

Robot localization from minimalist inertial data using a Hidden Markov Model

Abstract—Hidden Markov Models (HMM) are applied to interoceptive data (in this case the sense of rotation by way of a gyroscope) acquired by a moving wheeled robot when contouring an indoor environment. We demonstrate the soundness of HMM to solve the problem of robot localization in a topological model of the environment, particularly the kidnapped robot problem and position tracking. In this approach, the environment topology is described by the sequence of movements a robot executes when contouring the environment. Movements are described in a fuzzy domain using distance traveled and curvature as features.

I. INTRODUCTION

Localization is one of the most important problems in mobile robotics. A robot whose success depends on this capability cannot be ascribed other goals if it is not capable of localizing itself – localization is then fundamental.

In addition to odometry, visual or otherwise, localization could be improved by associating places with interesting features (a particular corridor, a green plant in a pot, etc.). In order to recognize such interesting modal diverse features, the robot needs an adequate set of sensors, just because sensed information is a building block of the robot’s maps, and, consequently, the robot’s actions.

Multiple sensors imply sensor fusion, which is not easy to program. In fact, depending on the robot’s mission, or the part of the mission the robot is in, not all sensations should be reckoned. We approach navigation by a rather minimalist sensor apparatus – the robot will rely only on gyroscope data; hence favoring the richness of proprioception.

Any sensor is characterized by inherent uncertainty, i.e., when measuring physical quantities one does not know the true value, but only an approximation. Depending on the sensor, data can soon be outdated (if it takes too much processing on a moving robot), inaccurate (e.g., due to bad calibration), or erroneous (e.g., due to aliasing), upon which the best one can do is to establish an upper bound on the error ($error = x_{true} - x_{measured}$). Therefore it’s important to supply the robot with the means to deal with uncertainty.

Probability theory and statistics are important frameworks to deal with uncertainty, either about the sensed information, the robot’s internal state and the outcome of its actions, of which Markov Chains play a central role.

In particular, in Markov localization a probability distribution over the space of all possible robot positions in a Cartesian frame of reference is maintained. Typically, by all possible positions it is meant every position/place in the environment, even those that are uninteresting from the point of view of the robot’s goals. The arrival of a measurement

triggers a belief update about the robot’s position, which is a daunting computational task [1].

This is the dense model approach to robot localization (maps are metric and grid oriented), to which we confront a sparse approach (maps are topological or hybrid), where modeling is oriented to “interesting” parts of the environment, while the rest is ignored [2],[3].

The preference for the low dimensionality of the data representation over the high dimensionality of the sensor data has been investigated by others (e.g., [4]), either because of gains in computational efficiency, or because brains appear to be very efficient at processing saliency [5].

II. HIDDEN MARKOV MODELS

The probability framework used is HMM [6],[7], which supplies the means to make inferences about states (the localization of the robot) which are not directly and unmistakably observed by the robot.

The parameters of HMM λ are $(Q, O, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$:

- $Q = \{s_1, s_2, \dots, s_N\}$ is the (ordered) set of possible model states, with N representing the number of states.
- $O = \{\nu_1, \nu_2, \dots, \nu_M\}$ is the (ordered) set of possible observable symbols, with M representing the number of symbols emitted by the system.
- $\mathbf{A} = \{a_{ij} : a_{ij} = P(S_{t+1} = s_j | S_t = s_i)\}$, $1 \leq i, j \leq N$, is the state transition probabilities, $S \in Q$ refers to the random variable (RV) taking values on Q . In S_t , t represents time. Since Q is an ordered set, $S_t = s_i$ is the same as $S_t = i$. This convention also applies to observations.
- $\mathbf{B} = \{b_{ij} : b_{ij} = P(V_t = j | S_t = i)\}$, $1 \leq j \leq M, 1 \leq i \leq N$, is the observation probability distribution conditioned upon the state. $V \in Q$.
- $\boldsymbol{\pi} = \{\pi_i\} : \pi_i = P(S_1 = i)$, $1 \leq i \leq N$ denotes the initial state distribution.

Two notations are used for sequences: i) normal, as in V_1, V_2, \dots, V_n ; and ii) compact, as in $V_{1:n}$. RV’s are written with capital letters, their respective values are written in lower case, and arrays are written in boldface.

