

where matrix $\Pi_{ja_k}^i$ is built recursively as $\Pi_{ja_{k+1}}^i = \Pi_{ja_k}^i - \Gamma_{k+1}^j \Gamma_{k+1}^a{}^T$ and $\Pi_{ja_0}^i$ is initialized to zero.

Then, the local state and the corresponding covariance matrices are updated as follows

$$\begin{aligned}\hat{s}_{k+1}^i &= \hat{s}_{k+1}^{i+} + \Psi_{k+1}^i \Gamma_{k+1}^i M_{k+1}^a{}^{-1} z_{k+1} \\ P_{k+1}^i &= P_{k+1}^{i+} - \Psi_{k+1}^i \Gamma_{k+1}^i M_{k+1}^a{}^{-1} \Gamma_{k+1}^i{}^T \Psi_{k+1}^i{}^T\end{aligned}\quad (5.9)$$

Also, the following matrices are computed recursively, i.e.

$$\Pi_{jl_{k+1}}^i = \Pi_{jl_k}^i - \Gamma_{k+1}^j \Gamma_{k+1}^l{}^T \quad (5.10)$$

for agents $j = \{1, \dots, N-1\}$ and $l = \{j+1, \dots, N\} \setminus \{a\}$. In addition, the matrices for $l < j$ can be computed by symmetry, i.e. $\Pi_{lj_{k+1}}^i = \Pi_{jl_{k+1}}^i{}^T$.

2. If agent a detects agent b , agent b sends the triple $(P_{k+1}^{b+}, \Psi_{k+1}^b, \hat{s}_{k+1}^{b+})$ to agent a which computes the following terms

$$\begin{aligned}M_{k+1}^{ab} &= H_{k+1}^a P_{k+1}^{a+} H_{k+1}^a{}^T + H_{k+1}^{ab}{}^T P_{k+1}^{b+} H_{k+1}^{ab} \\ &+ H_{k+1}^a \Psi_{k+1}^a \Pi_{ab_{k+1}}^a \Psi_{k+1}^b{}^T H_{k+1}^{ab}{}^T + H_{k+1}^{ab} \Psi_{k+1}^b \Pi_{ba_{k+1}}^a \Psi_{k+1}^a{}^T H_{k+1}^a{}^T \\ &+ R_{k+1}^{ab} \\ \Gamma_{k+1}^a &= \Pi_{ab_{k+1}}^a \Psi_{k+1}^b{}^T H_{k+1}^b{}^T + (\Psi_{k+1}^a)^{-1} P_{k+1}^{a+} H_{k+1}^a{}^T \\ \Gamma_{k+1}^b &= \Pi_{ba_{k+1}}^a \Psi_{k+1}^a{}^T H_{k+1}^a{}^T + (\Psi_{k+1}^b)^{-1} P_{k+1}^{b+} H_{k+1}^b{}^T\end{aligned}\quad (5.11)$$

where H_k^a and H_k^{ab} are the Jacobians of output function (5.4) with respect to the state variables of agents a and b , respectively, computed in $\hat{s}_k^{a+}$ and $\hat{s}_k^{b+}$. Once the results of (5.11) are ready, agent a broadcasts to all the other agents the following data, i.e. $(a, z_{k+1}, M_{k+1}^{ab}, \Gamma^a, \Psi_{k+1}^a{}^T H_{k+1}^a{}^T, b, \Gamma^b, \Psi_{k+1}^b{}^T, H_{k+1}^b{}^T)$. Each agent $i=1, \dots, N$ first computes the following matrices

$$\Gamma^j = \Pi_{ja_{k+1}}^i \Psi_{k+1}^a{}^T H_{k+1}^a{}^T + \Pi_{jb_{k+1}}^i \Psi_{k+1}^b{}^T H_{k+1}^b{}^T \text{ for } j = \{1, \dots, N\} \setminus \{a\} \quad (5.12)$$

Then, the local state, the corresponding covariance matrices and matrices $\Pi_{jl_{k+1}}^i$ are finally updated exactly as in (5.9) and (5.10).

