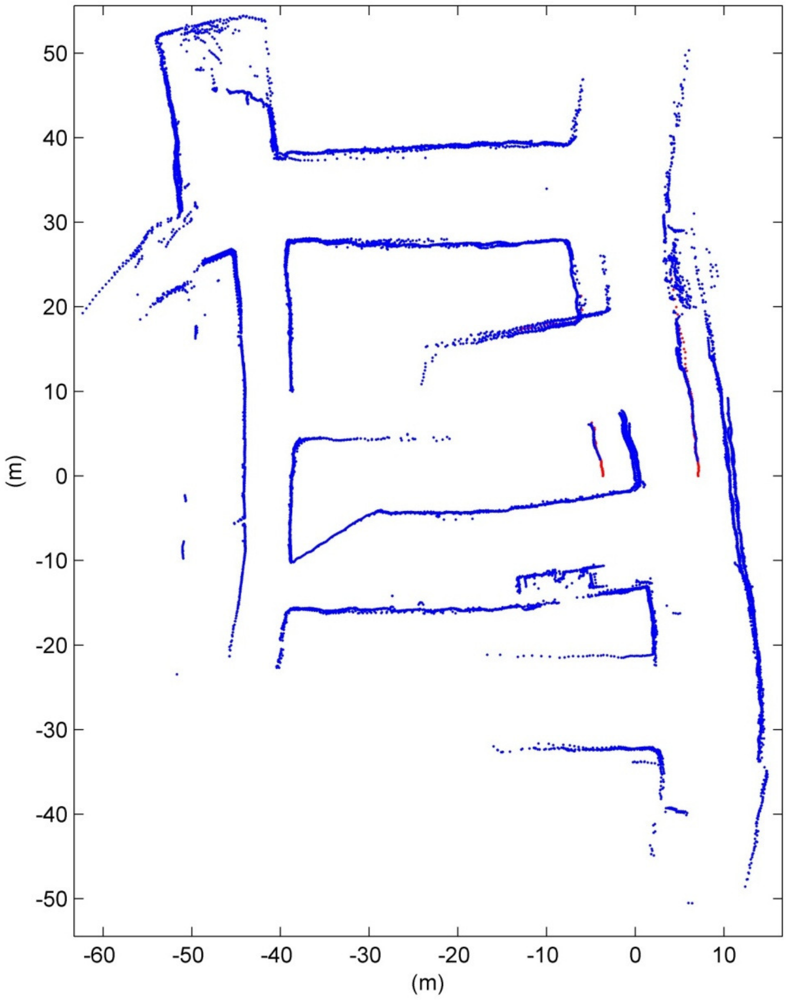

Figure 5.18: “Mina” experiment acquired using scan matching process

5.2. Motion Model

The analysis made using the fitness value on DE optimization helps to determine the quality of the obtained map. But there are some fitness values (in special in the “Diiga” experiment) greater than 30. In these cases, it is advised to manually examine these critical parts from the map. This, of course, suggests the need of a non automatic mapping, for such large fitness values. These results can be viewed as a first mapping step. The scan matching errors are managed in a second mapping step: the motion model in DP-SLAM.

In fact, computing the displacement error histogram (using the simulated experiment), they are distributed as Gaussians, as shown in Figure 5.19.

The distribution error in Δy (movement along robot's facing direction) as shown in Figure 5.19(c), can be approximated by a Gaussian with zero mean and standard deviation of 0.02m plus another Gaussian with mean on −0.20m and standard deviation of 0.10m. However, because always the robot movement is along its facing direction (unless it has significant slippage), the distribution error in Δx is different, and can be approximated by a simple Gaussian, as shown in Figure 5.19(a). In the same way, the distribution error in Δθ can be approximated by a Gaussian, as shown in Figure 5.19(b).

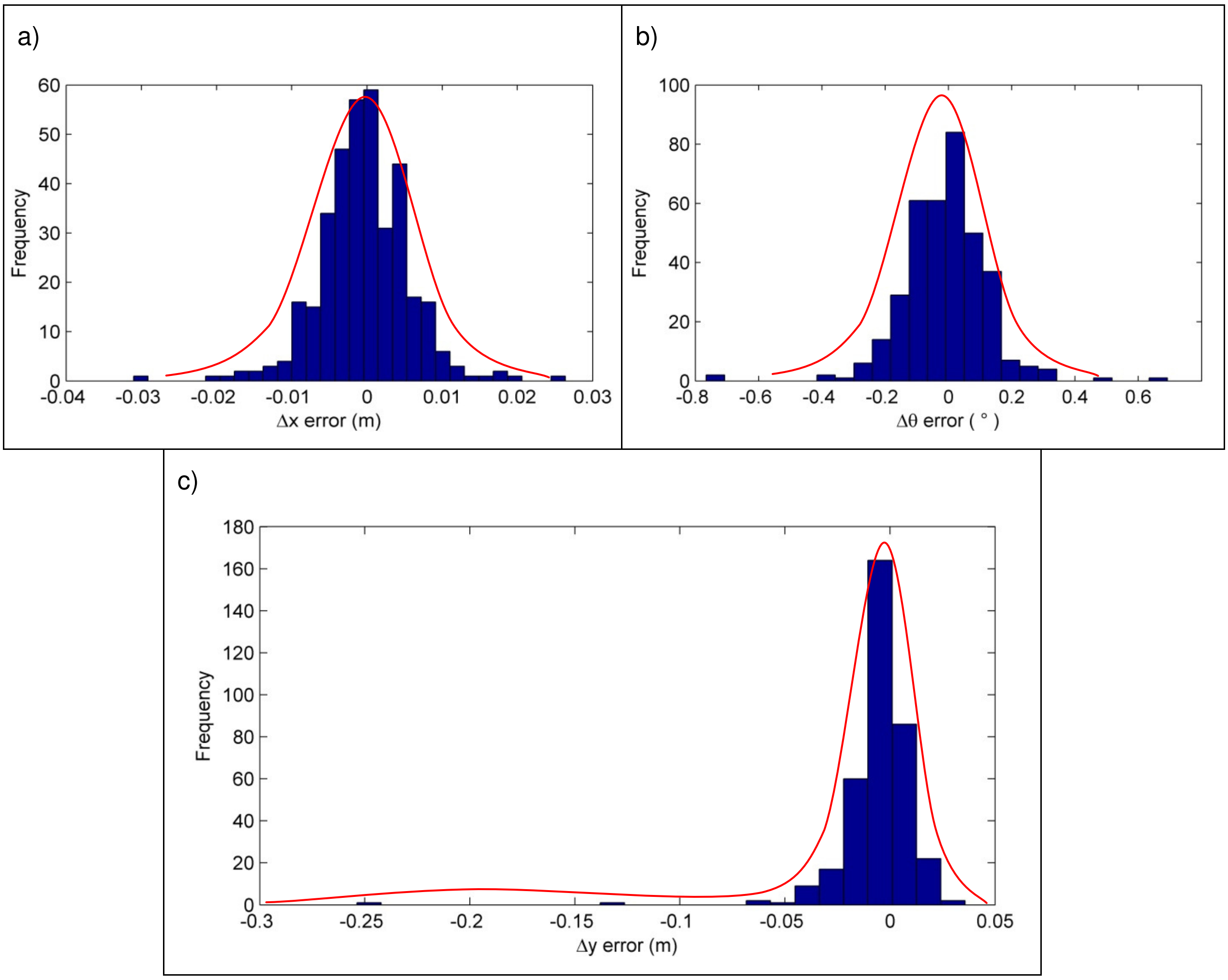

Figure 5.19: Error distribution in displacement: a) Δx, b) Δθ and c) Δy

These displacement errors in Δy are evident when looking in detail the maps acquired.

Figure 5.20 shows an example of this displacement error, taken from the experiment on a simulated environment from Section 5.1.

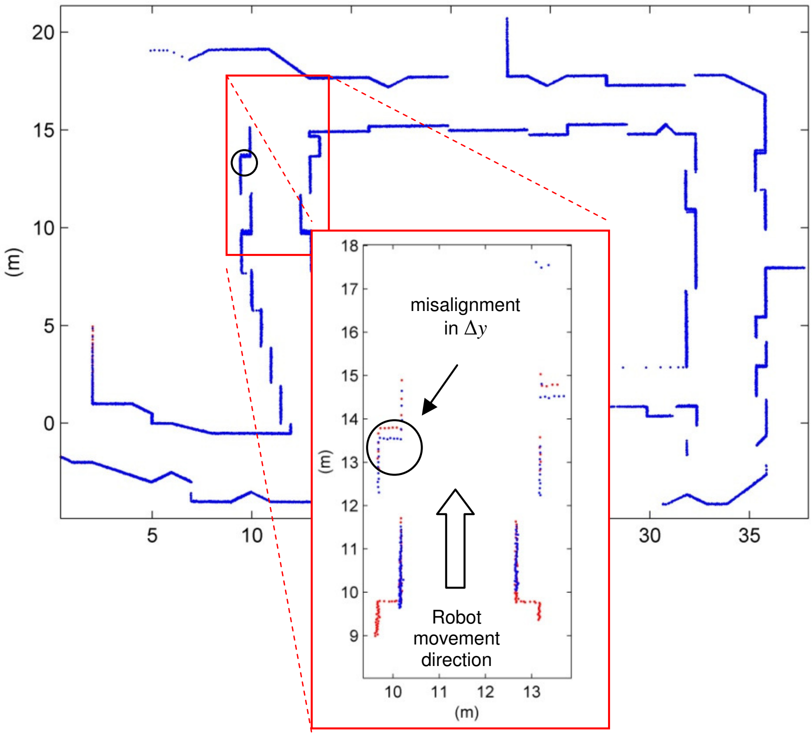

Figure 5.20: Misalignment in Δy (respect to the current robot position).