We make the usual assumptions:

- **The Markov assumption** – The next state is dependent upon the current state: $a_{ij} = P(S_{t+1} = j | S_t = i)$, i.e., we consider first order Markov models.
- **The stationary assumption** – The state transition probabilities are independent of the actual time at which the transitions take place.

$$P(S_{t_1+1} = j | S_{t_1} = i) = P(S_{t_2+1} = j | S_{t_2} = i), t_1 \neq t_2$$

- **The output independence assumption** The current observation is independent of the previous ones, given the state. Formally, considering sequence $v_{1:T}$ of observations and sequence $q_{1:T}$ of states, then

$$P(v_{1:T}|q_{1:T}, \lambda) = \prod_{t=1}^T P(v_t|q_t, \lambda)$$

A. The three problems for HMM in robotic navigation

Following [6], the three problems in HMM whose solution would allow interesting real-world applications are:

- 1) given an observation sequence, $v_{1:n}$, and a model, λ , how the probability of $v_{1:n}$, given the model, $P(v_{1:n}|\lambda)$, can be computed?
- 2) Given an observation sequence, $v_{1:n}$, and a model, λ , there must be a state sequence, $q_{1:n}^*$, that best explains the observations. How can $q_{1:n}^*$ be computed?
- 3) Given an observation sequence, $v_{1:n}$, and a model, λ , how model parameters $(\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ can be adjusted in order to maximize $P(v_{1:n}|\lambda)$?

From a robotic navigation point of view, the interest of these three problem is obvious:

- 1) problem 1 provides the means to localize the robot, i.e., given a sequence of movements it evaluates the most probable location;
- 2) problem 2 provides the means to evaluate the most probable trajectory in state space, i.e., the one that best explains a sequence of movements. It is advised for state tracking as it provides a state history given a sequence of movements.
- 3) problem 3 provides the means to keep the map in sync with the environment (i.e., learning), a most desirable feature for dynamic environments.

In this paper we approach the first three problems.

III. MOBILE ROBOT SETUP

We adopt a map of a structured, indoor environment of a topological nature, where the environment is defined by a sequence of linguistic (i.e., fuzzy) descriptions representing movements like *small straight movement, followed by a large (with respect to distance) wide (with respect to curvature) curve to the left*. This requires that the robot maintains sensory contact with the environment, which is achieved by having the robot following the environment contours (see [8] for details).

Movements are delimited by the places in the environment where the spatial derivative of curvature changes sign or is zero. For example, while the robot finds itself moving straight, it collects odometry information that will be used to classify the distance traveled (is it short, medium or big?). This process is interrupted when there is a change of a certain amount in the curvature of the robot's trajectory. There, the straight movement ends and is finally classified (more on that in a while), and a new (curved) one starts. In this way, a robot trajectory is segmented into curved and straight pieces of different scales. See [8] for further details.

More concretely, *distance* is described by three linguistic terms: *small* (label S), *medium* (M) and *large* (L). Curvature of movements is described by three linguistic terms: *small* (s), *medium* (m), and *large* (l), and a label for zero curvature (Z). These labels were chosen to represent small curves (about 45°), 90° turns and U turns (about 180°), respectively. Since the robot will have to perform curves to either side, *curvature* is defined by seven fuzzy labels: *large left* (Ll), *medium left* (Lm), *small left* (Ls), *Straight* (Z), *small right* (Rs), *medium right* (Rm) and *large right* (Rl). In this vocabulary, a movement like *medium straight* is described as MZ , and one like *large U curve to the right* as LRL .

The amount of labels for classifying distance and curvature can be enlarged, providing more discriminative power to describe the environment, but at the expense of an higher computational load. In fact, if distance and curvature are described by U and V fuzzy terms, respectively, then a path made of w movements is a point in a $(UV)^w$ dimensional space. In the case of indoor environments we observe that most of the curves are at right angles; small curves are made to contour small obstacles; and sometimes the robot needs to perform U turns when reaching corridor ends. So, our choice of three classes (or fuzzy labels) to the left, three to the right and one for straight movements reflects this reality.

We also assume a home-centric robot, starting and ending all its missions in a unique place called *home*, which the robot innately recognizes with a high degree of success (e.g., home can be characterized by a pattern of colors, sounds, a tight corridor with known dimensions, a radio beacon, etc.).

Given this setup, the classification of the performed movements is what the robot observes from the environment. The states will be the environment singular locations where one movement ends and a new one starts.

For the sake of presenting a practical context that will help us explaining the methodology, we consider the particular real environment depicted in fig. 1.

