Effective Landmark Placement for Robot Indoor Localization with Position Uncertainty Constraints

Valerio Magnago[1], *Student Member, IEEE*, Luigi Palopoli[1], Roberto Passerone[1], *Member, IEEE*, Daniele Fontanelli[2], *Member, IEEE*, David Macii[2], *Senior Member, IEEE*

[1]Dep. of Information Engineering and Computer Science, University of Trento, Trento, Italy
E–mail: {valerio.magnago,luigi.palopoli,roberto.passerone}@unitn.it

[2]Dep. of Industrial Engineering, University of Trento, Trento, Italy
E–mail: {david.macii, daniele.fontanelli}@unitn.it

***Abstract*—A well-known, crucial problem for indoor positioning of mobile agents (e.g., robots) equipped with exteroceptive sensors is related to the need to deploy reference landmarks in a given environment. Normally, anytime a landmark is detected, an agent estimates its own location and attitude with respect to landmark position and/or orientation in the chosen reference frame. When instead no landmark is recognized, other sensors (e.g., odometers in the case of wheeled robots) can be used to track the agent position and orientation from the last detected landmark. At the moment, landmark placement is usually based just on common-sense criteria, which are not formalized properly.**

As a result, positioning uncertainty tends to grow unpredictably. On the contrary, the purpose of this paper is to minimize the number of landmarks, while ensuring that localization uncertainty is kept within wanted boundaries. The developed approach relies on the following key features: a dynamic model describing agents' motion, a model predicting the agents' paths within a given environment and, finally, a Conjunctive Normal Form (CNF) formalization of the optimization problem, which can be efficiently (although approximately) solved by a greedy algorithm. The effectiveness of the proposed landmark placement technique is first demonstrated through simulations in a variety of conditions and then it is validated through experiments on the field, by using non-Bayesian and Bayesian position tracking algorithms.

***Keywords*—Indoor navigation, optimization, greedy algorithms, performance evaluation, measurement uncertainty, service robots.**

I. Introduction

Indoor localization and position tracking systems rely on a variety of sensing technologies including (but not limited to) fingerprinting-based techniques based on radio signal strength intensity (RSSI) measurement [1], [2], electronic circuits measuring the Time-of-Flight of wireless signals [3], [4], inertial platforms [5], [6], [7], calibrated vision systems [8], [9], or a combination thereof [10], [11], [12]. While sensing technologies and accuracy specifications depend on the target application or the type of agents to be tracked (e.g., pedestrians or robots) [13], [14], common general requirements for indoor localization are:

- Robustness and continuous position tracking in indoor scenarios where the signals from Global Navigation Satellite Systems (GNSS) can be hardly detected;
- Positioning uncertainty in the order of a few tens of cm;
- Good scalability as the number of agents in the same environment grows.

At the moment, a one-size-fits-all solution able to meet all the basic requirements listed above does not exist. Reasonable trade-offs can be achieved by fusing multiple proprioceptive and exteroceptive sensor data. In particular, in the case of service robotics, the data from proprioceptive sensors such as odometers (for wheeled robots) or accelerometers and gyroscopes on board of Inertial Measurement Units (IMUs) can be combined with the distance and/or heading values measured through wireless, optical, ultrasonic or vision systems with respect to fixed devices (e.g., wireless anchor nodes, Radio-Frequency Identification Tags, or visual landmarks) placed at known locations in a given reference frame [15], [16], [17]. Unfortunately, the need to deploy such devices poses a crucial placement problem. Active devices (e.g., wireless nodes) can be detected from a longer distance (in the order of tens of meters), which drastically reduces the amount of devices to deploy. This is particularly important if the cost per unit is not negligible. However, ranging accuracy is typically quite low and tends to decrease with distance. Moreover, it further degrades in Non-Line-Of-Sight (NLOS) conditions. In addition, system scalability is limited by the number of agents that can communicate with the same wireless nodes at the same time.

On the other hand, passive tags or landmarks are much cheaper, but can be detected only when they are within a few meters from an agent. As a result, the system becomes inherently scalable (since fixed nodes and moving agents do not need to communicate, no congestion issues arise). However, the price to pay in this case is the need to deploy and to maintain a massive infrastructure.

To make the use of passive tags or landmarks feasible, it is essential to minimize their number. Unfortunately, this general optimization problem is NP-complete, as it depends on a variety of parameters such as detection range, sensor accuracy, agents trajectories and other environment-specific constraints (e.g., room geometry and possible obstacles). In fact, most of existing solutions are heuristic, i.e., based on common-sense approaches that strongly depend on the features of the specific

doi: 10.1109/TIM.2018.2887071

setup considered [18]. The problem of visual landmark selection has been widely investigated in the scientific literature. However, to the best of Authors' knowledge, most existing approaches can be hardly compared with the one described in this paper. This is due to a key conceptual difference. The classic landmark selection techniques attempt to decompose the environment into a minimal number of maximally sized regions, such that a minimum set of landmarks is visible from any position of a given region, thus ensuring continuous robot localization [19]. Therefore, assuming that different classes of landmarks exist, a mobile agent should see just one landmark of a given class at any time [20], whereas unnecessary landmarks are discarded a priori or ignored online, e.g., to reduce the computational burden required for localization [21], [22].

The placement technique described in this paper instead does not require that landmarks are always in view. In fact, even if it exploits, as a starting point, the purely geometric optimal placement criterion presented in [23] (which holds indeed only when at least one landmark is supposed to be detected at any time), it relaxes this requirements as it relies on the assumption that an agent can track its own position through dead reckoning even when landmarks are not detected for a while.

A further major difference between the proposed technique and others reported in the scientific literature is that agent positioning uncertainty is set as a constraint for landmark placement optimization and, consequently, it is kept under control a priori. This idea stems from the purely Monte-Carlo-based analysis applied in [16] and it is quite uncommon for landmark placement. In fact, most techniques are just focused on optimal coverage, while positioning uncertainty is evaluated only a posteriori [24]. Moreover, even when positioning uncertainty is taken into account, generally just a single trajectory from a continuous set of solutions is used to drive landmark placement [25], [26]. On the contrary, the technique described in this paper relies on a model able to predict the possible agent paths in a given environment, thus ensuring that the uncertainty constraint is met with a high level of confidence.

Even though the present paper is based on the preliminary work described in [27], [28], it relies on a more effective problem formulation, a novel path planner algorithm and a proper sensor characterization. Moreover, in this case the effectiveness of the proposed approach is validated, not only through simulations, but also through experiments performed in a real scenario. Indeed, the adopted greedy solver returns a nearly optimal solution within a reasonable time even when the indoor environment considered is particularly large [29].

The rest of the paper is structured as follows. Section II deals with the mathematical models and the case study presented, i.e., the *FriWalk* developed in the EU project ACANTO [30]. The landmark placement problem is introduced and properly formalized in Section III. Section IV describes approach to solve the problem. Section V summarizes the main metrological features of the sensors installed on the *FriWalk*. Afterwards, the model parameter values (estimated as described in Section V) are used to generate the possible agent paths and to run the landmark placement algorithm in a real indoor environment, i.e., a building of the University of Trento. The corresponding placement results obtained through simulations in different conditions are reported in Section VI. The effectiveness of the placement strategy is finally validated experimentally in Section VII. Section VIII concludes the paper.

II. Models Overview

This section describes the dynamic models used to formalize the landmark placement optimization problem presented in Section III. In particular, two kinds of kinematic models are considered, i.e., first a very general robot model and then a more specific model belonging to the same class, but tailored to better describe the dynamic of the *FriWalk*. The general model in Section II-A emphasizes the fact that the optimal landmark placement technique can be applied to a broad class of drift–less, input–affine wheeled robots used in indoor environments. Indeed, the only underlying assumptions are: the presence of a landmark detector with a limited Sensor Detection Area (SDA) and a dead reckoning position tracking system. The specific model described in Section II-B is instead just a special case of the general one and it is needed to validate the proposed approach in a practical case study, as shown in Section VII. For the sake of generality, in the following we assume that the location and attitude data measured anytime a landmark is detected agent position, i.e., without relying on the fusion with data collected from other sensors (e.g., odometers). The corresponding uncertainty analysis is described in Section II-C. Under these conditions, the landmark placement results are indeed expected to be more conservative than the results obtained when a Bayesian filter (e.g., an Extended Kalman Filter – EKF) is used. In fact, if a given selection of landmarks is able to keep positioning uncertainty below given boundaries using only raw sensor data, it is reasonable to assume that the same constraints can be even more safely met when some data fusion algorithm is used, as it will be shown in Section VII.