The averages and standard deviations of these three movements are the parameters used in the proposed motion model. It is expected that scan matching in real data follows the same distribution shapes, but it doesn't mean that the parameter values are the same.

The most complicated error parameter is related to the robot's face displacement, Δy. One important thing observed in real data, using the proposed scan matching, is that error displacements in this movement direction could depend on robot velocity. Figure 5.21 shows this idea: after a robot displacement in the time step Δt, the actual position is $R_t = [Rx_t \ \ Ry_t \ \ R\theta_t]$, but the scan matching gives a displacement $\boldsymbol{p} = [\Delta x \ \ \Delta y \ \ \Delta\theta]$ where Δy is near to cero or, in the best case, a value much lower than the actual displacement Δy_R. Thus the mean of the small

Gaussian in Figure 5.19(c), would be in $-\lambda$, where $\lambda = |\Delta t{\cdot}V_R|$ and V_R is the robot velocity.

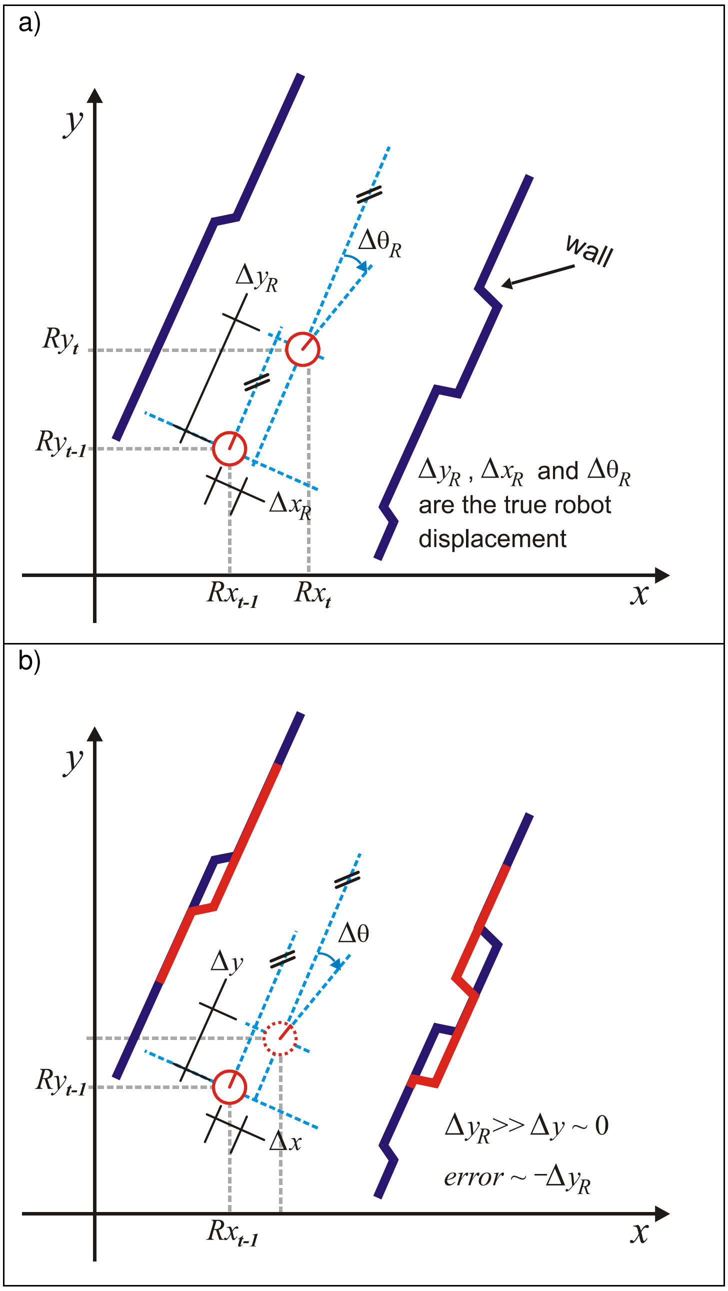

Figure 5.21: a) The true robot displacement. b) The displacement given by scan matching, showing that the most common error is in Δy.

Since one of the objectives in this work is no to use odometers, such robots with a single LRF cannot directly measure their velocity. However, it is possible to use the instants when the scans were taken and, in this way, compute the robot velocity(estimated using the displacement between scans on *t-2* and *t−1* and the time interval between them). Note, however, that this value need not be exact and could be approximated; this is an advantageous consequence of using a probabilistic approach.

Misalignments in Δy were also found in real data, as shown in Figure 5.22. Another important thing observed in real data is the fact that large displacement in Δy corresponds to small rotations $\Delta\theta$; this behavior is shown in Figure 5.23. This fact gives a restriction to the proposed motion model. Thus only when scan matching reports a small rotation, the motion model will be able to sample from the small Gaussian in Figure 5.19(c). The standard deviation assumed for this small Gaussian is $\sigma = (\lambda/2)$. On the other hand, the higher Gaussian in the same figure is assumed with zero mean and standard deviation: $\sigma = 0.02\text{m} + abs(\Delta y/10)$.

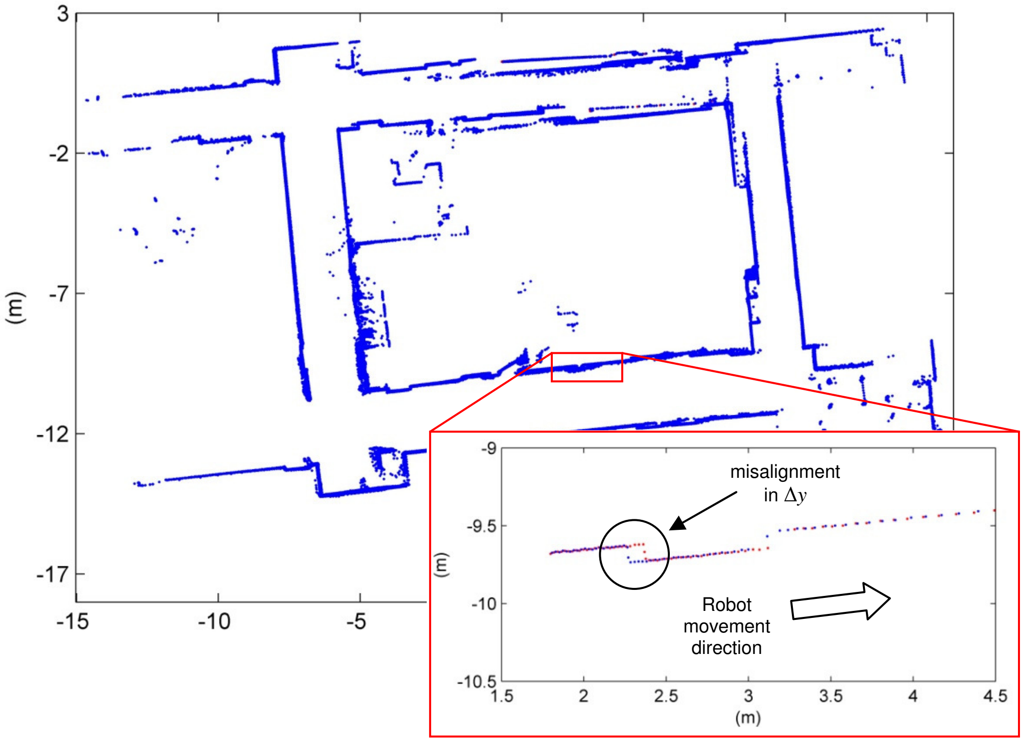

Figure 5.22: Misalignment in Δy (D-Wing experiment)