Near places P and S there is a double door, that when opened discloses a small corridor, whose trajectory by the robot is represented by a dashed line. The dotted line represents a possible trajectory around the environment, with the respective singular places annotated through the character X and denoted by capital letters. Home is at the bottom-left. The robot is depicted as having the pose prior to leave home. It contours the environment by its left side, so, when leaving, it proceeds up, hence the first curve from place A to B.

The parameterization of the HMM must be derived from this environment. As explained, states will correspond to the locations where one movement ends and a new one starts, i.e., $Q = \{A, B, \dots, U\}$. N , the cardinal of Q , i.e., $N = |Q|$, will be 21, corresponding to the total of capital letters in fig. 1. The observation set corresponds to the Cartesian product between the set of labels for distance and for curvature, which is, $O = \{SLl, SLm, SLs, SZ, SRs, SRm, SRI, MLI, MLm, MLs, MZ, MRs, MRm, MRI, LLI, LLm, LLs, LZ, LRs, LRm, LRI\}$. Thus, M , the cardinal of O , amounts to $M = |Q| \times |O| = 3 \times 7 = 21$.

The mobile robot we have used is a Pioneer 3 [9],

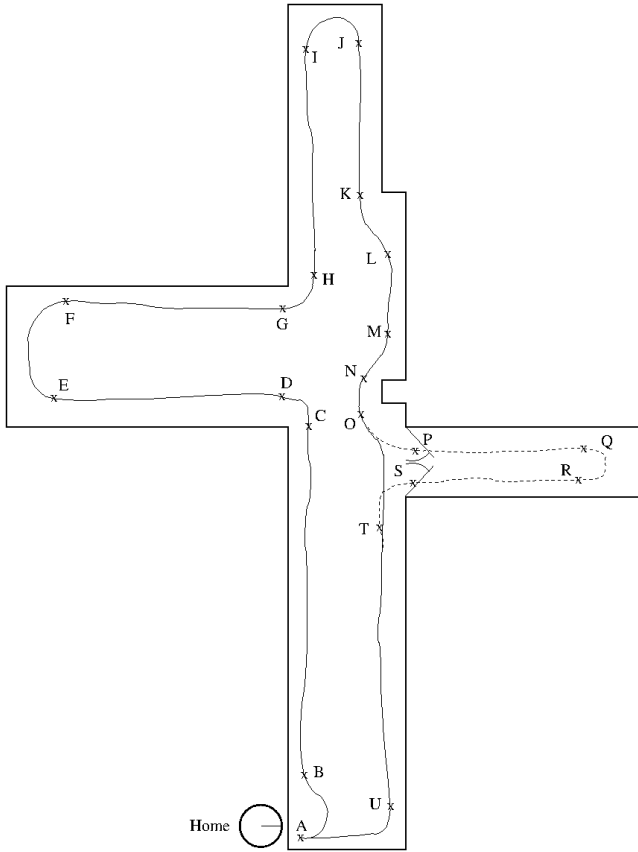


Fig. 1. Air-view of the test environment, with an ordinary trajectory resulting from a mobile robot contouring it, properly annotated with places.

equipped with a ITG-3200 gyroscope [10] from which all the data is sampled. Figure 2 presents the raw data collected in a typical tour around the environment (denoted original).

This data is filtered through an Infinite Impulse Response (IIR) filter defined by $y_i = \alpha y_{i-1} + (1-\alpha)x_i$, where $\alpha = 0.8$, x_i represents the i -th sample from the gyro and y represents its filtered version (see fig. 2). This can be regarded as a convex combination between the past and the present, through α . It's interesting because is computationally efficient (two multiplications and one sum), in comparison with a moving window average, for instance.

The resulting data passes through a threshold filter, that separates straight movements from curved ones.

Finally, small straight movements and curved movements with small traveled distance are deleted. These correspond to the spikes in the graph denoted *filt+thresh* in fig. 2.