A. General model

The fixed, right-handed reference frame for platform localization is referred to as $\langle W \rangle = \{O_w, X_w, Y_w, Z_w\}$ and it is shown in Fig. 1. The robotic vehicle is regarded as a rigid body $\mathcal{B}$ moving in the plane $X_w \times Y_w$. If t_s denotes the sampling period common to all onboard sensors, the generalized coordinates of the robot at time kt_s are $p_k = [x_k, y_k, \theta_k]^T$, where (x_k, y_k) are the coordinates of the origin of frame $\langle \mathcal{B} \rangle = \{O_b, X_b, Y_b, Z_b\}$ attached to the rigid body, while θ_k represents the angle between X_b and X_w. The kinematic model of a generic drift–less, input–affine wheeled robot can be represented by the following discrete–time system, i.e.,

$$\begin{cases} p_{k+1} = p_k + G_k(p_k, q_k + \epsilon_k) \\ z_k = h(p_k) + \eta_k \end{cases} \tag{1}$$

where q_k is the piecewise input vector of the system between $(k-1)t_s$ and kt_s, ϵ_k is the additive zero-mean uncertainty term affecting input quantities and $G_k(\cdot)$ is the input vector function. Furthermore, z_k (namely the vector of output quantities that can be observed at time kt_s) is given by the sum

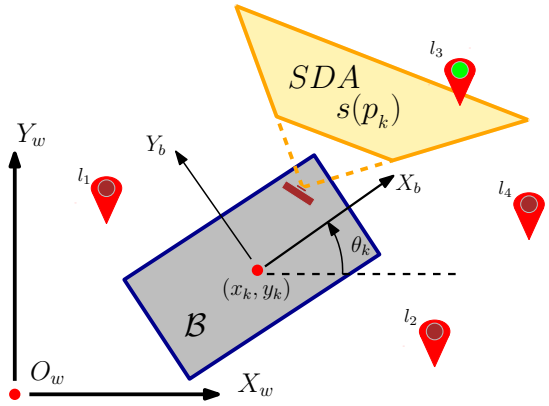


Fig. 1. Generic representation of a robot to be localized in reference frame $\langle W \rangle$. Landmarks l_1, l_2, l_3 and l_4 are also represented. In particular, l_3 lies inside the SDA of the landmark detection sensor when the robot is located in $s(p_k)$.

of $h(p_k)$ (i.e., a generic nonlinear output function of the state) and η_k, which represents the vector of zero-mean uncertainty contributions when output quantities are measured. If the agent position is estimated through dead reckoning, the accumulation of random contributions ϵ_k unavoidably leads to large position and orientation uncertainty after a while. If instead the robot detects, at least sporadically, some artificial landmarks placed at known positions in $\langle W \rangle$, the positioning uncertainty is kept bounded. Consider that, in general, the SDA (denoted as $s(p_k)$ in Fig. 1) of any landmark detector exhibits a finite range and a limited angular aperture. However, both detection range and angular aperture may depend on robot position p_k.

B. A more specific model: the FriWalk case

The *FriWalk* is equipped with relative encoders on the rear wheels and with a front monocular camera used to detect specific landmarks (i.e., Aruco codes) placed at known positions in $\langle W \rangle$ (e.g., on the floor) and with a given orientation with respect to X_w [12]. The kinematic model of the *FriWalk* is a unicycle [31]. In this case, the robot planar coordinates (x_k, y_k) (namely, the origin of the body frame O_b with axis X_b pointing forward) refer to the mid-point of the rear axle (see Fig. 2). Observe that, with reference to Fig. 1, the robot generalized coordinates are still $p_k = [x_k, y_k, \theta_k]^T$. The camera measures the relative position and orientation of the robot with respect to every detected Aruco code. Absolute position and orientation in $\langle W \rangle$ are then estimated as described in [12]. The main parameters of the SDA (which in this case coincides with the field of view of a front camera) are: the maximum and minimum detection ranges (denoted as R and r, respectively) and the camera aperture angle α, as shown in Fig. 2. It is worth noting that unlike the preliminary study reported in [27], the SDA exhibits a trapezoidal and not a triangular shape, because landmarks excessively close to the camera are certainly out of its field of view.

With reference to the general model described by expression (1), in the case of the *FriWalk* the input vector function

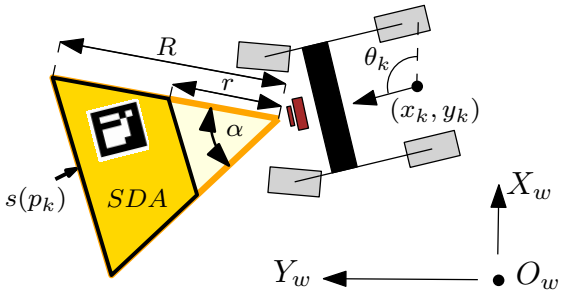


Fig. 2. Geometrical parameters of the *FriWalk* model and of the SDA of the landmark detection sensor.

of the system is

$$G_k(p_k, q_k + \epsilon_k) = \begin{bmatrix} (v_k + \epsilon_{v_k}) t_s \cos\theta_k \\ (v_k + \epsilon_{v_k}) t_s \sin\theta_k \\ (\omega_k + \epsilon_{\omega_k}) t_s \end{bmatrix}. \quad (2)$$

where input vector $q_k = [v_k, \omega_k]^T$ includes the angular and linear velocities of the robot (denoted as ω_k and v_k, respectively) at time t_k. The additive input noise $\epsilon_k = [\epsilon_{v_k}, \epsilon_{\omega_k}]^T$ due to the finite resolution and tick reading errors of both wheels encoders can be reasonably assumed to be white and normally distributed. As a consequence, the covariance matrix of the noise associated with v_k and ω_k is

$$E = \begin{bmatrix} \sigma_v^2 & \sigma_{v\omega} \\ \sigma_{v\omega} & \sigma_\omega^2 \end{bmatrix}. \quad (3)$$

where σ_v^2 and σ_ω^2 represent the variances of v_k and ω_k, respectively, while $\sigma_{v\omega}$ is the covariance between them. Finally, the output function $h(p_k)$ of system (1) just coincides with the state of the system itself, i.e. $h(p_k) = p_k$. Therefore, $z_k = p_k + \eta_k$, where η_k is the vector of uncertainty contributions associated with the measurement of position and orientation based on landmark detection (see Section V for further details). In particular, the covariance matrix of η_k is

$$N = \begin{bmatrix} \sigma_{c_x}^2 & \sigma_{c_{xy}} & \sigma_{c_{x\theta}} \\ \sigma_{c_{xy}} & \sigma_{c_y}^2 & \sigma_{c_{y\theta}} \\ \sigma_{c_{x\theta}} & \sigma_{c_{y\theta}} & \sigma_{c_\theta}^2 \end{bmatrix} \quad (4)$$

where $\sigma_{c_x}^2$, $\sigma_{c_y}^2$ are the variances associated with the camera-based measurements of the robot planar position along axis X_w and Y_w, $\sigma_{c_\theta}^2$ is the variance of the orientation measurements with respect to X_w, and terms $\sigma_{c_{xy}}$, $\sigma_{c_{x\theta}}$ and $\sigma_{c_{y\theta}}$ represent the covariances between pairs of measured quantities.

C. Uncertainty Analysis

If a non-Bayesian estimator is used and one landmark is detected at time kt_s, the covariance matrix $P_k \in \mathbb{R}^{3\times3}$ of the state estimation error simply coincides with (4), i.e., $P_k = N$. In such conditions, the positioning uncertainty depends on the metrological features of the vision system used to measure the relative position and orientation of the robot with respect to the landmark lying in the SDA of the camera. However, when no landmarks are detected, positioning uncertainty tends to

grow due to the accumulation of the noise introduced by dead reckoning (e.g., due to the wheels encoders used for odometry, as explained in Section II-B). In this case, the evolution of P_k as a function time can be obtained, to a first approximation, from the linearization of the state equation of system (1) around the estimated state. Thus, assuming that ϵ_k and p_k are uncorrelated $\forall k$, it follows that

$$P_{k+1} \approx \left(I + \frac{\partial G_k(p_k, q_k)}{\partial p_k}\right) P_k \left(I + \frac{\partial G_k(p_k, q_k)}{\partial p_k}\right)^T + \frac{\partial G_k(p_k, \epsilon_k)}{\partial \epsilon_k} E \frac{\partial G_k(p_k, \epsilon_k)}{\partial \epsilon_k}^T. \quad (5)$$

Expression (5) can be regarded as an application of the law of propagation of uncertainty in the multivariate case [32]. Moreover, assuming that the initial state of the system is known, it is reasonable to set $P_0 = N$.

Consider that, since P_k is a 3×3 matrix, a scalar uncertainty parameter is preferable to monitor and to keep positioning uncertainty under control. Therefore, in the rest of this paper the following function will be used to evaluate positioning uncertainty, i.e.,

$$u_p(P_k) = \sqrt{\max \mathrm{Eig}\left(P_k^{x,y}\right)}, \quad (6)$$

where $P_k^{x,y}$ refers to the upper 2×2 matrix of P_k, i.e.,

$$P_k^{x,y} = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix}, \quad (7)$$

and operator $\mathrm{Eig}(\cdot)$ returns the eigenvalues of the argument matrix.

The rationale for choosing function (6) to set uncertainty constraints is threefold. First of all, it is simple to apply. Secondly, even if $u_p(\cdot)$ might include the orientation contribution, in practice just the uncertainty associated with planar position is typically of interest [25], [26]. Finally, the use of function (6) is conservative because, from the geometrical point of view, it can be regarded as the radius of a circle centered in the estimated position and circumscribing the ellipse representing the actual positioning uncertainty in the plane $X_w \times Y_w$. In particular, $u_p(P_k) \in [u_p^-, u_p^+]$, where $u_p^- = \max(\sigma_x, \sigma_y)$ if the correlation coefficient ρ in (7) is equal to 0, while $u_p^+ = \sqrt{\sigma_x^2 + \sigma_y^2}$ if $|\rho| = 1$.