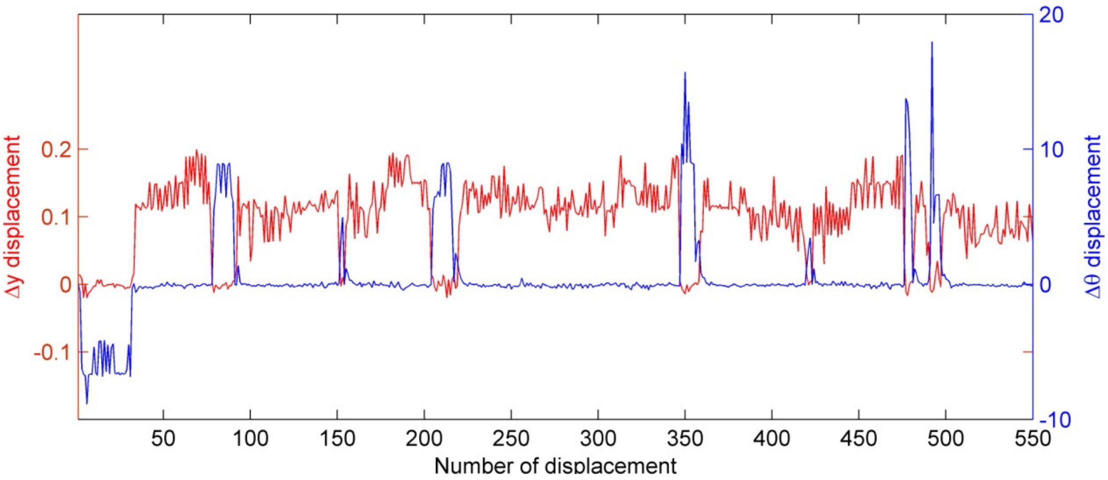

Figure 5.23: Larger displacement in Δy correspond to small rotations $\Delta\theta$ (D-Wing experiment).

The displacement error in Δx follows the distribution shape from Figure 5.19(a), which is simpler than the Δy error distribution, since it could simply be approximated by a zero mean Gaussian with standard deviation σ = 0.02m + *abs*(Δx/10). In the same way the error in $\Delta\theta$ is approximated by a zero mean Gaussian with σ = 0.5°+*abs*($\Delta\theta$ /10).

The final motion model is shown in Table 5.1, Table 5.2 and Table 5.3. Note that this algorithm is a more detailed version than the one from Table 4.7.

Table 5.1: Proposed Scan Matching Motion Model.

Sample SM Motion Model algorithm (R_{t-1}, u_t)	
1:	$\Delta\hat{x} = \Delta x + sampleA\left(0.02\text{m} + abs(\Delta \text{x}/10)\right)$
2:	$\Delta\hat{y} = \Delta y + sampleB\left(0.02\text{m} + abs(\Delta \text{y}/10), \lambda, \Delta\theta\right)$
3:	$\Delta\hat{\theta} = \Delta\theta + sampleA\left(0.5° + abs(\Delta\theta/10)\right)$
4:	$d = \sqrt{(\Delta\hat{x})^2 + (\Delta\hat{y})^2}$
5:	$\alpha = R\theta_{t-1} - \pi/2 + \text{atan2}(\Delta\hat{y}, \Delta\hat{x})$
6:	$Rx_t = Rx_{t-1} + d\cos(\alpha)$
7:	$Ry_t = Ry_{t-1} + d\sin(\alpha)$
8:	$R\theta_t = R\theta_{t-1} + \Delta\hat{\theta}$
9:	*return* $R_t=(Rx_t, Ry_t, R\theta_t)$

Table 5.2: Approximate algorithm to sampling from normal distribution [1]

```
sampleA: sample_normal_distritubtion ( v )
    return $\frac{v}{6}\sum_{i=1}^{12} rand\,(-1,1)$
```

Table 5.3: Approximated Algorithm to sample from the Δy distribution error.

```
sampleB (v, λ , Δθ)
    $x = rand(0,1)$
    if $(x > 0.95)$ && $(\Delta\theta < 3°)$
        return   sampleA(λ/2) + λ
    else
        return   sampleA(v)
```

5.3. DP-SLAM

From the incorporation of the motion model in the DP-SLAM algorithms, improved 2D maps are now obtained from the simulated environment and the experimental ones taken from the literature.

Figure 5.24 shows the simulated environment as 2D grid map. This map was acquired using 1000 particles in the LSLAM and 1500 particles in the HSLAM, with a resolution of 50mm.

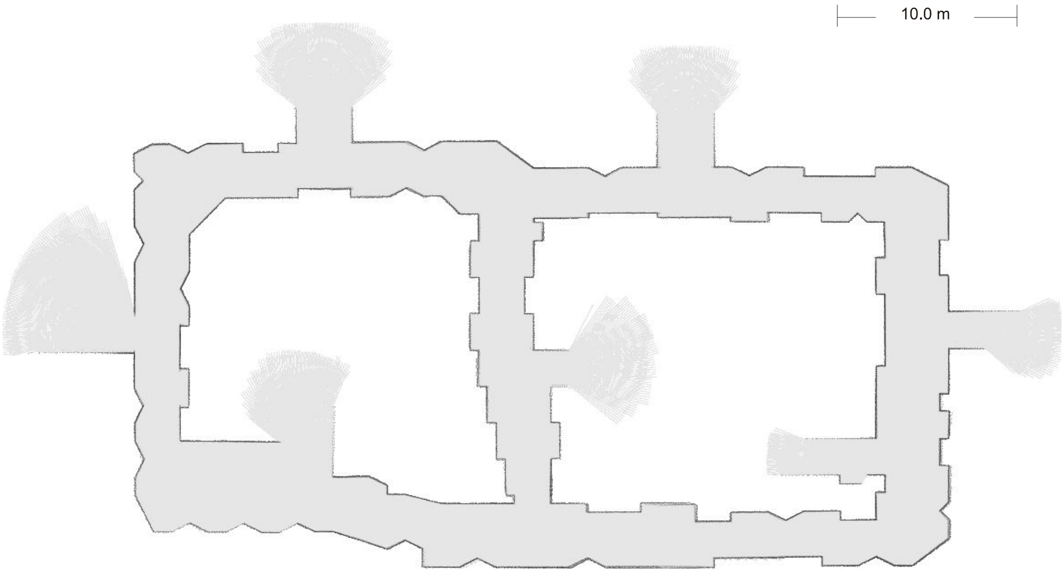

Figure 5.24: 2D grid map from the simulated environment experiment, obtained with DP-SLAM using the proposed motion model.

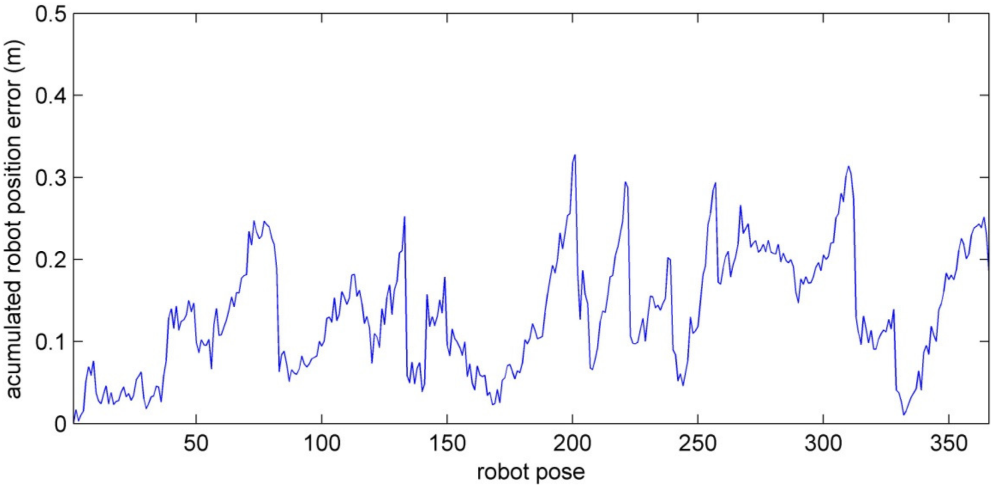

Figure 5.25: Accumulated error position obtained with DP-SLAM using the proposed motion model, on simulated experiment.

Figure 5.25 shows the accumulated error in robot position. Comparing this figure with Figure 5.7, it is clear that the error does not increase with time, being kept mostly under 300mm. Figure 5.26, Figure 5.27 and Figure 5.28 show the histograms of the errors in Δx, Δy, $\Delta \theta$.

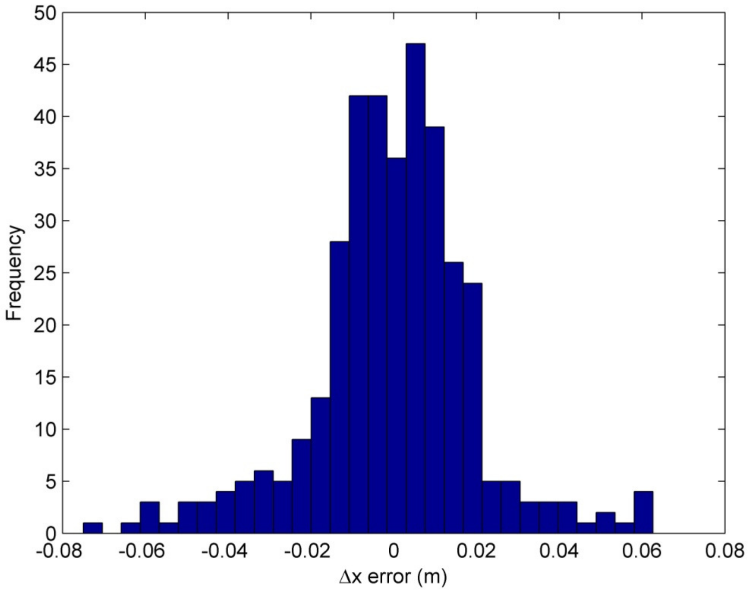

Figure 5.26: Error distribution in Δx obtained with DP-SLAM using the proposed motion model.