The last step corresponds to classify the crisp values of distance and curvature to the membership functions of fuzzy variables *distance* and *curvature*, respectively, thus calculating the labels of the movements. Remember, distance is defined by three fuzzy predicates and curvature by seven. Membership functions are of trapezoidal shape (refer to

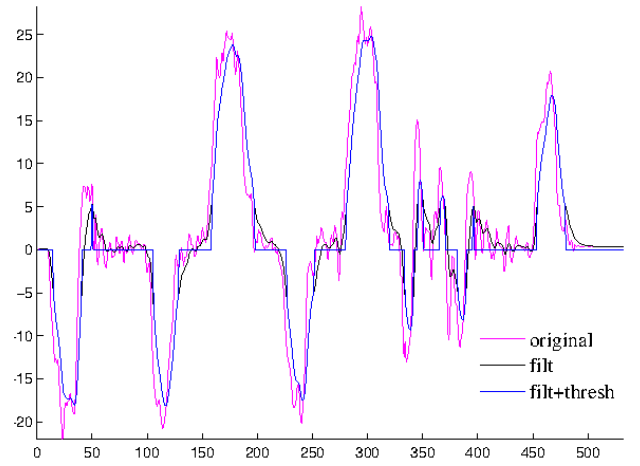


Fig. 2. Raw data sampled from a gyro, its filtered version and its thresholded version.

fig. 3), whose definitions are:

small(distance)	= (a = 0, b = 0, c = 16, d = 22)
medium(distance)	= (16, 22, 40, 60)
large(distance)	= (40, 60, 100, 100)
large-left(curvature)	= (-800, -800, -600, -400)
medium-left(curvature)	= (-600, -400, -200, -100)
small-left(curvature)	= (-200, -100, -5, 0)
straight(curvature)	= (-5, 0, 0, 5)
small-right(curvature)	= (0, 5, 100, 200)
medium-right(curvature)	= (100, 200, 400, 600)
large-right(curvature)	= (400, 600, 800, 800)

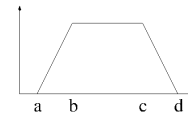


Fig. 3. Definition of a trapezoidal membership function.

The classification of the crisp values of distance and curvature is made by choosing the fuzzy set having the higher degree of membership. Consequently, as an example, the label sequence of the trajectory referred to in fig. 2 is $\{MLm, LZ, MLm, MZ, MRl, MZ, MLm, MZ, MRl, MLs, SRs, SZ, SRs, SZ, MLs, LZ, MRm\}$.

IV. SOLUTIONS TO THE THREE PROBLEMS

The initial state of the robot is home, so the initial state probability is $\pi_i = [i = A]$, using the Iverson bracket notation ($[X] = 1$ if proposition X is true, zero otherwise).

In this particular setup, there is no need to estimate π_i , since the robot always departs from home. Note that the environment is assumed to be closed, consequently the robot always finds home as long it contours the environment by the same side.

A. Problem 3 – mapping

Problem 3 is about learning the parameters of the HMM: $\lambda = (N, M, Q, O, A, B, \pi)$. We make the following simplifications:

- the environment was correctly analyzed, so there are no places that escaped our analysis; hence there is nothing to learn about N and Q .
- Our movement classification is complete, i.e., there are no movements that our classification strategy is not able to classify correctly, thus making M and O constant.
- The mobile robot starts and finishes every mission in home, so there is nothing to learn about π .

Learn \mathbf{A} means learn the map of the environment, i.e., know which places follow any other place. \mathbf{B} is a gateway to \mathbf{A} , since the state is hidden but the observations are not, so learning \mathbf{B} is as important as is learning \mathbf{A} . In this paper we restrict learning to establish priors on \mathbf{A} and \mathbf{B} .

There are several ways to assign priors to \mathbf{A} and \mathbf{B} , which we denote by $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$: Maximum Likelihood, Maximum Entropy (ME), Expectation/Maximization, among others. We will limit ourselves to use ME, which we briefly explain by recurring to the data collected from the environment in fig. 1. Without explanation, the training data is gathered from having the robot completing 10 laps around the environment; five around the regular environment and five around the extended environment.

The principle of maximum entropy is used to derive priors based on the information available, and use no other information besides that. Typically, the available prior information is incomplete, so, by the principle of maximum entropy, the prior density function should also be as uncertain as possible. In the case of a RV with an infinite but countable domain, i.e., in the $[0, \infty[$ interval, as is the case of the count of movements of the robot in the environment, the distribution that maximizes the entropy is Uniform.

By observing the contouring trajectories we can extract the sequence of places (states), and thus $\hat{\mathbf{A}}$ has the form of a connectivity matrix (i.e., $a_{ij} = 1$ if place j follows place i). Because $\hat{\mathbf{A}}$ is stochastic, i.e., their rows must sum to one, if place i connects to n_i other places, then we write $1/n_i$ in the intersection of the line corresponding to i with the columns corresponding to each successor. This way to define $\hat{\mathbf{A}}$ corresponds to an unbiased estimator for a_{ij} , as $1/n_i$ is an non-informative (Uniform) prior, thus conforming to the principle of maximum entropy. The same principle is applied to $\hat{\mathbf{B}}$.