Observe that, using a non-Bayesian estimator, the minimum positioning uncertainty is achieved anytime a landmark is detected. Therefore, $u_p(P_k) \geq u_p(N)\ \forall k$.

III. Problem Formulation

Let $\mathcal{P} \subseteq \mathbb{R}^2 \times [0, 2\pi)$ be the set of all configurations reachable by an agent inside the environment, so that $p_k \in \mathcal{P}$ $\forall k$. If $\mathcal{D}$ denotes the detectable area (namely the set of points lying in the SDA for at least one of the possible positions of the robot), i.e.

$$\mathcal{D} = \left\{(x, y) \in \mathbb{R}^2 \mid \exists p_k \in \mathcal{P}, (x, y) \in s(p_k)\right\},$$

then $\mathcal{L}_p \subseteq \mathcal{D}$ can be referred to as the set of points where landmarks can be actually deployed. Let $\xi(p_k)$ be the maximum wanted (or target) positioning uncertainty. Note that, in general, $\xi(p_k)$ can be a function of the current robot position (e.g., because locations close to walls require more accurate localization to avoid collisions). Observe also that $\mathcal{L}_p$ has an infinite cardinality. Therefore, to make the landmark placement problem tractable, a finite–element set $\mathcal{L}_f \subseteq \mathcal{L}_p$ should be defined, to ensure that the minimum possible target uncertainty is always achieved, i.e., $u_p(P_k) = u_p(N) \leq \xi(p_k)$, $\forall k$. This condition holds true if, in every position of the chosen environment, at least one landmark lies within the SDA, i.e., $\mathcal{L}_f \cap s(p_k) \neq \emptyset, \forall p_k \in \mathcal{P}$. Of course, the cardinality of set $\mathcal{L}_f$ (denoted with symbol $|\cdot|$ in the rest of this paper) should be as little as possible to minimize the search space of possible landmark positions. The resulting minimization problem can be formulated as follows:

Problem 1: Given $\mathcal{P}$ and $s(\cdot)$, find

$$\mathcal{L}_f = \arg\min_{\mathcal{L}_x} |\mathcal{L}_x| \text{ s.t.} \quad \forall p_k \in \mathcal{P},\ \mathcal{L}_x \cap s(p_k) \neq \emptyset \wedge \mathcal{L}_x \subseteq \mathcal{L}_p.$$

A geometry-based closed-form optimal solution to Problem 1 is reported in [23]. The set $\mathcal{L}_f$ thus obtained is indeed the starting point for the placement optimization problem addressed in this paper.

In this respect, to refine the search for optimal solutions, some knowledge of the possible paths followed by the agents is essential. An obvious constraint to ensure observability is that at least one landmark must be detected along every path. If fully autonomous vehicles are considered, usually the set of possible paths has a finite cardinality and it is well defined. If human beings are involved instead (like in the case of the *FriWalk*), the set of possible trajectories is infinite, but the regions of space that are explored with highest probability (i.e., the most likely paths) can be derived statistically from empirical observations [33], [34].

Even if a path $T_i \in \mathcal{T}$ (where $\mathcal{T}$ is the set of all available paths) ideally consists of an infinite number of points, in practice it can be discretized by using the elements of $\mathcal{L}_f$. Indeed, $\forall i, k,\ \exists p_k \in T_i : \mathcal{S}_{i,k} = s(p_k) \cap \mathcal{L}_f \neq \emptyset$. Of course, the mapping between $p_k \in T_i$ and $\mathcal{S}_{i,k}$ is *not* bijective since multiple landmarks can be potentially detected by the same robot. Thus, the landmark placement optimization problem addressed in the next section can be formalized as follows, i.e.,

Problem 2: Given $\mathcal{P}$, $\mathcal{L}_f$, $\mathcal{T}$ and $\xi(p_k) \geq u_p(N), \forall p_k \in \mathcal{P}$, find:

$$\mathcal{L} = \arg\min_{\mathcal{L}_x} |\mathcal{L}_x| \text{ s.t.} \quad \mathcal{L}_x \subseteq \mathcal{L}_f, \quad \forall i\ T_i \in \mathcal{T}, \forall k\ p_k \in T_i,\ u_p(P_k) \leq \xi(p_k).$$

Observe that $\mathcal{L} \subseteq \mathcal{L}_f$. Therefore, the problem is well-posed since at least one solution (i.e., $\mathcal{L}_f$) certainly exists.

IV. Optimal Landmark Placement

A variety of strategies can be used to solve Problem 2. In this section, first the minimization problem is represented in a Conjunctive Normal Form (CNF) [27]; then an exact, but computationally intensive approach, and a faster, heuristic method for its solution are described.

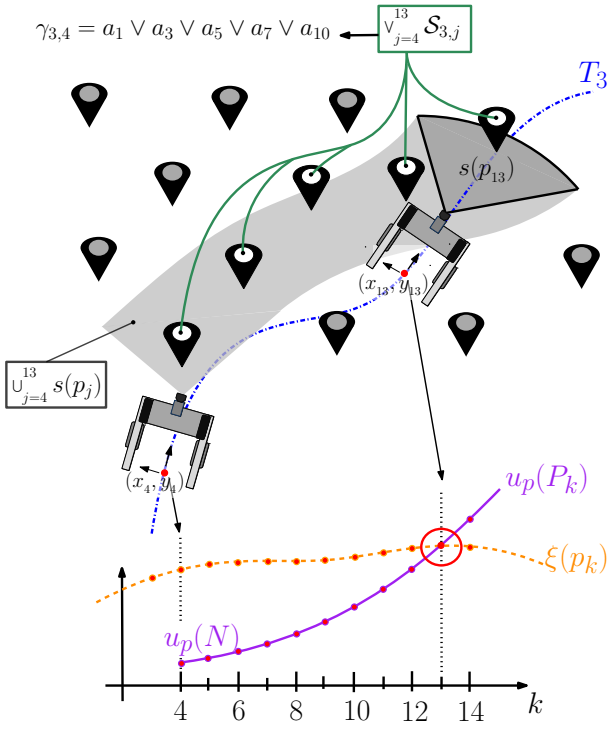


Fig. 3. Example of uncertainty growth along a sample trajectory. The dashed line represents the uncertainty threshold $\xi(p_k)$, while the solid line is the uncertainty growth when no landmark is in the SDA starting with minimum uncertainty N at step $k = 4$. At step $k = 14$ we have $u_p(P_{14}) > \xi(p_{14})$, therefore there is a violation. On the upper part of the figure, the shaded band represent the union of the areas observed by the camera from step 4 to 13. The placeholder icon stand for possible landmarks position and those belonging to the shaded band will generate the clause $\gamma_{3,4}$.

A. Conjunctive Normal Form (CNF) Problem Representation

To formalize the problem, a boolean variable a_i can be associated with each possible landmark location $l_i \in \mathcal{L}_f$ such that

$$a_i = \begin{cases} 1, & \text{if a landmark is placed in } l_i, \\ 0, & \text{otherwise.} \end{cases}$$

Thus, a landmark deployment corresponds to an assignment to the boolean variables. The objective is to find a *least* assignment, i.e., an assignment such that the minimum number of variables is assigned the value 1, which satisfies the uncertainty constraints. We model the constraints by identifying all the partial assignments to the variables that lead to a violation. Consider a position $p_s \in T_i$, and assume $u_p(P_s) = u_p(N)$, i.e., the minimum uncertainty in our setting. We simulate the trajectory and compute the evolution of $P_{s+1}, P_{s+2}, \ldots$ along T_i. At the same time, let $\mathcal{S}_{i,j}$ represent the landmark positions within the field of view of the landmark detector at time j along path i. If at time $k+1 > s$, $u_p(P_{k+1}) > \xi(p_{k+1})$, then we have a violation since $\xi(p_{k+1})$ is the maximum position uncertainty allowed at point p_{k+1} (see Fig. 3 at time $k+1 = 14$). In order to avoid it, at least one landmark has to lie in one of the positions $\cup_{j=s}^{k} \mathcal{S}_{i,j}$ in view. This condition can be expressed as

$$\gamma_{i,s} = \bigvee_{j=s}^{k} \mathcal{S}_{i,j},$$

where, with a slight abuse of notation, the boolean variables associated with the landmark positions are denoted with $\mathcal{S}_{i,j}$ (see Figure 3). In plain words, $\gamma_{i,s}$ evaluates to true if and only if at least one landmark lies in the SDA when the robot moves on path i from time s to time k, where k, in this case, is the instant immediately before the time when the position uncertainty constraint is violated. Clearly, a landmark deployment $\mathcal{L}$ that does not satisfy $\gamma_{i,s}$ cannot be a solution to Problem 2, since between p_k and p_{k+1} the uncertainty constraint would be violated. We can repeat this analysis for all starting positions and all trajectories, and collect the clauses in a set Γ. To find a solution to the problem, it is necessary and sufficient that all the generated clauses evaluate to true. Thus, the function

$$\varphi(a_i, \ldots, a_n) = \bigwedge \Gamma = \bigwedge_{i,s} \gamma_{i,s}$$

evaluates to true for all and only those assignments to the boolean variables $a_1, \ldots, a_n$ which correspond to a correct deployment. Given its form, φ is expressed in CNF.