3. Finally, if no measurements of any kind are available at time $(k+1)T_s$, the estimates each agent $i=1, \dots, N$ can just rely on prediction, i.e.

$$\hat{s}_{k+1}^i = \hat{s}_{k+1}^{i+}, P_{k+1}^i = P_{k+1}^{i+}, \Pi_{jl_{k+1}}^i = \Pi_{jl_k}^i \quad (5.13)$$

5.1.3 Results

To evaluate the proposed decentralized cooperative localisation algorithm, multiple simulation-based tests have been performed in different conditions. In particular, 200 random trajectories of $N = \{1, 2, 4, 6, 8\}$ *FriWalks* have been generated in a 100 m² room using grids of QR code with a different granularity. The main settings adopted in all tests are listed below:

- Duration of each simulated path: about 100 s;
- Distance d between landmarks over a regular triangle-patterned grid: $\{2, 3, 4\}$ m. These values provide a reasonable trade-off between observability and deployment costs;
- *FriWalks* linear velocity range: $[0, 2]$ m/s;
- *FriWalks* angular velocity range: $[-\pi/2, \pi/2]$ rad/s;
- Encoders sampling period: $T_s = 4$ ms;
- Encoders mean and standard uncertainty values: $\mu_\phi = 0.08 \cdot \Delta\Phi + 0.001$ and $\sigma_\phi = 0.12 \cdot \Delta\Phi + 0.002$, as shown in Figure 25.
- Monocular and stereo camera image acquisition rate: 10 Hz;
- Reading range of the ground-facing monocular camera for QR code detection: 1 m with a horizontal aperture angle of about 40 degrees;

- Covariance matrix associated with camera-based QR-code distance measurements (based on experiments on the field): $R_k^a = \text{diag}(1.6 \times 10^{-3} \text{m}^2, 5 \times 10^{-5} \text{m}^2, 10^{-3} \text{rad}^2)$;
- RGB-D camera (i.e. ASUS Xtion) reading range: from 0.8 m to 3.5 m with a horizontal aperture angle of about 58 degrees;
- Covariance matrix associated with RGB-D position measurements whenever an agent a detects an agent b : $R_k^{ab} = \text{diag}(7.8 \times 10^{-4} (d^{ab})^2 \text{m}^2, 12.5 \times 10^{-3} (d^{ab})^2 \text{m}^2)$, with d^{ab} being the distance between the two agents along the focal axis of the camera.

A sample trajectory estimated with or without collaborative localisation, is shown in Figure 28. In the latter case, four agents move in the same room. However, for the sake of clarity, the trajectories of three of them are not shown in Figure 28 not to overcrowd the picture. In this example, the distance between QR codes (represented as star-shaped markers) is $d = 2$ m. The performance improvement is evident especially in the worst case. On the left side, a sudden drift of the estimated trajectory (probably due to some wheel skidding not promptly compensated by QR code detection) is greatly mitigated by the collaborative approach.

A more extensive and in-depth performance comparison is reported in Figure 29. The various bar diagrams refer to the RMS estimation errors of state variables x_k, y_k and θ_k computed over time and over multiple trajectories for grids of QR codes with a different granularity and when a different number of agents moves in the room. Of course, when just 1 agent is present only QR code based measurements can be performed, i.e. no collaborative localisation is possible.