\mathbf{A} and \mathbf{B} are manually learned from the data resulting from the robot contouring the environment a few times, consequently it is a supervised process. It is tedious for large environments but results in correct values if the analysis is complete.

In the case of our test environment, $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are depicted in figs. 4 and 5.

B. Problem 1 – localization

When the robot contours the environment by its left side, a set of places are visited (the place occupied by the

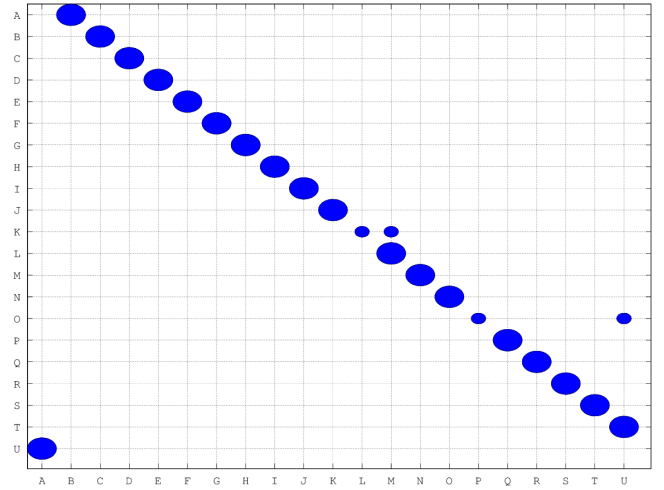


Fig. 4. $\hat{\mathbf{A}}$ evaluated by ME viewed in graphic form. The radius of the circles is proportional to the respective elements in $\hat{\mathbf{A}}$. The radius of the circles in a line sum to one.

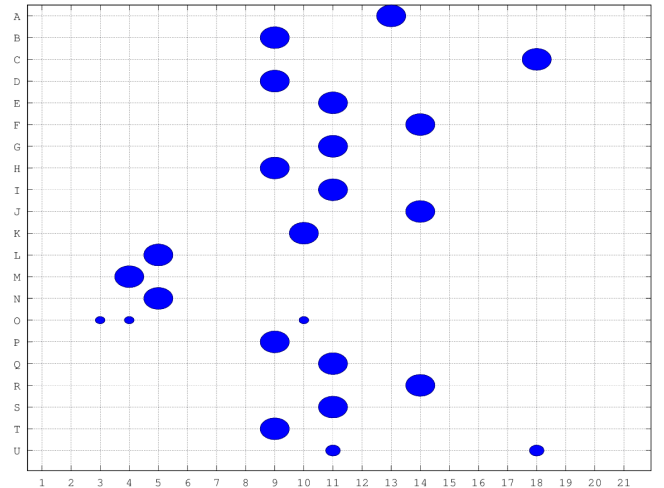


Fig. 5. $\hat{\mathbf{B}}$ evaluated by ME viewed in graphic form. In every state there is at least one observation; the converse is not true. The radius of the circles in a line sum to one.

robot is the hidden variable) and the corresponding set of observations made. Given a sequence of T observations, $v_{1:T}$, and a model, λ , we may want to find the state, q^* , that maximizes the probability

$$\alpha_t(i) = P(v_{1:t}, S_t = i | \lambda), 1 \leq t \leq T, 1 \leq i \leq N,$$

i.e., we may want to seek the most probable robot localization at time t taking in mind the T observations,

$$q^* = \arg \max_i \alpha_t(i)$$

$\alpha_t(i)$ is defined recursively by the Forward algorithm [6].