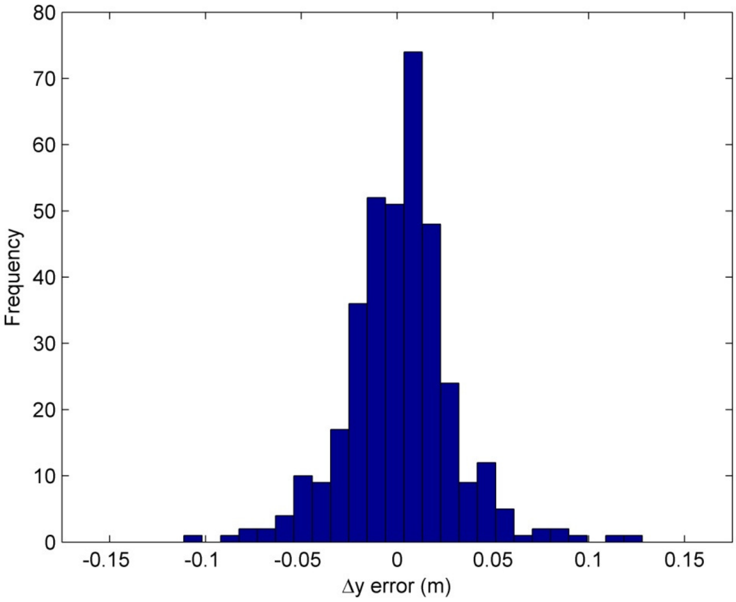

Figure 5.27: Error distribution in Δy obtained with DP-SLAM using the proposed motion model.

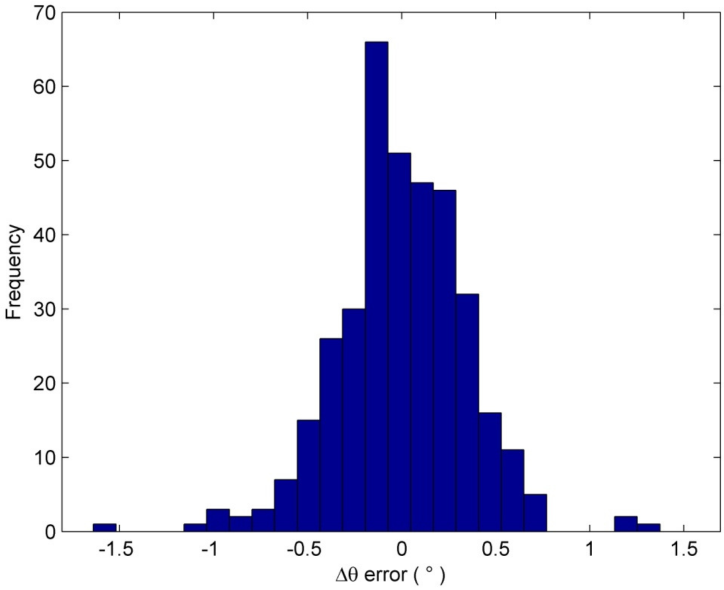

Figure 5.28: Error distribution in $\Delta\theta$ obtained with DP-SLAM using the proposed motion model.

Figure 5.29 shows another simulated environment that was also shown in Figure 3.11. The difference between them is that now the map is better because it was built using hierarchical DP-SLAM. Here, 1000 particles were used in the LSLAM and 1000 particles in the HSLAM, with 50mm of resolution.

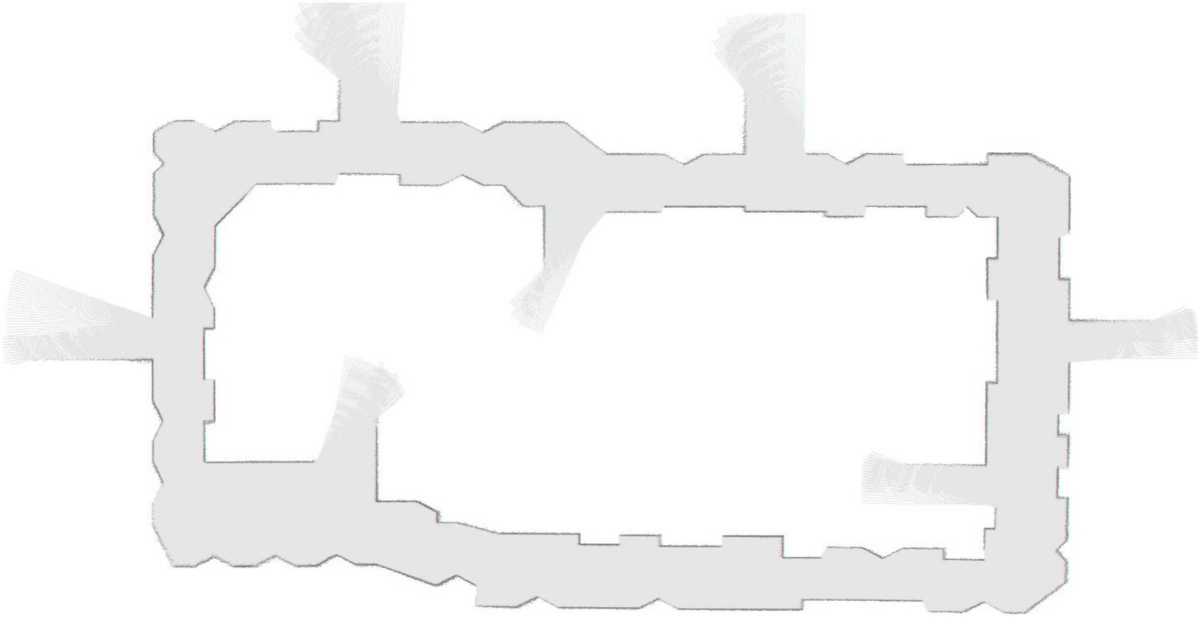

Figure 5.29: 2D grid map from simulated environment presented in Figure 3.11.

Let’s now apply the proposed method to the experimental data from the literature. The first real experiment (D-Wing) in the form of 2D grid map is shown in Figure 5.30. This map was acquired using 500 particles in the LSLAM and 600 particles in the HSLAM, and it has a resolution of 50mm.

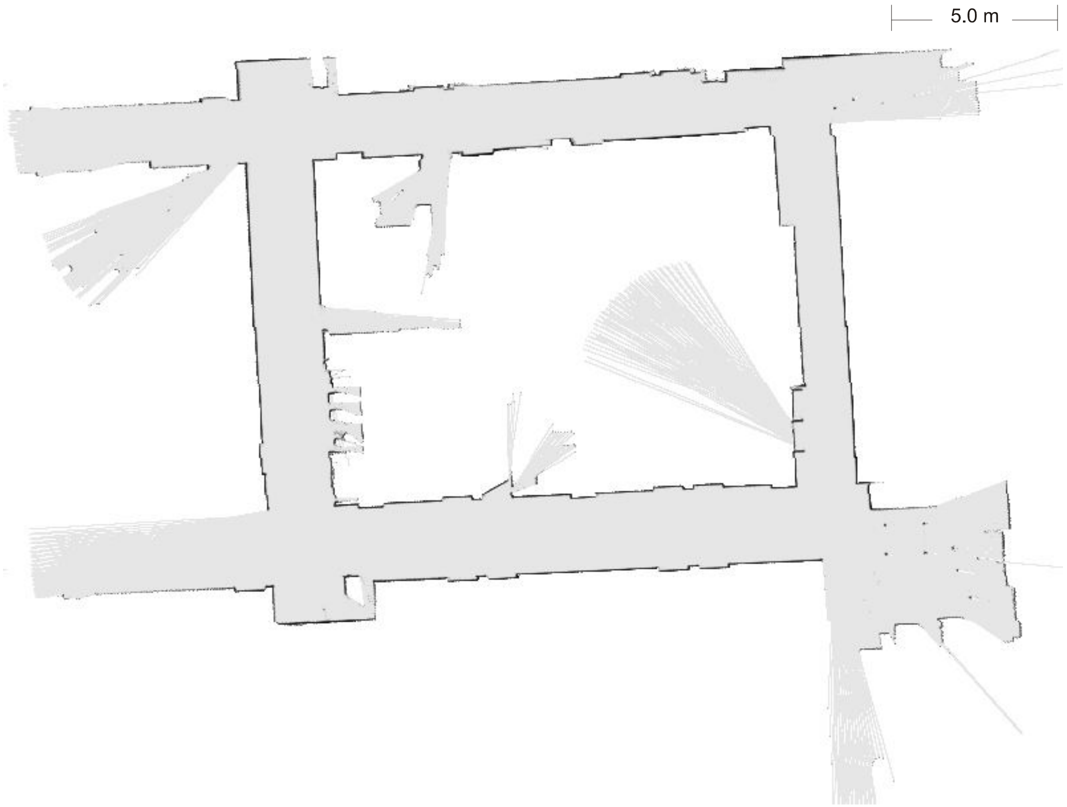

Figure 5.30: 2D grid map of “D-wing” experiment, acquired with DP-SLAM using the proposed motion model.