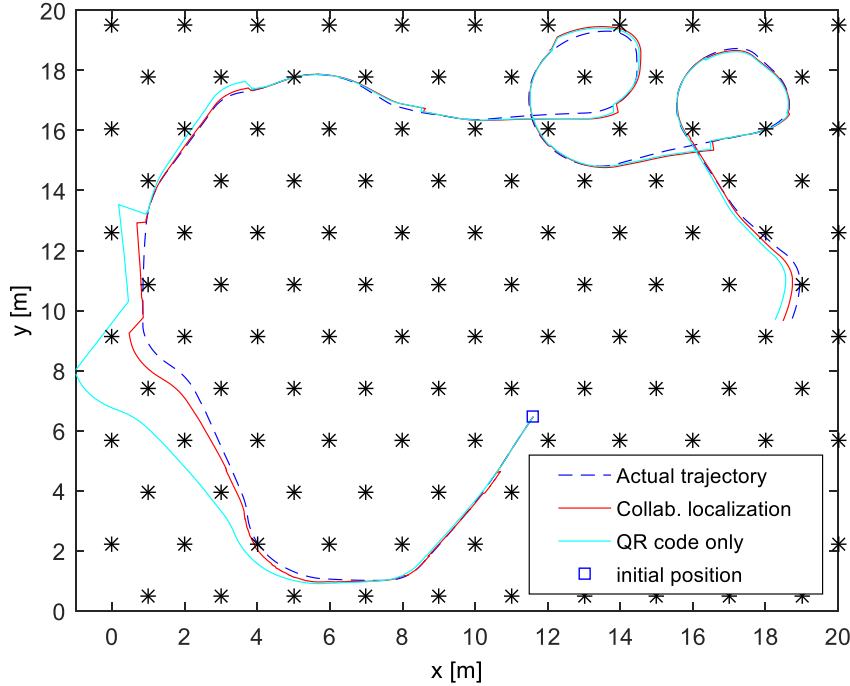


Figure 28: An example of trajectory estimated by the *FriWalk* with wheel odometry + QR code detection with and without the collaborative localisation algorithm when other three agents are present in the room (not shown for the sake of clarity). The distance between QR codes (star-shaped markers) is $d = 2$ m.

The results on the left side of Figure 29 (i.e. Figure 29 (a)) have been obtained using a classic 3-state variable unicycle model, namely assuming that μ_k^i, δ_k^i in (5.1) and (5.2) for $i = 1, \dots, N$, are negligible. The results on the right side of Figure 29 (i.e. Figure 29 (b)) refer instead to the adopted 5-state model. It is worth emphasizing that the assumption that the relative systematic offsets of the encoders is negligible is clearly incorrect, as testified by the results of Figure 25. However, it is interesting to evaluate to what extent using the 5-state variable model can improve the estimation accuracy with respect to position estimators based on a classic unicycle model.

Ultimately, the results of the performance comparison have led to the following conclusions, i.e.

- With both models, by increasing d from 2 m to 3 m and then to 4 m, the RMS errors along axes x and y tend to grow more than twice when $d=3$ m and become between three and four times larger when $d=4$ m. Heading uncertainty follows a similar trend, but degrades more smoothly;
- Using the 5-state model instead of the 3-state one reduces the RMS estimation errors of all state variables of about 20% on average. The improvement is a bit larger (i.e. 25%) when $d=2$ m and slightly worse (i.e. 15%) when $d=4$ m.
- The relative impact of collaborative localisation on both models is approximately the same. In particular, RMS error reduction along axes x and y ranges from about 10% when just 2 agents are present in the room up to almost 60% when 8 agents are used. As far as the heading uncertainty is concerned the relative accuracy improvement is lower (i.e. between 5% and 33%). Quite notably, in all cases the accuracy improvement grows as the grid of QR code become sparser. This results suggests that collaborative localisation can be particularly beneficial in areas that for some reasons are poorly instrumented with landmarks.
- To have a positioning uncertainty lower than 50 cm with a high probability, a fine-grained grid of QR codes is needed, regardless of whether the collaborative localisation is used or not. Nonetheless, collaborative localisation improves both accuracy and robustness, as expected.