Initialization:

$$\alpha_1(j) = \pi_j b_{jv_1}, 1 \leq j \leq N.$$

Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_{jv_{t+1}}, \quad (1)$$

$$1 \leq j \leq N, 1 \leq t \leq n-1. \quad (2)$$

As an example of application, consider that the robot is lost and observes MZ . From fig. 1 we see that there are six places ($\{E,G,I,Q,S,U\}$) where the observed movement is MZ . The values of $\alpha_1(j)$ are depicted in the top graph of fig. 6. The places where $\alpha_1(j) : 1 \leq j \leq 21$ attains its maximum are ($\{E,G,I,Q,S\}$), so these are the most probable places for the robot to be. The robot continues to be lost because there are many places explaining the observation, consequently it performs one more movement, which turns out to be MRI . The observed sequence is now $\{MZ, MRI\}$. In the second graph from the top of fig. 6, we see that $\alpha_2(j)$ attains its maximum for $j \in \{6, 10, 18\}$, which translates to $\{F,J,R\}$. One more movement must be performed in order for the robot to be localized unequivocally. This line of reasoning continues until the robot completes the sequence MZ, MRI, MZ, MLm, MZ , leading the robot to be localized at place 9 (I). Performing one more movement leads the robot to be localized with probability one, hence the graph at the bottom of fig. 6.

In applying the Forward algorithm in this example, we considered π_j distributed uniformly over the 21 states. Finally, this methodology proves the adequacy of this procedure to solve the wake-up robot and the kidnapped robot problems [1], where the robot has no prior knowledge about its position.

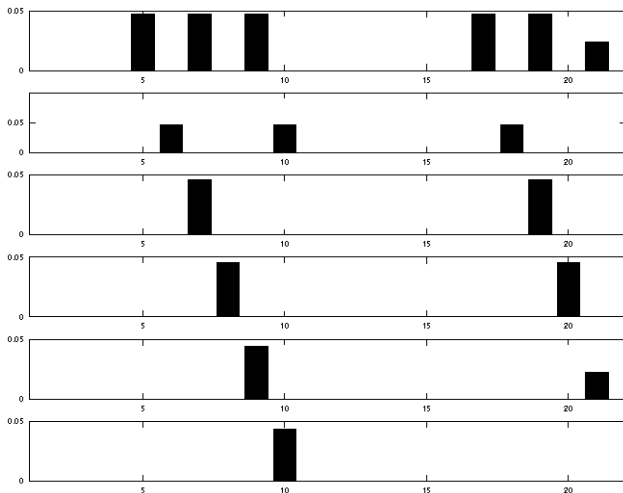


Fig. 6. $\alpha_1(j)$ (top), through $\alpha_6(j)$ (bottom), for each of the 21 places, corresponding to the example that shows how a lost robot can re-localize itself (see text for explanations).

C. Problem 2 – tracking

In problem 2 one is interested in computing the most probable state sequence, $q_{1:t}^*$, which is the one that best explains the observed sequence, $v_{1:n}$.

$$q_{1:t}^* = \arg \max_{q_1 q_2 \dots q_{t-1}} P(q_1 q_2 \dots q_{t-1} q_t, v_1 v_2 \dots v_t)$$

<i>SLI</i>	<i>SLm</i>	<i>SLs</i>	<i>SZ</i>	<i>SRs</i>	<i>SRm</i>	<i>SRI</i>
1	2	3	4	5	6	7
<i>MLI</i>	<i>MLm</i>	<i>MLs</i>	<i>MZ</i>	<i>MRs</i>	<i>MRm</i>	<i>MRI</i>
8	9	10	11	12	13	14
<i>LLI</i>	<i>LLm</i>	<i>LLs</i>	<i>LZ</i>	<i>LRs</i>	<i>LRm</i>	<i>LRI</i>
15	16	17	18	19	20	21

TABLE I
INDEXES FOR OBSERVATIONS

This problem is named the decoding problem, and is solved by the Viterbi algorithm [6], [11].

We apply this algorithm to the environment in fig. 1. We assume that the robot contours the environment starting at home. As it does, observations are collected and the corresponding most probable sequence of visited places is computed by the Viterbi algorithm.

Given an observation sequence and a state sequence, the corresponding joint probability is evaluated by eq. 3.

$$\begin{aligned} P(v_{1:n}, q_{1:n} | \lambda) &= \pi_{q_1} P(v_1 | q_1) \prod_{k=2}^n P(v_k | q_k) P(q_k | q_{k-1}) \\ &= \pi_{q_1} b_{v_1 q_1} \prod_{k=2}^n b_{v_k q_k} a_{q_k q_{k-1}} \end{aligned} \quad (3)$$

As a matter of limiting the space used in presenting the results, we index the movements to numbers, following table I. So, instead of mentioning SLm , for instance, numeral 2 is used.