The second real experiment (C-Wing) in the form of 2D grid map is shown in Figure 5.31. This map was acquired using 600 particles in the LSLAM and 1,300 particles in the HSLAM, and a resolution of 40mm.

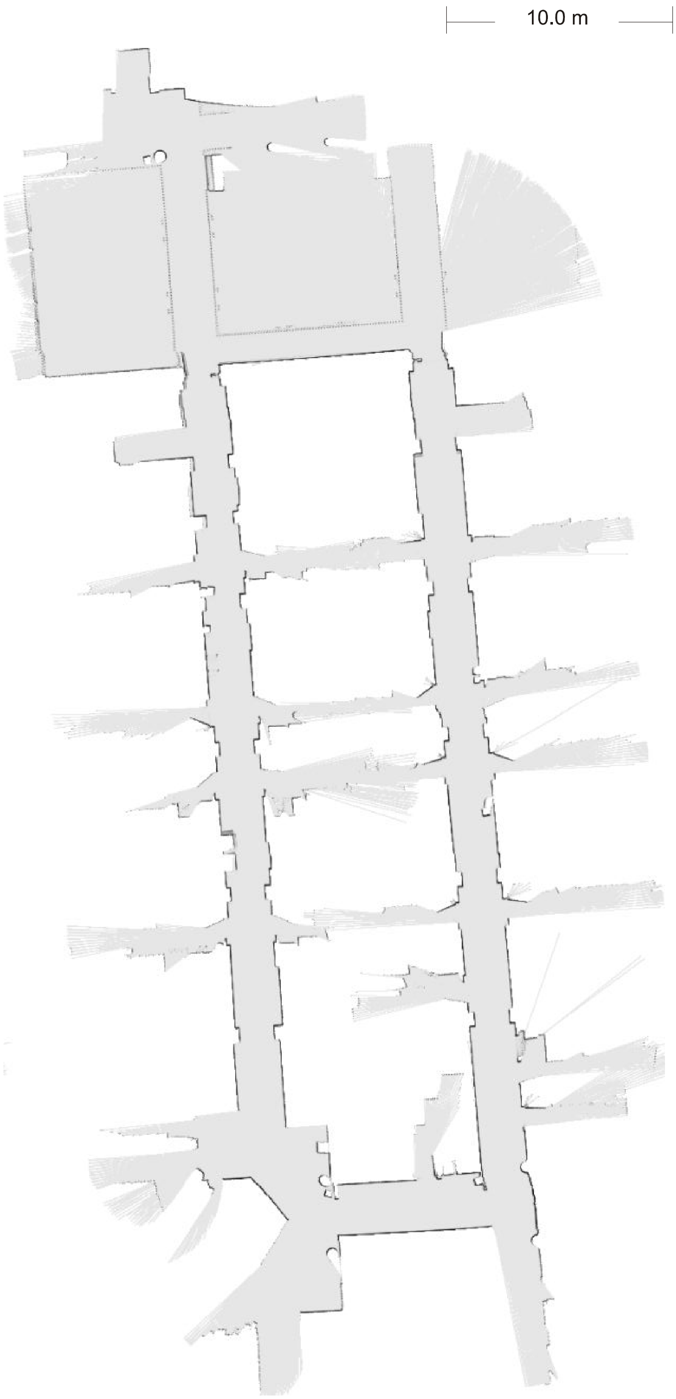

Figure 5.31: 2D grid map of “C-wing” experiment, acquired with DP-SLAM using the proposed motion model.

The third real experiment (“Diiga”) in the form of 2D grid map is shown in Figure 5.32. This map was acquired using 600 particles in the LSLAM and 1800 particles in the HSLAM, and it has a resolution of 40mm.

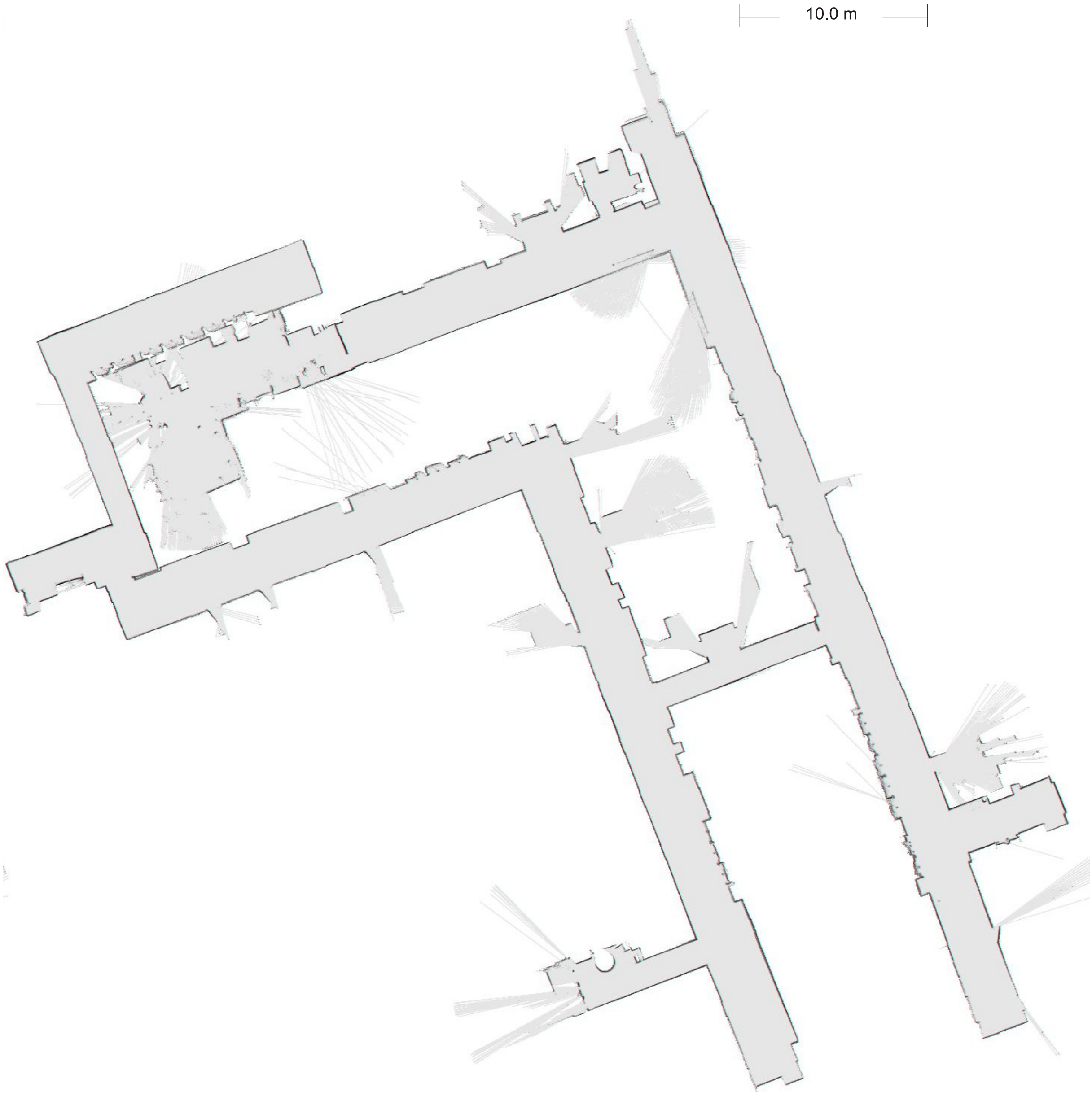

Figure 5.32: 2D grid map of “Diiga” experiment, acquired with DP-SLAM using the proposed motion model.

Finally, the fourth real experiment (“Mine”) in the form of 2D grid map is shown in Figure 5.33. This map was acquired using only the LSLAM with 1,200 particles, with a 50 mm of resolution.