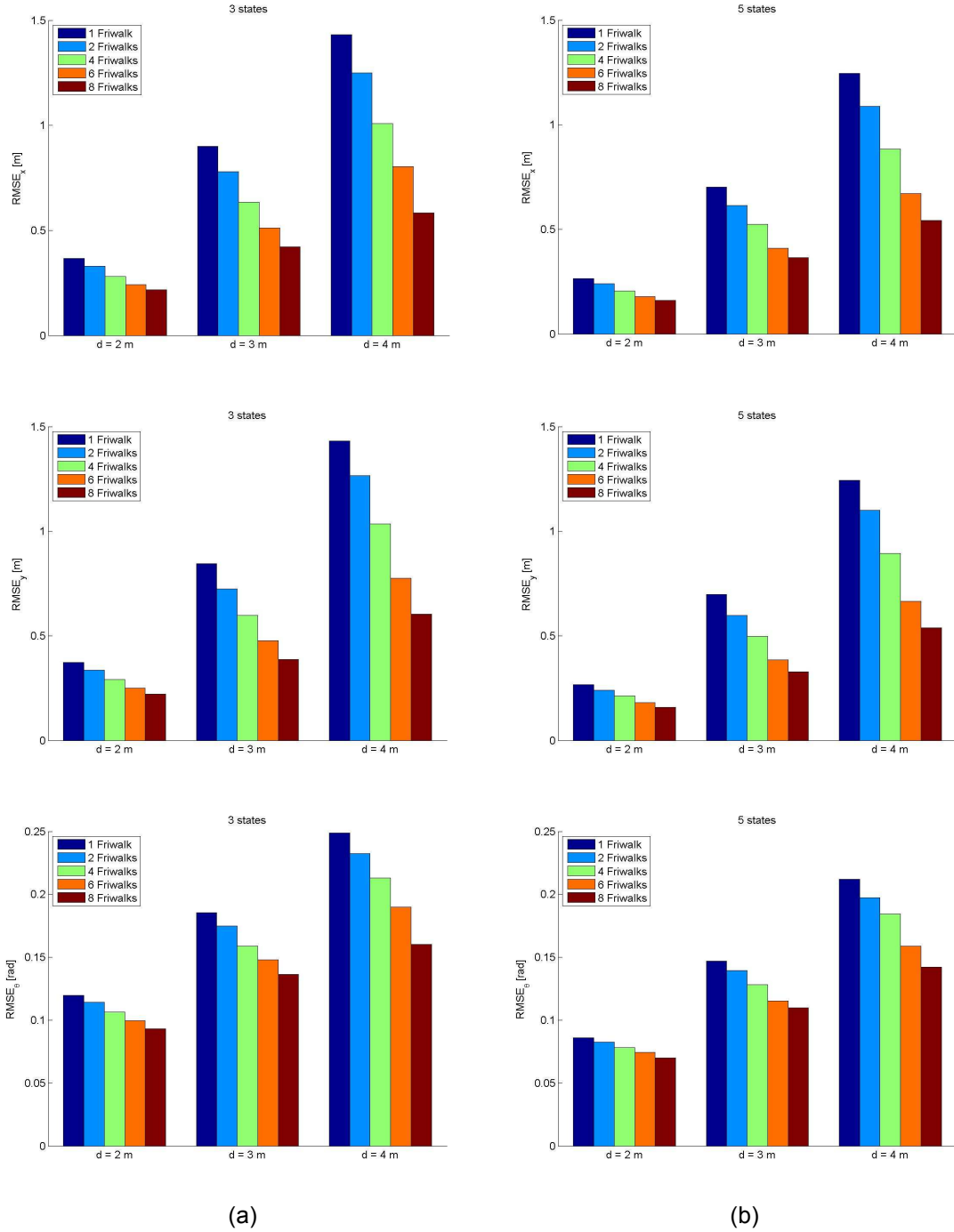


Figure 29: RMS estimation errors of the state variables x_k , y_k and θ_k computed over 200 trajectories when a different number of *FriWalks* (i.e. $N=1, 2, 4, 6, 8$) and triangle-patterned grids of QR codes with a different granularity (i.e. $d=2, 3$ and 4 m) are deployed in the environment. The bar diagrams on the left (a) and on the right (b) side refer to the 3-state variable and 5 state variable models, respectively.