V. APPLICATION OF HMM TO ROBOTIC NAVIGATION

1) *Small environment*: We start by having the robot contouring the small environment (i.e., the double door near places P and S is closed). Table II shows the results. The column at left shows the sequence of sensed movements as the robot progresses in the environment, the column in the middle shows the inferred place sequence, and the third column presents the probability of the state sequence.

The results of eq. 3 (and for the same purpose, the values of $\alpha_{t+1}(j)$) can approach zero (i.e., can be of the order of the smallest representable double number) for moderate real problems. The third column shows the probability values for this particular case; they are far from the minimum, hence there is no need to use scaling procedures (e.g., as the one presented in [6], pag. 272).

The complete movement sequence sensed by the robot is presented next. From its translation (see indexes in tab. I, and refer to fig. 1), we see that the robot's position is tracked correctly all the way.

$$\{MLm, LZ, MLm, MZ, MRI, MZ, MLm, MZ, MRI, MLs, SRs, SZ, SRs, MLs, LZ, MRm\} \quad (4)$$

2) *Big environment*: The results for the big environment are presented in tab. III. Again, tracking success is 100%.

observation	state	probability
9	B	1.0
9,18	BC	0.9
9,18,9	BCD	0.9
9,18,9,11	BCDE	0.72
9,18,9,11,14	BCDEF	0.72
9,18,9,11,14,11	BCDEFG	0.576
9,18,9,11,14,11,9	BCDEFGH	0.576
9,18,9,11,14,11,9,11	BCDEFGHI	0.4608
9,18,9,11,14,11,9,11,14	BCDEFGHIJ	0.4608
9,18,9,11,14,11,9,11,14,10	BCDEFGHIJK	0.2304
9,18,9,11,14,11,9,11,14,10,5	BCDEFGHIJKL	0.1152
9,18,9,11,14,11,9,11,14,10,5,4	BCDEFGHIJKLM	0.1152
9,18,9,11,14,11,9,11,14,10,5,4,5	BCDEFGHIJKLMN	0.1152
9,18,9,11,14,11,9,11,14,10,5,4,5,3	BCDEFGHIJKLMNO	0.04608
9,18,9,11,14,11,9,11,14,10,5,4,5,3,18	BCDEFGHIJKLMNOU	0.004608
9,18,9,11,14,11,9,11,14,10,5,4,5,3,18,13	BCDEFGHIJKLMNOUA	0.004608

TABLE II
STATE TRACKING FOR THE SMALL ENVIRONMENT

observation	state	probability
9	B	1.0
9,18	BC	0.9
9,18,9	BCD	0.9
9,18,9,11	BCDE	0.72
9,18,9,11,14	BCDEF	0.72
9,18,9,11,14,11	BCDEFG	0.576
9,18,9,11,14,11,9	BCDEFGH	0.576
9,18,9,11,14,11,9,11	BCDEFGHI	0.4608
9,18,9,11,14,11,9,11,14	BCDEFGHIJ	0.4608
9,18,9,11,14,11,9,11,14,10	BCDEFGHIJK	0.2304
9,18,9,11,14,11,9,11,14,10,5	BCDEFGHIJKL	0.1152
9,18,9,11,14,11,9,11,14,10,5,4	BCDEFGHIJKLM	0.1152
9,18,9,11,14,11,9,11,14,10,5,4,5	BCDEFGHIJKLMN	0.1152
9,18,9,11,14,11,9,11,14,10,5,4,5,3	BCDEFGHIJKLMNO	0.04608
9,18,9,11,14,11,9,11,14,10,5,4,5,3,9	BCDEFGHIJKLMNOP	0.02304
9,18,9,11,14,11,9,11,14,10,5,4,5,3,9,11	BCDEFGHIJKLMNOQP	0.02304
9,18,9,11,14,11,9,11,14,10,5,4,5,3,9,11,14	BCDEFGHIJKLMNOQRS	0.02304
9,18,9,11,14,11,9,11,14,10,5,4,5,3,9,11,14,11	BCDEFGHIJKLMNOQRST	0.02304
9,18,9,11,14,11,9,11,14,10,5,4,5,3,9,11,14,11,9	BCDEFGHIJKLMNOQRSTU	0.018432
9,18,9,11,14,11,9,11,14,10,5,4,5,3,9,11,14,11,9,11,13	BCDEFGHIJKLMNOQRSTUA	0.018432

TABLE III
STATE TRACKING FOR THE BIG ENVIRONMENT

3) *Modified small environment*: Finally, we consider a case where the environment was modified by occupying the hollow near places L and M with cabinets, i.e., from place K to place N the environment is now straight. The respective results are presented in tab. IV, and again tracking success is 100%. This illustrates further the capabilities of this approach to localization, in this case in a dynamical environment.