B. Optimal Placement

To optimize the placement we need to find the least satisfying assignment, i.e., an assignment to the variables $a_1, \ldots, a_n$ such that φ is true and the least number of variables is assigned value 1. There are several ways to formally solve this problem. One approach is to cast it as a logic optimization problem [27]. Observe that the conjunction of the true variables of a satisfying assignment is an implicant of φ, that is, the product term "covers" some of the ones of φ. A *minimal* deployment corresponds to a *prime* implicant of φ. The *minimum* deployment is therefore the *largest* prime implicant. Logic optimization can then be used to find a minimum 2-level cover of φ. Each term of the resulting cover corresponds to a minimal deployment, and we choose the one with the least number of variables. This approach has the advantage that it provides several alternative solutions corresponding to the various terms of the cover. While this strategy gives us the best solution, the downside lies in its computational complexity, which is exponential in the number of variables and in the number of prime implicants [27].

There exist other possible and more efficient encodings of the boolean optimization problem. In fact, the choice of the set of locations corresponds to a minimal covering problem. In logic optimization, this is equivalent to the selection of a minimal cover, given the set of prime implicants of a boolean function. In other words, the solver may skip the search of the prime implicants, and only perform the selection, if the (prime) implicants (each corresponding to a sensor location covering a number of constraints) are provided ahead of time. This suggests a boolean function representation in which the role of the locations a_i and the constraints γ_j is reversed.

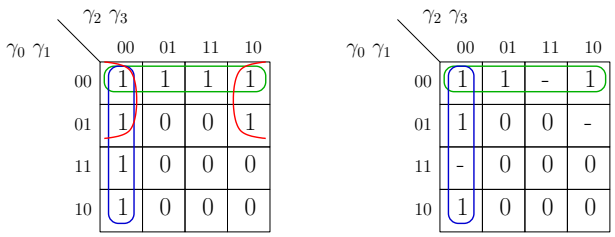


Fig. 4. Function ψ as a Karnaugh map. Left: full specification, showing the locations as prime implicants. Right: relaxed specification, with don't cares denoting simultaneous location coverage.

That is, the constraints become input variables to a function $\psi(\gamma_0, \ldots, \gamma_m)$, whose implicants are defined by the sensor locations. Logic optimization then selects the smallest set of prime implicants (locations) that still forms a cover. For this to work, we must ensure that the larger the location coverage, the larger the corresponding implicant. We therefore use the following encoding: a location, acting as an implicant, sets the value of ψ to 1 for all those input combinations for which all of the constraints which are *not* covered are assigned value 0. This approach can be illustrated through an example. Consider four constraints $\gamma_0, \gamma_1, \gamma_2, \gamma_3$, and three locations a_0, a_1, a_2 such that

$$\begin{aligned} a_0 \quad &\text{covers} \quad \gamma_0 \text{ and } \gamma_1, \\ a_1 \quad &\text{covers} \quad \gamma_2 \text{ and } \gamma_3, \\ a_2 \quad &\text{covers} \quad \gamma_1 \text{ and } \gamma_2. \end{aligned} \tag{8}$$

The minimum cover is clearly given by locations a_0 and a_1. We associate to each location a corresponding term:

$$\begin{aligned} a_0 \quad &\Rightarrow \quad \overline{\gamma_2} \cdot \overline{\gamma_3}, \\ a_1 \quad &\Rightarrow \quad \overline{\gamma_0} \cdot \overline{\gamma_1}, \\ a_2 \quad &\Rightarrow \quad \overline{\gamma_0} \cdot \overline{\gamma_3}. \end{aligned}$$

In this specific example, function $\psi = \overline{\gamma_2} \cdot \overline{\gamma_3} + \overline{\gamma_0} \cdot \overline{\gamma_1} + \overline{\gamma_0} \cdot \overline{\gamma_3}$ is shown in Fig. 4 (on the left) as a Karnaugh map. For any input assignment, ψ is 1 if the constraints which have value 1 in the assignment are covered by the same location. Effectively, logic minimization would need to include *all* locations a_0, a_1 and a_2 to cover the function (as shown on the left side of Fig. 4). Observe that the requirements in the map are too strict and the resulting coverage too conservative, since setting $a_0 = 1$ and $a_1 = 1$ would be sufficient. The key observation here is that we do not need the constraints to be covered simultaneously by the same location. For instance, in the cell $\gamma_0\gamma_1 = 00$ and $\gamma_2\gamma_3 = 11$ of the Karnaugh map, we require that γ_2 and γ_3 be covered by the same location, while other cells in the map ensure that they will be covered individually. At the same time, simultaneous coverage should not be ruled out, as much as it helps with reaching a minimum cover. To reconcile these two requirements, we replace the 1's of ψ corresponding to simultaneous coverage with *don't care* conditions, as shown in Figure 4, right. In this way, logic minimization will not be required to choose certain implicants, nor will it be prevented from doing so. The selection therefore leads to the minimum cover.

TABLE I. COVERAGE MATRIX EXPRESSING THE CLAUSE AS DISJUNCTION OF BOOLEAN VARIABLES: $\gamma_{2,3} = a_1 \vee a_2 \vee a_8 \vee a_9$; $\gamma_{4,1} = a_2 \vee a_3 \vee a_6$; $\gamma_{3,2} = a_2 \vee a_4$; $\gamma_{3,4} = a_1 \vee a_3 \vee a_5 \vee a_7 \vee a_{10}$; $\gamma_{3,5} = a_3 \vee a_5 \vee a_7 \vee a_{10}$.

	1	2	3	4	5	6	7	8	9	10
$\gamma_{2,3}$	1	1	0	0	0	0	0	1	1	0
$\gamma_{4,1}$	0	1	1	0	0	1	0	0	0	0
$\gamma_{3,2}$	0	1	0	1	0	0	0	0	0	0
$\gamma_{3,4}$	1	0	1	0	1	0	1	0	0	1
$\gamma_{3,5}$	0	0	1	0	1	0	1	0	0	1

This encoding results in much better performance. Using the SIS optimization software on an Intel i7-6700 CPU PC running at 3.50 GHz with 8 GB of RAM [35], a small problem with 15 constraints and 52 locations is solved in over 3 minutes using the first encoding, while it takes negligible time with the second encoding. A 21-constraint problem with 65 locations, which takes days with the first method, is instead solved in less than a second. While the performance improvement of five orders of magnitude allows the solver to address large problems, the exponential complexity can still be a limiting factor when the number of constraints becomes very large.

Alternatively, the problem can be rephrased as a constrained boolean optimization, i.e.,

$$\min \sum_i a_i, \quad \text{subject to} \quad \forall i, \forall s, \quad \gamma_{i,s} > 0$$

Even if the computational complexity of the problem is still exponential, one can solve the continuous relaxation of the same problem, which is polynomial. Of course, since in this case the variables may take any value between 0 and 1, the solution of the problem in general will be infeasible, although it can be used as a lower bound to estimate the optimality of heuristic solutions. In particular, we rely on a greedy approach [27], based on the greedy heuristic for sub-modular functions [36], which leads to a good approximation of the optimal solution within a negligible computation time. We start with a compact representation of Γ given by a coverage matrix. The matrix columns refer to the possible landmarks locations $l_i \in \mathcal{L}_f$, whereas the matrix rows represent the clauses $\gamma_{i,s}$. The matrix element in (r, c) is 1 if the r-th clause is satisfied by the c-th landmark, or 0 otherwise. An example is shown in Table I. To optimize the coverage, the columns are ordered according to the number of elements equal to 1, in a decreasing fashion. With reference to Table I, the first column would be l_2, then l_3 and so on. A landmark is placed in the position corresponding to the first column, i.e., the one satisfying the greatest number of clauses. The corresponding satisfied clauses (the matrix rows) are then removed from the matrix, together with the first column, and the matrix is reordered. With reference to Table I, l_2 is added to $\mathcal{L}_g$ and the first three rows are removed. A new matrix A_1 is obtained, and the procedure starts over. The procedure ends when there are no more clauses to meet. For the case of Table I, the procedure may end with $\mathcal{L}_g = \{l_2, l_5\}$ or with $\mathcal{L}_g = \{l_2, l_3\}$, namely when at most two landmarks are placed. As shown in Section VI, despite its simplicity, the greedy solution $\mathcal{L}_g$ turns out to be very effective when compared to the (infeasible)

Fig. 5. The *FriWalk* developed within the European project ACANTO.

lower bound given by the relaxation solution [27], and can handle problems of large size.

V. Experimental Setup

As mentioned in Sections I and II, the platform used to validate experimentally the proposed approach is the *FriWalk* (Fig. 5), a commercial trolley for seniors[1] endowed with sensors [16], [12], as well as processing [37], [31] and guidance functions [38], [39], [40]. The robot can estimate its own speed, enabling odometric trajectory estimation, through two encoders AMT-102V mounted on rear wheels with a resolution of 0.08 mrad per tick. In addition, the relative pose of the camera with respect to the Aruco code detected in the camera field of view (namely the SDA in the case at hand) can be measured by using a front RGB camera (PLAYSTATION Eye) and software application based on OpenCv 3.1.0.