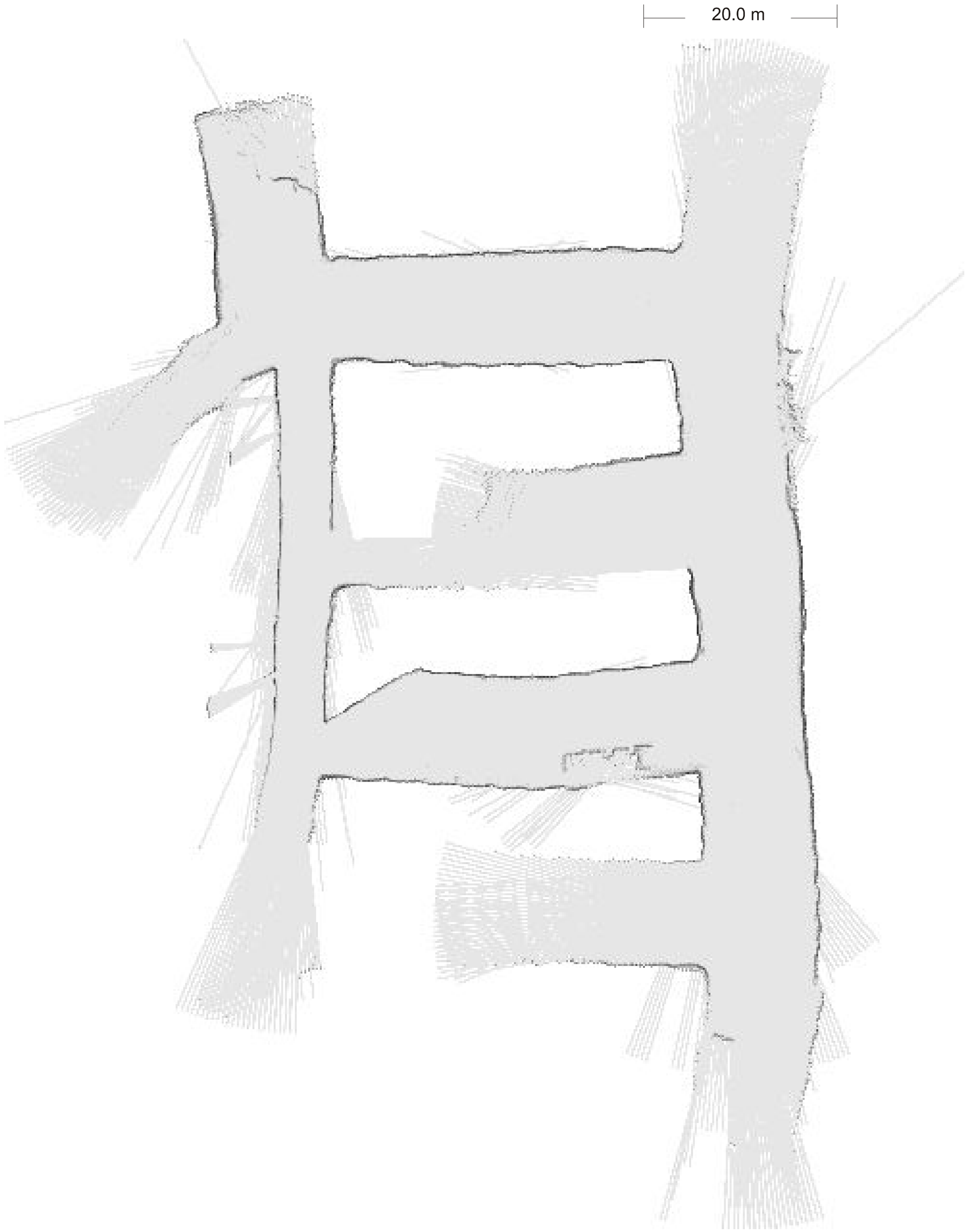

Figure 5.33: 2D grid map of “Mine” experiment, acquired with DP-SLAM using the proposed motion model.

5.4. 3D Mapping

The proposed mapping method using a single RLF uses Scan Matching and DP-SLAM to acquire 2D robot localization. Thus, as long as the parameter ranges shown in Table 4.4 are satisfied, it will be possible to create as well 3D maps. However, almost all 3D data from available from literature, in special in [33] and

[36], are out of these parameter ranges. Most of them are from unstructured environments, and those that are from indoor structured environments only picked up 3D data from sparse robot positions, not frequent enough to satisfactorily apply DP-SLAM without the use of odometer readings.

The previously simulated map is now used to evaluate the application of the proposed methods to 3D. The environment and LRF simulator presented in Section 4.3 is now applied to a 3D map.

As discussed in Section 3.3, trajectory given by the 2D DP-SLAM can be used to project the remaining three-dimensional data. Figure 5.34 shows a 3D map of the simulated environment exposed in previous section. It is represented in the form of a 3D point cloud map.

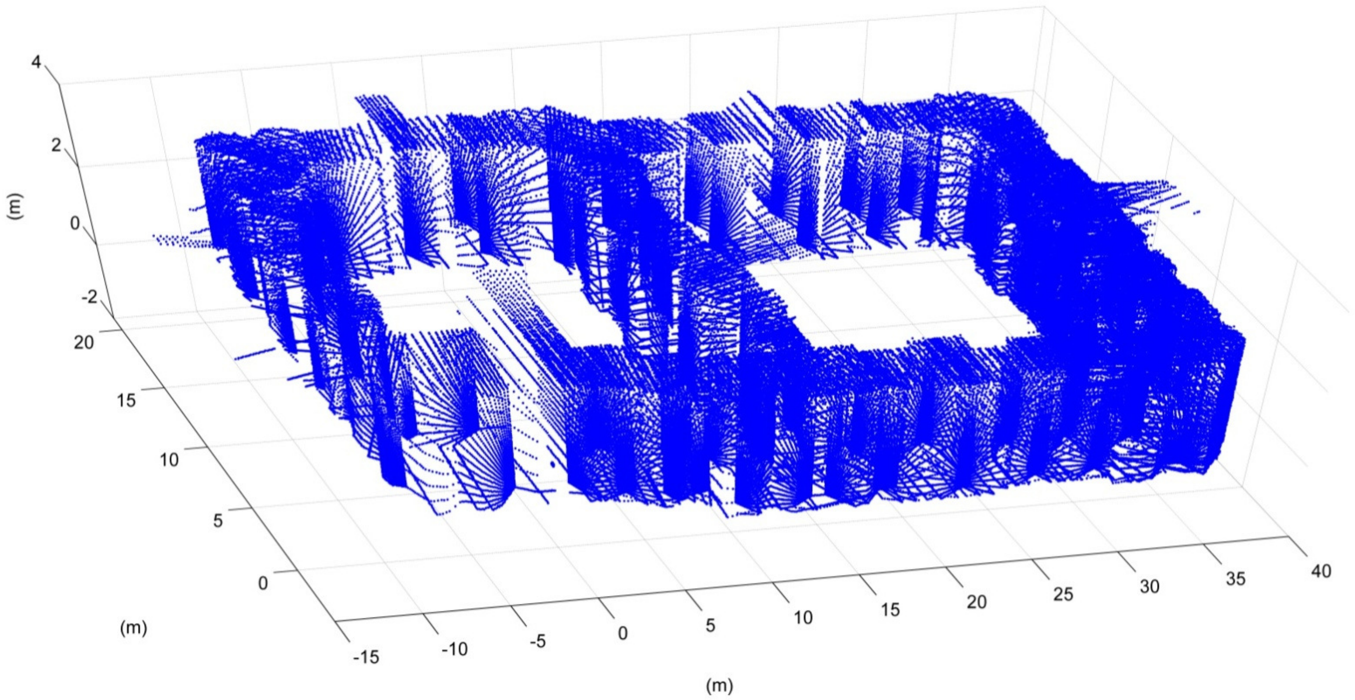

Figure 5.34: 3D point cloud map of the simulated environment.