5.2 Optimal QR code placement

Without using any kind of absolute position measurements, the position estimation uncertainty accumulates unboundedly over time. In this respect, collaborative localisation alone is not helpful since the state estimates performed by all agents would drift over time from their corresponding real values, thus leading to strongly biased results in a short time. This explains why the use of fixed landmarks is essential for correct system operation and convergence. On the other hand, the number landmarks in the environment should be as small as possible, while still providing reliable and accurate localisation. In this scenario, a major problem is where to

place the QR codes in such a way that the QR detection rate (and consequently localisation accuracy) is maximum, while using a minimal number of deployed landmarks.

In general, depending on the application and on the chosen sensing technology, landmarks can be detected continuously or intermittently. In the former case, the optimal landmark placement is the one for which, ideally, just one landmark is detected at every sampling time. In the latter case, the number of landmarks can be much smaller, thus reducing deployment complexity and costs. Of course, in both cases, the landmark placement problem strongly depends on the field of view (FoV) of the sensor in use. If the FoV is omnidirectional, the agent's direction does not influence the landmark placement problem and only its position should be taken into account [65]. In this case, to ensure that in an open wide room at least one landmark lies inside the FoV of the camera at every time, it is sufficient to place the landmarks on the vertices of an equilateral triangular pattern with a triangle side length equal to the sensor range [66].

Unfortunately, in the case of the *FriWalk*, the FoV of the front camera for QR code detection is not omnidirectional and it can be roughly approximated as a triangle with a limited angular range, as already shown in Figure 26. Moreover, the trajectories can be hardly defined a priori. As a result, the optimal placement problem is more complicated than in the omnidirectional case. Some heuristic solutions can be determined by solving the so-called “art gallery problem”. In such a solution, the environment is partitioned into regions which are completely coverable by the landmarks detection range [67]. However, this technique is useful mainly in the case of active landmarks, which is not the case of QR codes. This problem has been indeed addressed in [63] within the specific boundaries of the ACANTO project. In the following, a solution to the placement problem is formalized and a comparison between analytical and simulation based results are reported.

5.2.1 Problem formulation

Consider a polytope $\mathcal{P}$, representing the FoV of the front camera, namely an isosceles triangle with a vertex angle 2α , height h and length of either equal side $R = h/\cos(\alpha)$. If γ represents the orientation angle of the polytope with respect to axis X_W , as shown in Figure 30, $\mathcal{P}$ can be analytically defined as follows

$$\mathcal{P} \triangleq \left\{ \sum_{i=0}^2 \lambda_i q_i \mid \sum_{i=0}^2 \lambda_i q_i = 1, \lambda_i \in [0,1] \text{ for } i = 0,1,2 \right\}$$

where $q_0 = O_w = [0,0]^T$, $q_1 = r \begin{bmatrix} \sin(\beta + \gamma) \\ -\cos(\beta + \gamma) \end{bmatrix}$, $q_2 = r \begin{bmatrix} \sin(\beta - \gamma) \\ \cos(\beta - \gamma) \end{bmatrix}$ with $\beta = \pi/2 - \alpha$. If $p_{0,0} = [x_0, y_0]^T$ denotes a given lattice reference point (see Figure 30), the coordinates of any other point of the lattice can be expressed as

$$p_{i,j} = \begin{bmatrix} x_{i,j} \\ y_{i,j} \end{bmatrix} = p_{0,0} + \begin{bmatrix} j \frac{d}{2} + id \\ j h_d \end{bmatrix} \quad \forall i, j \in \mathbb{Z}$$

where d is the distance between landmarks and $h_d = d \cos(\alpha_d)$ is the height of equilateral triangles, with $\alpha_d = \pi/6$.