Encoder information is collected by a BeagleBone Black board via a Controller Area Network (CAN) bus. The BeagleBone Black board processes encoder data and sends odometry results to an Intel NUC mini PC (equipped with a microprocessor i7-5557 and 8 GB of DDR3 RAM) through Local Area Network (LAN) router. The PLAYSTATION Eye is connected directly to the Intel NUC mini PC through a USB link. The NUC mini PC is in turn also connected to the LAN router. The router provides Wi-Fi connectivity between the *FriWalk* and an external PC used for telemetry, e.g. to log the encoder measurement data and the relative position and orientation measures with respect to every detected Aruco code while the robot is moving. Accuracy and precision of the linear and angular velocity estimates v_k and ω_k based on odometry were evaluated by comparing the values returned by the BeagleBone Black board with those obtained by differentiating the position and orientation measured by an OptiTrack reference localization system. In all experiments, *FriWalk* and

[1]Trionic Walker 12er

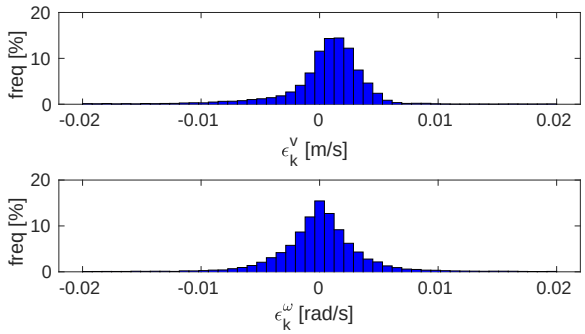


Fig. 6. Experimental distribution of the linear velocity estimation error ϵ_k^v and of the angular velocity estimation error ϵ_k^ω due to encoder data.

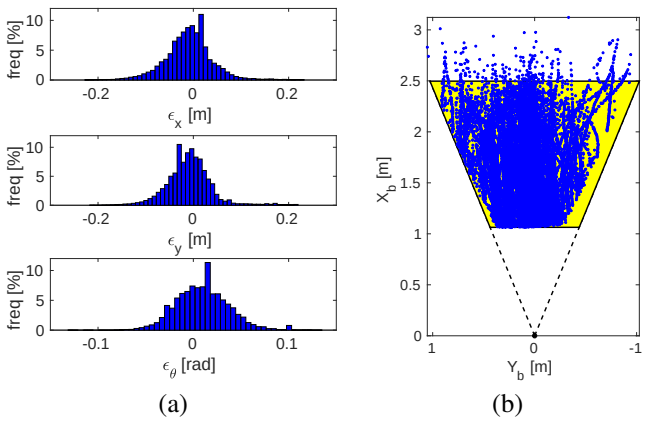
TABLE II. Parameter values for *FriWalk*localization.

Symbol	Value	Unit
$\xi(p_k)$	0.8	m
r	1.15	m
α	0.78	rad
$\sigma^2_{c_x}$	34	cm^2
$\sigma^2_{c_\theta}$	0.001	rad^2
$\sigma_{c_{x\theta}}$	-1	mm·mrad
$\sigma_{c_{y\theta}}$	-8	mm·mrad

Symbol	Value	Unit
t_s	0.2	s
R	2.70	m
$\sigma^2_{c_y}$	30	cm^2
$\sigma_{c_{xy}}$	-2	mm^2
σ^2_v	20	mm^2/s^2
σ^2_ω	20	mrad^2/s^2
$\sigma_{v\omega}$	2	mrad·mm/s^2

OptiTrack data were properly aligned in time. Moreover, the robot was driven repeatedly (i.e. about 50 times) and at a different speed (ranging from 0.3 m/s to 1.2 m/s) over an eight-shaped path. The OptiTrack localization system consists of 14 calibrated cameras and it able to measure the position of ad-hoc reflective markers attached to the *FriWalk* with standard uncertainty of about 1 mm, i.e. negligible compared with the positioning uncertainty based on odometry. The histograms of the differences ϵ_k^v and ϵ_k^ω between the linear and angular velocity data, respectively, resulting from odometry and the OptiTrack-based localization system are shown in Fig. 6. Observe that the mean values of ϵ_k^v and ϵ_k^ω are negligible, while the elements of the covariance matrix E are reported in Tab. II. Such values, although apparently small, have a significant impact on odometry-based positioning uncertainty since they tend to accumulate over time due to dead reckoning.

The OptiTrack reference localization system was also used to evaluate accuracy and precision of distance and orientation measurements based on the PLAYSTATION Eye, whenever an Aruco code is detected. Again, the *FriWalk* was driven repeatedly over an eight-shaped path. The histograms of the differences between the position and orientation values measured by the on-board vision system and those obtained with the OptiTrack are shown in Figure 7(a). Such differences are realizations of the components of the random vector η_k in (1), whose covariance matrix is (4). Observe that the mean values of the elements of η_k are -7.8 mm, -8.3 mm and 10 mrad, respectively, and can be easily compensated, thus obtaining a zero mean process, as assumed in Section II-A. The corresponding standard uncertainty values (i.e., about 58 mm,



(a) (b)

Fig. 7. Estimation of the camera parameters. (a) Error histograms of the camera reading pose. (b) Trapezoidal approximation of the SDA: the dots represent the relative measured positions of the Aruco codes with respect to the camera, while the shaded trapezoid is the estimated SDA.

54 mm and 34 mrad) are considered adequate for the intended application. The positions of the landmarks detected in repeated trials (represented with about 20000 dots in Figure 7(b)) was also used to estimate the SDA of the PLAYSTATION Eye installed on the *FriWalk*. In particular, the SDA exhibits approximately a trapezoidal shape and the values of parameters r, R and α shown in Fig. 2 are summarized in Tab. II. The same Table reports also the values of the elements of covariance matrices E and N, the sampling period t_s and the target uncertainty $\xi(p_k)$ for landmark placement. For the sake of simplicity (but without loss of generality) in the rest of this paper $\xi(p_k)$ is assumed to be constant, i.e. equal to 0.8 m regardless of actual *FriWalk* position. This value is just an example, but it is reasonable for the purposes of project ACANTO.

VI. Placement implementation

The indoor scenario chosen to validate the proposed placement strategy is the Department of Information Engineering and Computer Science (DISI) of the University of Trento. Given the map of the environment, the placement technique described in Section IV requires to know the possible agent paths. Unfortunately, the trajectories generated using a planner based on elastic bounds, as proposed in [27], does not guarantee that a generated path is likely to happen in practice. In [42] instead, 90% out of 1560 human trajectories were generated with accuracy better than 10 cm, by using paths consisting of arcs of a clothoid. In [37], [41] it is shown that the clothoid-based model is able to describe the natural behavior of the *FriWalk* even in crowded environments, i.e., when the presence of other robots and human beings may affect the path of a moving agent [43]. Therefore, in this paper, the same approach is adopted to generate a set of 2085 possible paths, as shown in Fig. 8.

Consider that path regularity increases the number of shared boolean variables between clauses. This situation makes the solution based on a greedy approach quite challenging. For given values of R, r and α (see Tab. II), the Aruco code potential locations can be determined by applying the geometrical criterion described in [23]. Their total number in the DISI premises amounts to $|\mathcal{L}_f| = 9420$. Such positions are represented by cross-shaped markers in the inlet of Fig. 8. Along the 2085 generated paths, it was verified through simulations that 8685 out of 9420 possible landmarks lie in the SDA of *FriWalk* vision system at least once. The number of derived clauses, assuming a maximum target uncertainty $\xi(p_k) = 0.8$ m, is 38947. By solving the relaxed optimization problem described in Section IV-B, the resulting optimal number of landmarks, computed in 90 minutes, is 20.63. Even though this number corresponds to an infeasible solution (the amount of landmarks of course cannot be fractional), it can be regarded as a lower bound to optimal placement [27]. To obtain a feasible solution from the relaxed one, we can select incrementally the landmark positions with the highest value (i.e., the locations whose value is closer to 1) until all of the clauses are satisfied. A total of 67 landmarks can be placed in this way. Conversely, the greedy algorithm selects only $|\mathcal{L}_g| = 35$ Aruco codes (represented by white circles in Fig. 8). Notice that most of the selected Aruco codes are located in the corridors of the building. This is reasonable, as path density is obviously higher than in offices.

It is worth noting that, even if the number of paths and potential landmark locations is quite large, the computation time of the greedy algorithm implemented in Matlab and running on a PC provided wit a 3.50 GHz Intel Core i7 microprocessor and 8 GB RAM is about 55 minutes. Therefore, the greedy algorithm is computationally more efficient than the relaxed linear programming optimization problem, and it returns a solution that is reasonably close to the infeasible lower bound.

A further benefit of the proposed placement technique is that the uncertainty constraint $\xi(p_k)$ does not need to be constant all over the environment considered. For instance, positioning uncertainty has to be lower in rooms cluttered with objects or including forbidden areas, whereas it can be larger in the case of wide open environments. Fig. 9 shows the result of an alternative landmark placement when different uncertainty constraints $\xi(p_k)$ are used, i.e., 0.2 m (halls next to staircases), 0.4 m (narrow vertical corridors), 0.7 m (wide vertical corridors), 1.0 m (long horizontal corridors) and 2.0 m (other rooms). Observe that, in this case, landmark positions are quite different from those shown in Fig. 8 even if the computation time is approximately the same. Moreover, the number of landmarks selected by the greedy algorithm is $|\mathcal{L}_g| = 57$, with a higher density where the maximum target uncertainty is lower, as expected. This example confirms the flexibility of the proposed placement strategy.