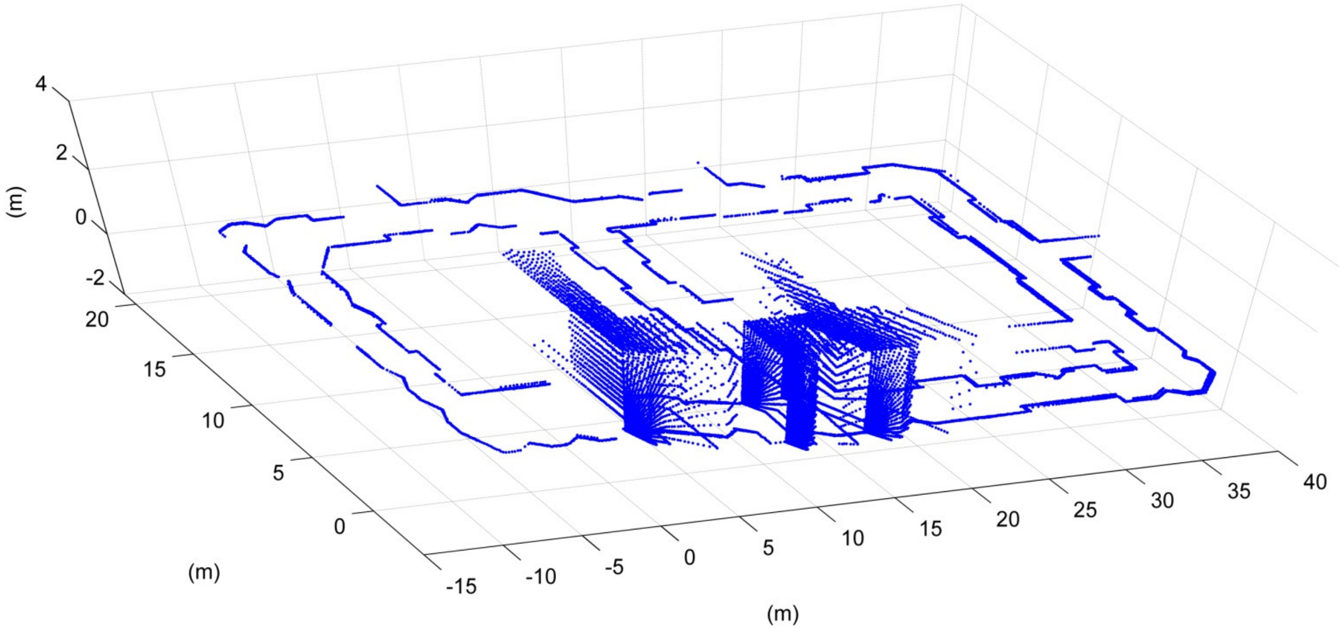

Figure 5.35: 3D point cloud of the simulated environment (only three 3D scans are shown)

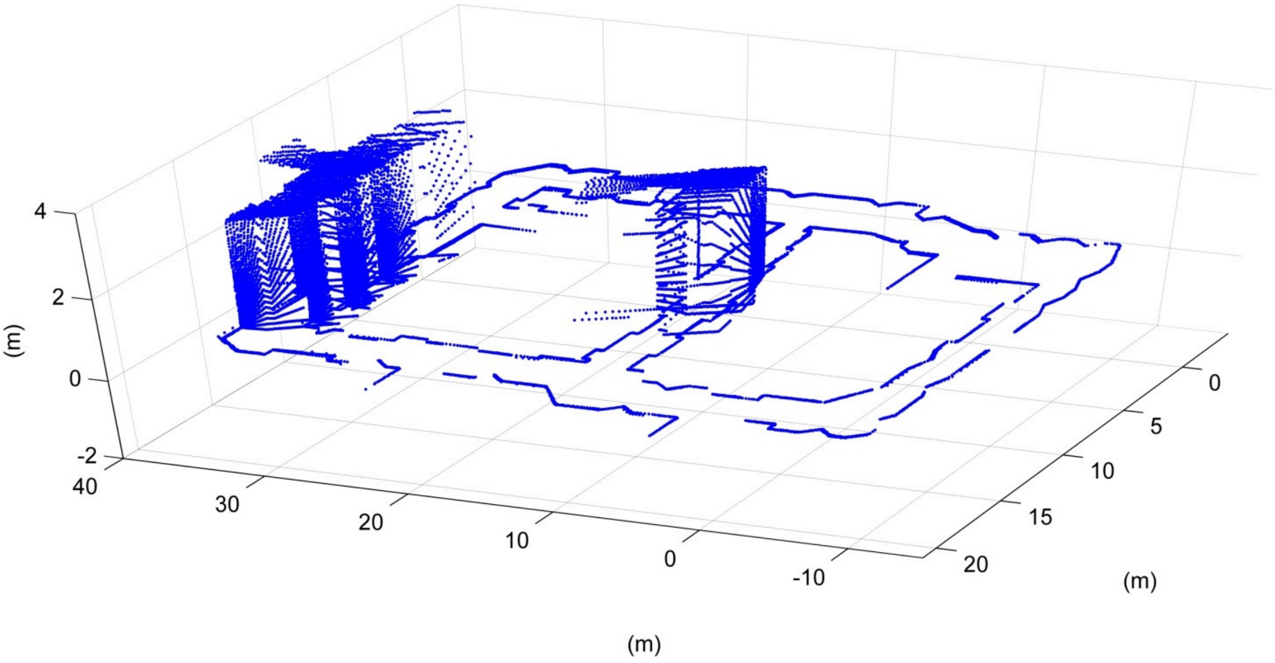

Figure 5.36: 3D point cloud of the simulated environment (only four 3D scans are shown).

The “Mine” experiment, shown in the previous section, is the only 3D real environment used in this work. However, it is an example of 3D data acquired from sparse robot positions; for this reason, its original dimensions were reduced by a factor of four and, after processing, returned to its original values.

Figure 5.37, Figure 5.38, Figure 5.39 and Figure 5.40 show the “Mine” experiment, but now in three dimensions, in the form of 3D point cloud map. This figures show only some 3D scans (not all) for visualization purposes.

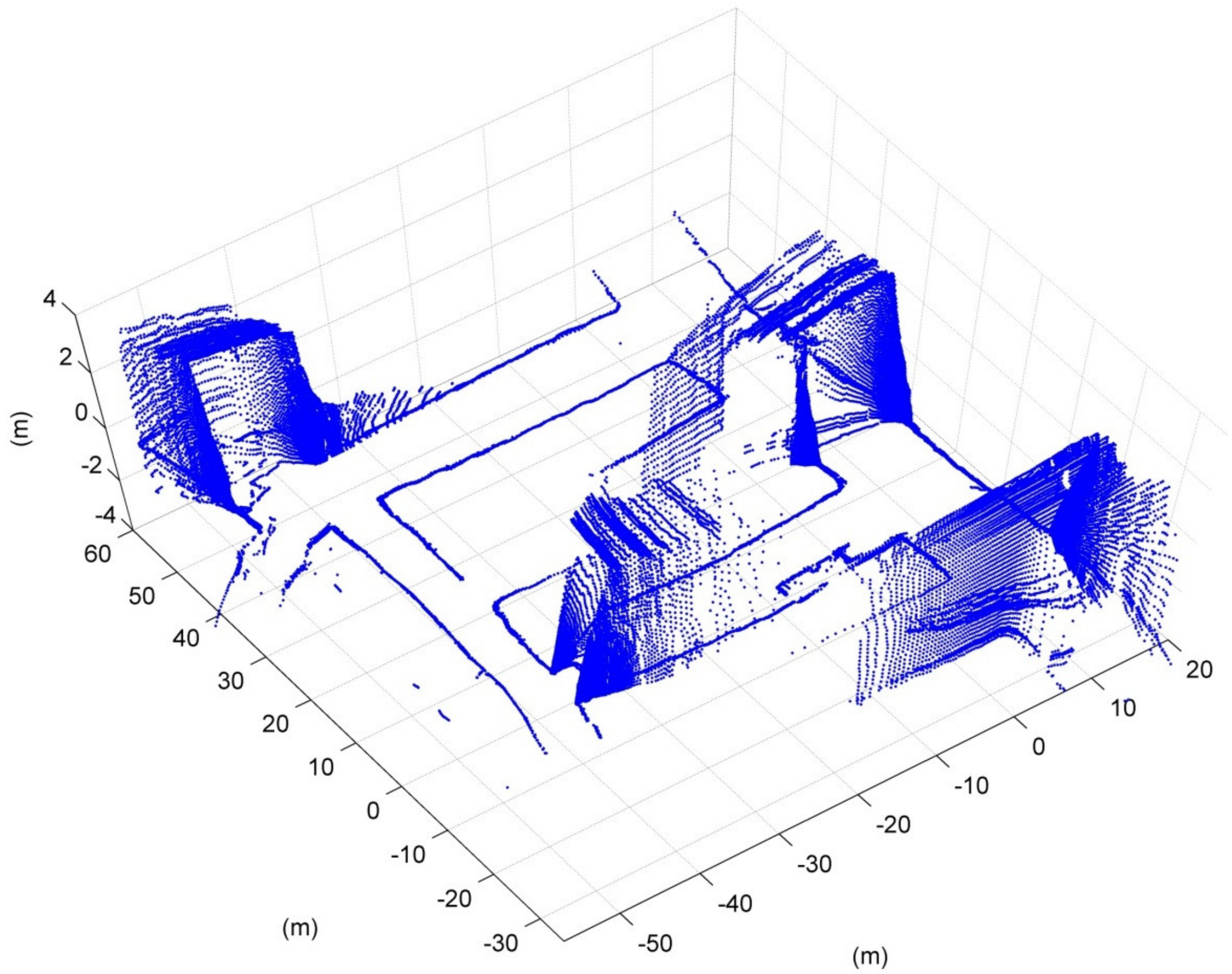

Figure 5.37: 3D point cloud of the “Mine” experiment.