VI. CONCLUSION

The structure of indoor environments imposes specific shapes in the trajectories a robot executes when performing missions in it. The robot's internal sense of rotation, provided

observation	state	probability
9	B	1.0
9,18	BC	0.9
9,18,9	CD	0.9
9,18,9,11	BCDE	0.72
9,18,9,11,14	BCDEF	0.72
9,18,9,11,14,11	BCDEFG	0.576
9,18,9,11,14,11,9	BCDEFGH	0.576
9,18,9,11,14,11,9,11	BCDEFGHI	0.4608
9,18,9,11,14,11,9,11,14	BCDEFGHIJ	0.4608
9,18,9,11,14,11,9,11,14,18	BCDEFGHIJO	0.04608
9,18,9,11,14,11,9,11,14,18,18	BCDEFGHIJOU	0.004608
9,18,9,11,14,11,9,11,14,18,18,13	BCDEFGHIJOUA	0.004608

TABLE IV
STATE TRACKING FOR MODIFIED SMALL ENVIRONMENT

by ordinal inertial sensors, supplies the data from where the shape of the environment can be elicited. We have demonstrated that Hidden Markov Models provides a sound framework to localize the robot from this data.

We concentrated on the use of sparse models of the environment, where only interesting places are stored in the map. These are in contrast to dense ones, which require large computational resources. Interesting locations are defined by the places where the trajectory curvature changes. This representation could be complemented with other environment distinctive features, depending on the sensors available.

In our approach, movements are described by fuzzy predicates, which is a representation easily interpretable by humans and amenable to be exchanged among robots.

The learning problem (problem 3), which can be resumed as finding the values of \hat{A} and \hat{B} that maximizes $P(\Omega|\lambda)$, with Ω representing the set of all observation sequences made by the robot, starting from zero knowledge about the environment, was addressed by a supervised, manual, procedure, conforming to the principle of maximum entropy. The application of this procedure is not difficult for small to moderate environments, but it is tedious for large ones. An interesting follow-up is to make this procedure unsupervised.

Regarding problems 2 and 3, we proved (empirically) that HMM's are well suited to address the problems of localization and tracking in environments described by sparse models. The presented methodology can be extended to outdoor environments, in spite of the higher scale these have.

Mobile robots should have an action model, rendering actions dependent upon the interpretation of the environment, otherwise robots cannot navigate. We have not presented one, though. This topic is a natural continuation of our work.

REFERENCES

- [1] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, no. 11, pp. 391–427, 1999.
- [2] K. Konolige and K. Chou, "Markov localization using correlation," in *Proceedings of the International Joint Conference on AI (IJCAI)*, (Stockholm, Sweden), July 1999.
- [3] F. Martín, V. Matellán, J. M. Cañas, and C. Agüero, "Visual based localization of a legged robot with a topological representation," in *Robot Localization and Map Building* (H. Yussuf, ed.), pp. 409–422, InTech, 2010.
- [4] A. Torralba, K. Murphy, W. Freeman, and M. Rubin, "Context-based vision system for place and object recognition," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pp. 273–280 vol.1, oct. 2003.
- [5] C. K. Chang, C. Siagian, and L. Itti, "Mobile robot vision navigation and obstacle avoidance based on gist and saliency algorithms," in *Proc. Vision Science Society Annual Meeting (VSS11)*, May 2011.
- [6] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, feb 1989.
- [7] P. Dymarski, ed., *Hidden Markov Models, Theory and Applications*. InTech, 2011. (online edition).
- [8] A. Abreu and L. Correia, "An hybrid mapping approach with a fuzzy representation for mobile robots," in *The 11th International Conference on Advanced Robotics ICAR 2003*, June 30–July 3 2003.
- [9] *Pioneer 3-DX*, (accessed 23/Feb/2012). <http://www.mobilerobots.com/researchrobots/pioneer3dx.aspx>.
- [10] *ITG-3200 Integrated Triple-Axis Digital-Output Gyroscope*, (accessed 23/Feb/2012). <http://invensense.com/mems/gyro/itg3200.html>.
- [11] J. Forney, G.D., "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268 – 278, march 1973.