To evaluate to what extent the greedy solution is better than other naive landmark placement strategies, a comparison with several random layouts is reported. The box-and-whiskers plot in Fig. 10 shows the percentages of paths satisfying the uncertainty constraint $\xi(p_k) = 0.8$ m as a function of average Aruco code density, assuming that between 0.1% and 3.1% of all possible landmarks are selected randomly with the same probability. Each box of Fig. 10 refers to 100 random layouts with the same average density. The dashed vertical line refers to the average landmark density associated with

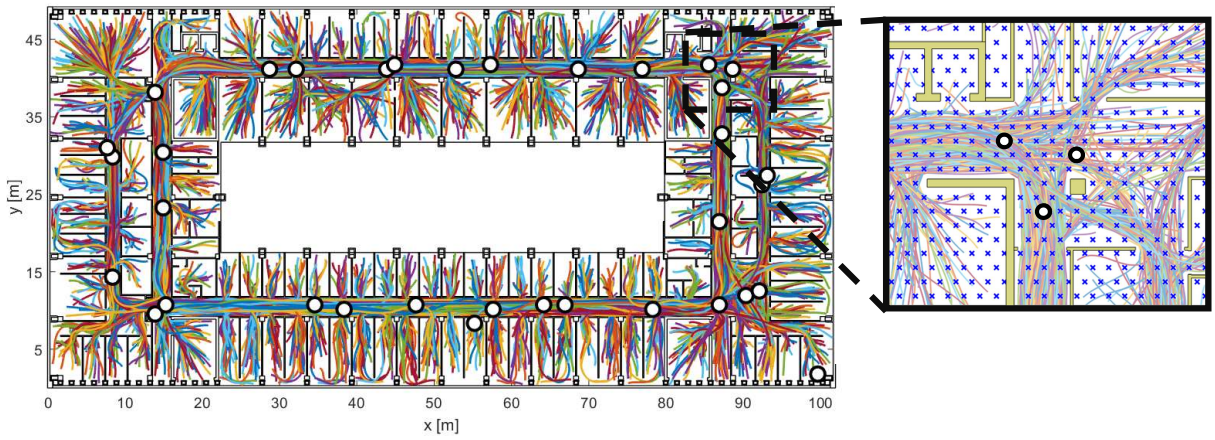


Fig. 8. DISI map with 2085 possible *FriWalk* trajectories generated by the path planner described in [37], [41]. The set of potential Aruco codes locations (namely the starting set for optimal landmark placement) are represented by cross-shaped markers (clearly visible in the inlet on the right) and consists of $|\mathcal{L}_f| = 9420$ elements. The landmarks selected by the proposed optimization algorithm (i.e. the elements of set $\mathcal{L}_g$) are highlighted with white circle markers.

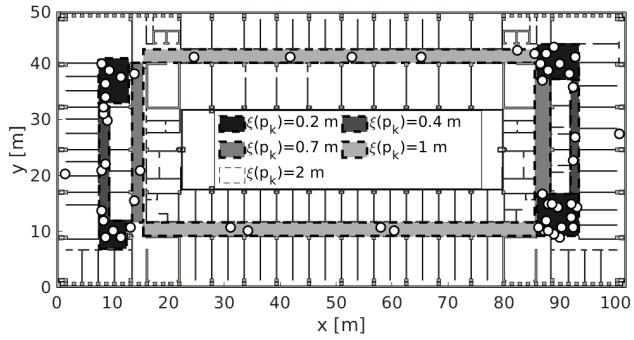


Fig. 9. Optimal landmark placement over the DISI map when different target uncertainty values $\xi(p_k)$ are used, i.e., 0.2 m (halls next to staircases), 0.4 m (narrow vertical corridors), 0.7 m (wide vertical corridors), 1.0 m (long horizontal corridors) and 2.0 m (other rooms).

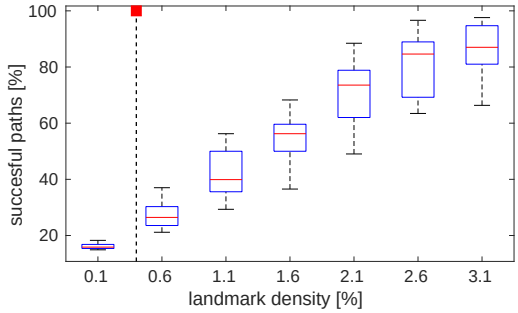


Fig. 10. Percentage of paths satisfying the uncertainty constraint $\xi(p_k) = 0.8$ m as a function of the average landmark density, assuming that they are randomly selected from $\mathcal{L}_f$ with the same probability among those shown in Fig. 8. The vertical dashed line highlights the average landmark density associated with the greedy solution, while the square on top of the line recalls that the uncertainty constraint is never violated in this case.

the greedy placement solution shown in Fig. 8, for which the uncertainty constraint is met over all paths. Clearly, the greedy solution outperforms the purely random approach. Indeed, to have a negligible probability that the positioning uncertainty constraint is violated in the random case, the average landmark density must be about one order of magnitude larger than when the greedy solution is adopted.

Of course, even when the greedy placement algorithm is applied, the resulting average landmark density is a function of the maximum wanted uncertainty, as qualitatively shown in Fig. 9. A better analysis of the relationship between average landmark density and $\xi(p_k)$ is shown in Fig. 11, where, unlike the case of Fig. 9, $\xi(p_k)$ is assumed to be constant over the whole DISI map and it is increased by steps of 0.1 m. If $\xi(p_k)$ is small, the average landmark density increases sharply. On the contrary, as $\xi(p_k)$ grows, it tends to decrease slowly. In particular, for $\xi(p_k) \geq 0.5$ m the average landmark density is well below 1%.

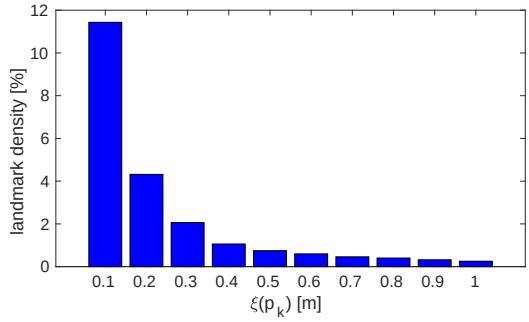


Fig. 11. Landmark density of the greedy placement solution, as function of the uncertainty constraint $\xi(p_k)$.

VII. Experimental Results

In principle, the placement results shown in Fig. 8 are based on the assumption that all points of every DISI room are

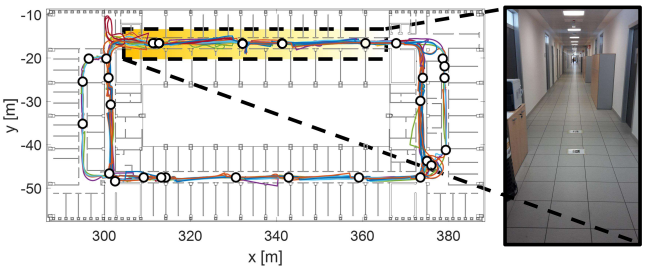


Fig. 12. Paths followed by the *FriWalk* for experimental validation. Again the white circle markers represent the locations where the Aruco codes are actually placed, i.e., the elements of set $\mathcal{L}_g$. The picture on the right shows a snapshot of the actual landmark layout in a corridor.

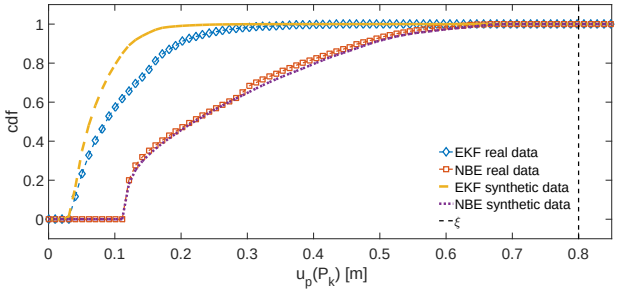


Fig. 13. Cumulative distribution functions of the uncertainty values $u_p(P_k)$ estimated by the NBE or the EKF using the data collected along the synthetic paths used to perform landmark placement and those collected during real experiments.

fully accessible. However, in practice this is not true, due to obvious privacy or security issues. Therefore, to plan a fair and appropriate experimental validation, the greedy placement algorithm was applied again considering a subset of all possible paths, i.e., limiting the analysis just to the rooms that are fully accessible. The results of this new landmark placement (assuming again that $\xi(p_k) = 0.8$ m) are shown in Fig. 12, along with a snapshot of the actual setup in a corridor. Again, landmark positions are indicated by white circle markers. Observe that in this case the amount of deployed Aruco codes is slightly smaller than in Fig. 8. In particular, $|\mathcal{L}_g| = 29$ instead of 35. This result is reasonable in consideration of the different area that can be actually explored.