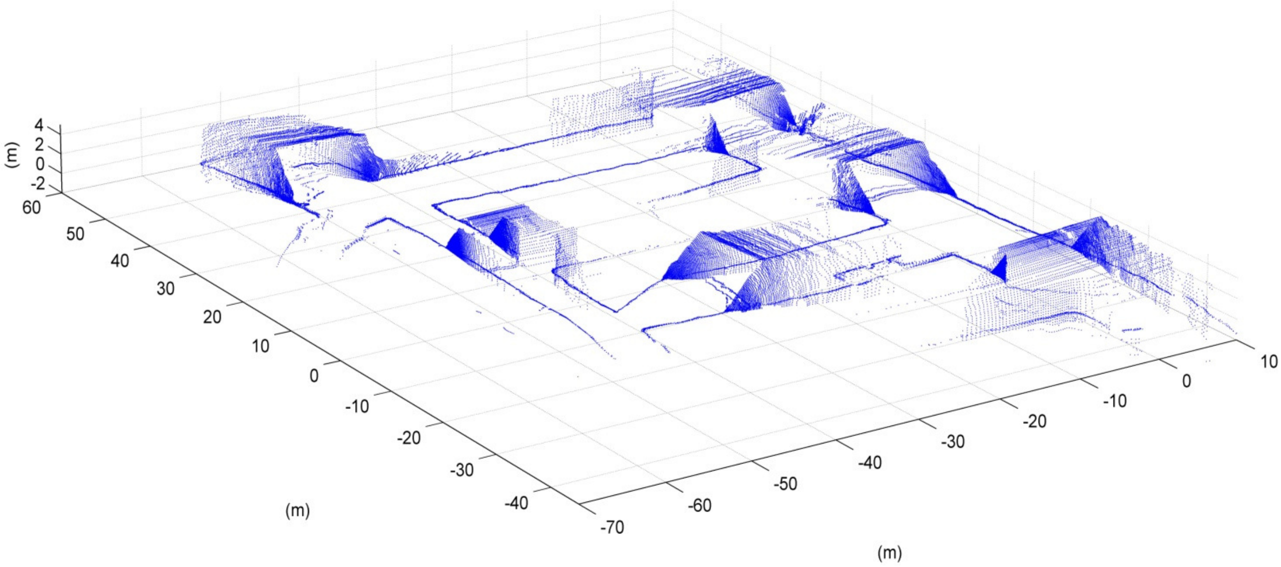

Figure 5.38: 3D point cloud of the “Mine” experiment.

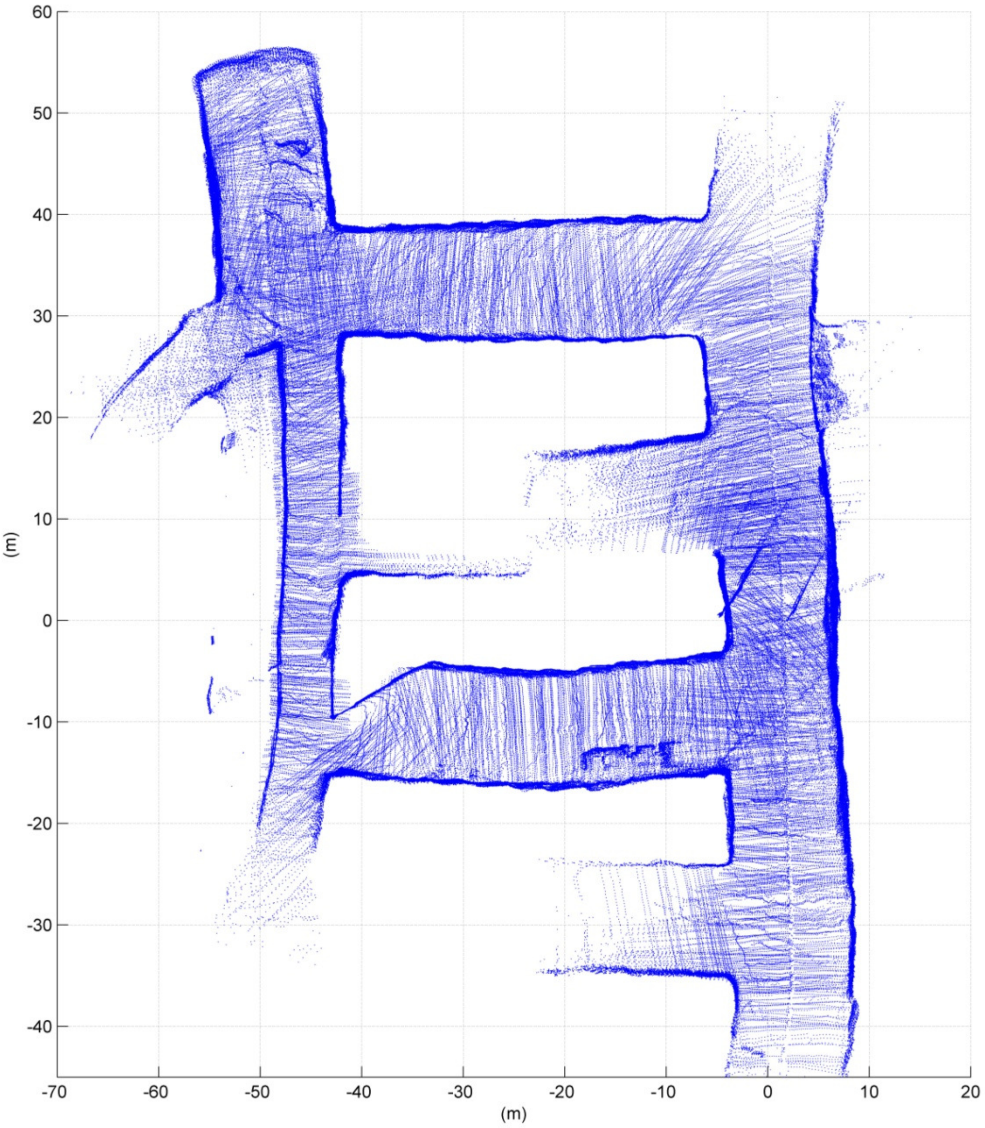

Figure 5.39: 3D point cloud of the “Mine” experiment (top view).

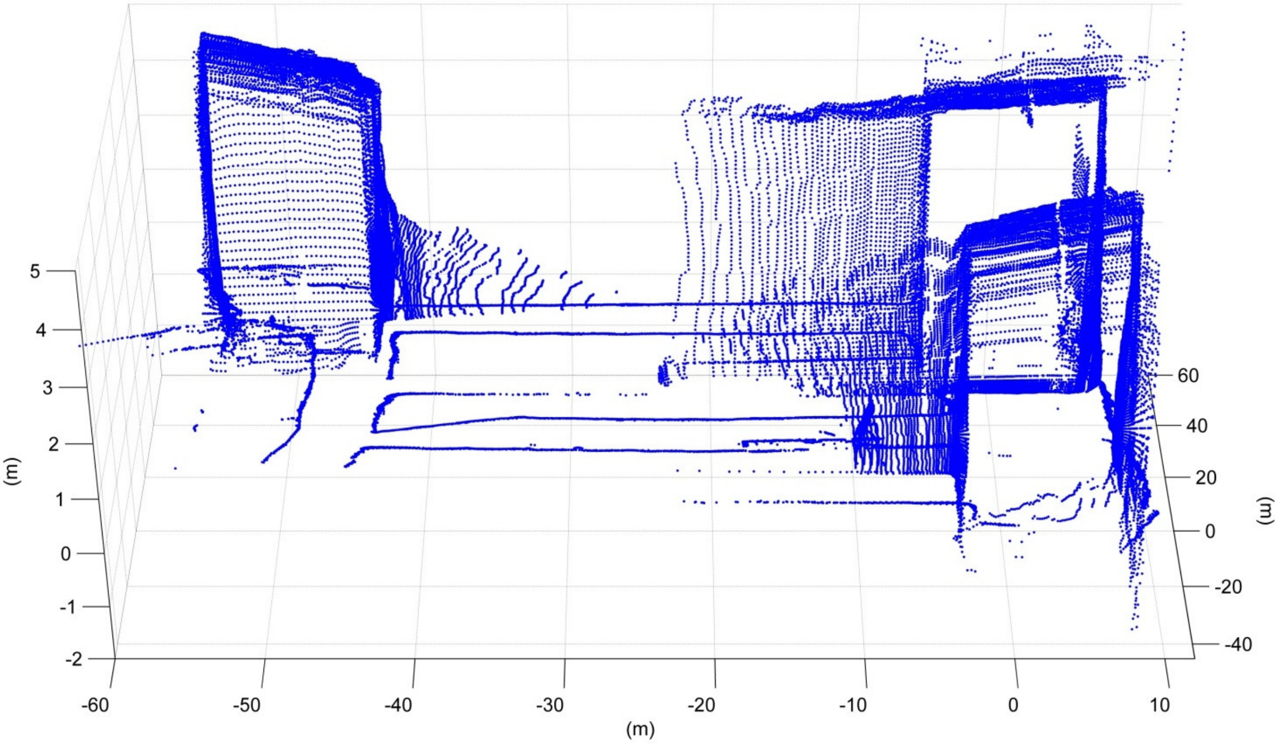

Figure 5.40: 3D point cloud of the “Mine” experiment.