With the Aruco codes deployed as shown in Fig. 12, 10 users were asked to move along various paths generated by the path planner, covering a total distance of about 4 km. The *FriWalk* position was estimated by the non-Bayesian algorithm (shortly referred in the following as NBE) described in Section II-C. The corresponding estimated paths are shown in Fig. 12. Observe that, due to the sporadic nature of landmark detection, the estimated paths may exhibit sudden and visible changes if an Aruco code is detected after a quite long time, i.e., when the uncertainty due to dead reckoning becomes particularly large. In principle, such sudden large errors should be smaller if a Bayesian estimator fusing odometry and vision system data (e.g. the Extended Kalman Filter - EKF - described in the Appendix) were used.

In order to highlight if and to what extent the uncertainty estimated over the aforementioned real paths is consistent with the uncertainty used to perform landmark placement, the cumulative distributions functions (CDFs) of $u_p(P_k)$ estimated on experimental and simulated paths are shown in Fig. 13. The dual results in the EKF case, namely the CDFs of $u_p(P_k)$ applied to the $P_k^{x,y}$ matrix extracted from (A.2), are also plotted for the sake of comparison. Note that, in the NBE case, the CDFs computed over real and simulated (i.e. synthetic) paths are perfectly consistent, and the percentile of $u_p(P_k)$ values exceeding $\xi(p_k) = 0.8$ m is negligible. This confirms that the landmark layout obtained as described in Section VI can be successfully applied to the chosen real paths, even if they are not exactly the same as those used to perform landmark placement. Moreover, the adopted landmark layout is clearly conservative, since, the percentiles associated with a given $u_p(P_k)$ value in the EKF case are always larger than the dual percentiles obtained with the NBE. However, in the EKF case a slight mismatch exist between the CDFs computed over the synthetic paths and those resulting from real experiments. This is probably due to the fact that not all assumptions underlying the use of the EKF (e.g., process or measurement noise whiteness and uncorrelatedness) hold in practice.

It is worth emphasizing that the CDF curves plotted in Fig. 13 refer just to the *estimated* positioning uncertainty based on (5) and (A.2) for the NBE and the EKF, respectively. Hence, to verify whether the uncertainty constraint is met, the positioning uncertainty has to be reconstructed from the differences e_x and e_y between the $x-$ and $y-$axis coordinates estimated by the *FriWalk* and those of some reference points, e.g., anytime one ARUCO code is detected. Unfortunately, no continuous position tracking is possible in DISI premises, since the Optitrack reference system cannot be used in such a large environment. The marginal empirical Probability Density Functions (PDFs) of e_x and e_y resulting from Gaussian fitting are shown in Fig. 14. For the sake of comparison, the PDFs of e_x and e_y obtained by applying the EKF to the same set of real paths are also reported. The position errors along the $x-$axis are affected on average by a 18 cm bias, probably because of the processing delays when the robot moves forwards, as described in [12]. However, this systematic contribution can be easily compensated. As expected, the positioning uncertainty associated with the EKF is smaller than the NBE one, due to the Bayesian nature of the former approach. Consider that, if e_x and e_y were uncorrelated, i.e., if $\rho = 0$ in (7), then $u_p(P_k) = u_p^- = \max(\sigma_x, \sigma_y)$. As a result, the actual values of u_p^- based on experimental data would be equal to 0.49 m for the NBE and 0.39 m for the EKF, respectively. However, the general scenario is worse, as some correlation between e_x and e_y could increase $u_p(P_k)$ (namely the maximum eigenvalue of $P_k^{x,y}$) till reaching $u_p^+ = \sqrt{\sigma_x^2 + \sigma_y^2}$ for perfectly correlated data, as explained in Section II-C. In this case, u_p^+ could reach 0.54 m for the NBE and 0.41 m for the EKF, respectively. Nevertheless, these values are well below $\xi(p_k) = 0.8$ m. Hence, in light of the discussion in Section II-C, the target uncertainty constraint is met for any $|\rho| \leq 1$, thus confirming

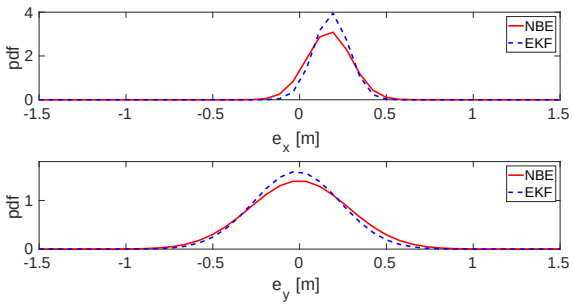


Fig. 14. Empirical marginal PDFs of estimation errors along $x-$axis and $y-$axis associated with the non-Bayesian position estimator described in Section II-C (dashed line) and the EKF described in the Appendix (solid line).

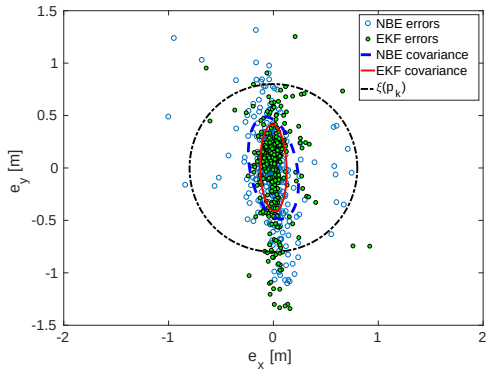


Fig. 15. Scatter diagrams of the position estimation errors e_x and e_y associated with the NBE (empty circles) and the EKF (filled circles). In the same graph, the ellipses corresponding to covariance matrices $P_k^{x,y}$ given by (7) for the NBE (dashed line) and for the EKF (solid line) are also shown for the sake of comparison with the circle of radius $\xi(p_k)$ delimiting the target uncertainty region (dash-dotted line).

the correct operation of the greedy placement algorithm. This theoretical achievement is confirmed by the scatter diagrams shown in Fig. 15, which reports 365 position estimation errors e_x and e_y associated with the NBE (empty circles) and the EKF (filled circles) immediately before landmark detection. The ellipses corresponding to the covariance matrices of either cloud of points are also plotted. In this case, the values of correlation coefficients ρ are approximately equal to -0.3 and -0.09 for the NBE and the EKF, respectively. Thus, the positioning uncertainly of either estimator is certainly included between u_p^- and u_p^+. The difference in ρ values explains also the diversity in shape between the ellipses shown in Fig. 15. However, in both cases the ellipses (as well as the possible circles circumscribing them) are safely included within the wanted uncertainty region, namely the dash-dotted circle of radius $\xi(p_k)$ shown in Fig. 15. Indeed, just about 10% of position error values lie outside that circle, i.e., much less than 33% that we would expect in a perfectly Gaussian case.

VIII. Conclusion

Indoor positioning techniques for mobile agents often rely on the deployment of a large amount of landmarks with a known position and orientation in a given reference frame. Since agents typically can estimate their own position through dead-reckoning techniques even when landmarks are not detected, usually landmarks are placed randomly or following just common-sense criteria that, however, do not ensure that given uncertainty constraints are met. In this paper, the problem of optimal landmark placement first is properly formalized in the framework of logic synthesis, and then it is solved through a greedy approach, which keeps into account the possible paths of the agents within the environment considered. The key advantage of the proposed technique is that it is able to place a very low number of landmarks, while ensuring that indoor localization uncertainty does not exceed a given limit. Even if the greedy algorithm generally does not converge to the globally optimal solution, multiple simulation results show that the number of landmarks deployed with the adopted heuristic approach is just slightly larger than the lower bound to the actual optimal solution. On the contrary, a much larger amount of randomly deployed landmarks is needed to achieve the same positioning uncertainty. Moreover, since the greedy algorithm is computationally light, it can be effectively used even when large environments are considered.

The proposed approach was validated on the field using a reasonable body of experimental data in a real–life scenario. In the future, the performance of the placement strategy could be further improved if the importance of different paths were taken into account, e.g., by classifying paths as mandatory or optional.

Appendix – EKF Description

This section describes the Bayesian estimator, i.e. an Extended Kalman Filter (EKF), selected to compare the localization accuracy achieved experimentally after optimal landmark placement with and without sensor data fusion, as explained in Section VII. The EKF relies on system equations (1) and (2), and, as customary of EKF implementation, it consists of two steps, i.e. *Prediction* and *Update*.

- The equations of the *Prediction* steps are:

$$\begin{aligned} p_{k+1}^- &= p_k + G_k(p_k) q_k, \\ P_{k+1}^- &= A_k P_k A_k^T + G_k(p_k)\, E\, G_k(p_k)^T, \end{aligned} \tag{A.1}$$

  where matrix $A_k = \left(I + \frac{\partial G_k(p_k) q_k}{\partial p_k}\right)$ is the same adopted by the non-Bayesian estimator;
- The equations of the *Update* step (which however is performed only when an Aruco code is detected) are instead

$$\begin{aligned} K_k &= P_{k+1}^- H_k^T \left(H_k P_{k+1}^- H_k^T + N\right)^{-1}, \\ p_{k+1} &= p_{k+1}^- + K_k (z_{k+1} - p_{k+1}^-), \\ P_{k+1} &= (I - K_k H_k)\, P_{k+1}^-, \end{aligned} \tag{A.2}$$

  where K_k is the Kalman gain, $H_k = \frac{\partial h(p_k)}{\partial p_k}$ is the Jacobian matrix of the system output function and z_{k+1}

is the vector of measurement data used to update the estimated state anytime an Aruco code is detected.

ACKNOWLEDGMENT

The activities described in this paper have received funding from the European Union Horizon 2020 Research and Innovation Programme - Societal Challenge 1 (DG CONNECT/H) under grant agreement no. 643644 for the project "ACANTO - A CyberphysicAl social NeTwOrk using robot friends".

REFERENCES

[1] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[2] S. He, B. Ji, and S. H. G. Chan, "Chameleon: Survey-free updating of a fingerprint database for indoor localization," *IEEE Pervasive Computing*, vol. 15, no. 4, pp. 66–75, Oct. 2016.

[3] D. Macii, A. Colombo, P. Pivato, and D. Fontanelli, "A data fusion technique for wireless ranging performance improvement," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 1, pp. 27–37, Jan. 2013.

[4] Z. Li, Z. Tian, M. Zhou, Z. Zhang, and Y. Jin, "Awareness of line-of-sight propagation for indoor localization using Hopkins statistic," *IEEE Sensors Journal*, vol. 18, no. 9, pp. 3864–3874, May 2018.

[5] F. Höflinger, J. Müller, R. Zhang, L. M. Reindl, and W. Burgard, "A wireless micro inertial measurement unit (IMU)," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 9, pp. 2583–2595, Sep. 2013.

[6] A. Colombo, D. Fontanelli, D. Macii, and L. Palopoli, "Flexible indoor localization and tracking based on a wearable platform and sensor data fusion," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 4, pp. 864–876, Apr. 2014.

[7] S. Li, M. Hedley, I. B. Collings, and M. Johnson, "Integration of IMU in indoor positioning systems with non-gaussian ranging error distributions," in *IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Savannah, GA, USA, Apr. 2016, pp. 577–583.

[8] E. Menegatti, A. Pretto, A. Scarpa, and E. Pagello, "Omnidirectional vision scan matching for robot localization in dynamic environments," *IEEE Transactions on Robotics*, vol. 22, no. 3, Jun. 2006.

[9] Q. Sun, J. Yuan, X. Zhang, and F. Sun, "RGB-D SLAM in indoor environments with sting-based plane feature extraction," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1071–1082, Jun. 2018.

[10] A. Perttula, H. Leppkoski, M. Kirkko-Jaakkola, P. Davidson, J. Collin, and J. Takala, "Distributed indoor positioning system with inertial measurements and map matching," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 11, pp. 2682–2695, Nov. 2014.

[11] R. Liu, C. Yuen, T. N. Do, and U. X. Tan, "Fusing similarity-based sequence and dead reckoning for indoor positioning without training," *IEEE Sensors Journal*, vol. 17, no. 13, pp. 4197–4207, Jul. 2017.

[12] P. Nazemzadeh, D. Fontanelli, D. Macii, and L. Palopoli, "Indoor Localization of Mobile Robots through QR Code Detection and Dead Reckoning Data Fusion," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 6, pp. 2588–2599, Dec. 2017.

[13] L. Mainetti, L. Patrono, and I. Sergi, "A survey on indoor positioning systems," in *Proc. International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, Croatia, Sep. 2014, pp. 111–120.

[14] D. Dardari, P. Closas, and P. M. Djuri, "Indoor tracking: Theory, methods, and technologies," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1263–1278, Apr. 2015.

[15] J. Kramer and A. Kandel, "Robust small robot localization from highly uncertain sensors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 4, pp. 509–519, Jul. 2011.

[16] P. Nazemzadeh, F. Moro, D. Fontanelli, D. Macii, and L. Palopoli, "Indoor positioning of a robotic walking assistant for large public environments," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 11, pp. 2965–2976, Nov. 2015.

[17] M. Wang, Y. Liu, D. Su, Y. Liao, L. Shi, J. Xu, and J. V. Miro, "Accurate and real-time 3-D tracking for the following robots by fusing vision and ultrasonar information," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 997–1006, Jun. 2018.

[18] A. A. Khaliq, F. Pecora, and A. Saffiotti, "Inexpensive, reliable and localization-free navigation using an RFID floor," in *European Conf. on Mobile Robots (ECMR)*, Lincoln, United Kingdom, Sep. 2015.

[19] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson, "Landmark selection for vision-based navigation," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 334–349, Apr. 2006.

[20] L. H. Erickson and S. M. LaValle, *An art gallery approach to ensuring that landmarks are distinguishable*. MIT Press, 2012, pp. 81–88.

[21] S. Thrun, "Finding landmarks for mobile robot navigation," in *IEEE Intl. Conf. on Robotics and Automation*, vol. 2. IEEE, 1998.

[22] H. Strasdat, C. Stachniss, and W. Burgard, "Which landmark is useful? Learning selection policies for navigation in unknown environments," in *IEEE Intl. Conf. on Robotics and Automation*. IEEE, 2009.

[23] P. Nazemzadeh, D. Fontanelli, and D. Macii, "Optimal Placement of Landmarks for Indoor Localization using Sensors with a Limited Range," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Madrid, Spain, Oct. 2016, pp. 1–8.

[24] R. Falque, M. Patel, and J. Biehl, "Optimizing placement and number of RF beacons to achieve better indoor localization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, May 2018, pp. 2304–2311.

[25] M. P. Vitus and C. J. Tomlin, "Sensor placement for improved robotic navigation," *Robotics: Science and Systems VI*, p. 217, 2011.

[26] M. Beinhofer, J. Müller, and W. Burgard, "Effective landmark placement for accurate and reliable mobile robot navigation," *ROBOT. AUTON. SYST.*, vol. 61, no. 10, pp. 1060–1069, Oct. 2013.

[27] V. Magnago, L. Palopoli, R. Passerone, D. Fontanelli, and D. Macii, "A Nearly Optimal Landmark Deployment for Indoor Localisation with Limited Sensing," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Sapporo, Japan: IEEE, Sep. 2017, pp. 1–8.

[28] V. Magnago, P. Bevilacqua, L. Palopoli, R. Passerone, D. Fontanelli, and D. Macii, "Optimal Landmark Placement for Indoor Positioning using Context Information and Multi-sensor Data," in *Proc. IEEE Int. Instrumentation and Measurement Technology Conference (I2MTC)*, Houston, Texas (US), 2018.

[29] M. Beinhofer, J. Müller, and W. Burgard, "Near-optimal landmark selection for mobile robot navigation," in *IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, 2011, pp. 4744–4749.

[30] "ACANTO: A CyberphysicAl social NeTwOrk using robot friends," http://www.ict-acanto.eu/acanto, February 2015, EU Project.

[31] L. Palopoli et al., "Navigation Assistance and Guidance of Older Adults across Complex Public Spaces: the DALi Approach," *Intelligent Service Robotics*, vol. 8, no. 2, pp. 77–92, 2015.

[32] ISO/IEC Guide 98-3:2008, *Uncertainty of measurement – Part 3: Guide to the expression of uncertainty in measurement (GUM:1995)*, Jan. 2008.

[33] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *Int. J. Robot. Res.*, vol. 24, no. 1, pp. 31–48, 2005.

[34] T. Sasaki, D. Brscic, and H. Hashimoto, "Human-Observation-Based Extraction of Path Patterns for Mobile Robot Navigation," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1401–1410, April 2010.

[35] E. Sentovich et al., "Sis: A system for sequential circuit synthesis," EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M92/41, 1992.

[36] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[37] P. Bevilacqua, M. Frego, D. Fontanelli, and L. Palopoli, "Reactive Planning for Assistive Robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1276–1283, April 2018.

[38] V. Magnago, M. Andreetto, S. Divan, D. Fontanelli, and L. Palopoli, "Ruling the Control Authority of a Service Robot based on Information Precision," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, Australia: IEEE, May 2018, to appear.

[39] M. Andreetto, S. Divan, F. Ferrari, D. Fontanelli, L. Palopoli, and F. Zenatti, "Simulating passivity for Robotic Walkers via Authority-Sharing," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1306–1313, April 2018.

[40] F. Moro, A. D. Angeli, D. Fontanelli, R. Passerone, D. Prattichizzo, L. Rizzon, S. Scheggi, S. Targher, and L. Palopoli, "Sensory stimulation for human guidance in robot walkers: A comparison between haptic and acoustic solutions," in *IEEE International Smart Cities Conference (ISC2)*, Trento, Italy, Sept. 2016, pp. 1–6.

[41] P. Bevilacqua, M. Frego, E. Bertolazzi, D. Fontanelli, L. Palopoli, and F. Biral, "Path Planning maximising Human Comfort for Assistive Robots," in *IEEE Conference on Control Applications (CCA)*. Buenos Aires, Argentina: IEEE, Sept. 2016, pp. 1421–1427.

[42] G. Arechavaleta, J. Laumond, H. Hicheur, and A. Berthoz, "An optimality principle governing human walking," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 5–14, Feb 2008.

[43] F. Farina, D. Fontanelli, A. Garulli, A. Giannitrapani, and D. Prattichizzo, "Walking Ahead: The Headed Social Force Model," *PLOS ONE*, vol. 12, no. 1, pp. 1–23, 01 2017.