Active Sensing for Range-Only Mapping using Multiple Hypothesis

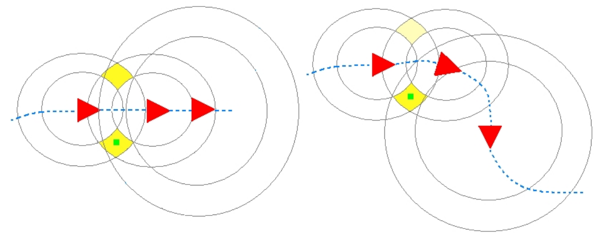
L. Merino, F. Caballero and A. Ollero

***Abstract*— Radio signal-based localization and mapping is becoming more interesting in robotics as applications involving the collaboration between robots and static wireless devices are more common. This paper describes a method for mapping with a mobile robot the position of a set of nodes using radio signal measurements. The method employs Gaussian Mixtures Models (GMM) for undelayed initialization of the position of the wireless nodes within a Kalman filter. Moreover, the paper extends the method to consider active sensing strategies in order to map the nodes. Entropy variation is used as a measurement of information gain, and allows to prioritize control actions of the robot. However, as there is no analytical expression for the entropy of a GMM, upper bounds of the entropy, for which close form computation is possible, are used instead. The paper describes simulations that show the feasibility of the approach.**

I. Introduction

Range-only mapping in wireless sensor networks is an active research area that poses a number of challenges, from range computation to map building. One of the key issues is the lack of bearing information in the measurements. This means that the real position of the emitter cannot be estimated based on a single measurement. The receiver must receive information in different positions (trilateration) to properly localize the emitter.

Initial results on wireless sensor localization using signal strength were conducted by Ladd et al. in [1], where Ethernet devices were used to localize and track a human operator inside a building using pre-computed probabilistic maps . Other approaches use delayed initialization through trilateration or employ other means to determine the bearing. For instance, in [2], radio-frequency (RF) transponders are used to build a range-only SLAM approach. An estimation of the time of flight is used to compute the distance between transmitter and receiver. Also, the partial directivity of the transponders simplifies the problem and allows undelayed gaussian initialization of the position. [3] also considers time of arrival and direction of arrival to successfully localize a wireless sensor network. [4] employs delayed initialization through trilateration. The authors proposed in [5] a solution for range-only mapping based on particle filtering over the received signal strength. However, it is not possible to integrate the estimation of the filter into more complex localization architectures such a SLAM until the particle filter has converged to a single solution, and thus it uses a delayed feature initialization.



(a) (b)

Fig. 1: Two examples of range-only localization. The robot (red triangle) receives range data from the beacon (green square) at three different positions. Yellow areas denote possible localizations of the beacon (as more intense is the yellow color, more likely this localization is). (a) Result of the node localization using a straight robot trajectory (there are two possible solutions for the localization). (b) Results of the node localization if the robot trajectory is adapted from active sensing considerations (the localization converges to single correct solution)

The problems related with multiple hypotheses in the early steps of the estimation in range-only localization approaches have been recently addressed in [6]. The paper describes an algorithm that allows delayed initialization of the node position by means of tracking the most probable two hypotheses. Subsequent measures provide enough information to discard the wrong hypothesis, then including the correct one into the SLAM filter. This problem has been also addressed and generalized by the authors to n-hypotheses in [7], where an undelayed initialization for wireless network mapping using range-only measurements is proposed. The paper makes use of Gaussian Mixtures Models (GMM) to represent the non-Gaussian prior distribution of the node position, allowing easy integration of the localization process into Kalman filters from the very first range measurement.

Most of the previous approaches do not take into account the possibility of controlling the robot, and the robot is just commanded a predefined path. However, active sensing strategies may lead to more efficient exploration and mapping using radio-signals. The robot can adapt its trajectory, avoiding for instance non-observable motions or following those

This work is partially supported by the ROBAIR Project funded by the Spanish Research and Development Program (DPI2008-03847), UAS-NET (P09-TEP-5120), funded by the Andalusian Government, and the Cooperating Objects Network of Excellence, CONET (ICT-224053), funded by the European Commission

L. Merino is with Pablo de Olavide University, Carretera Utrera km1, 41013, Sevilla (Spain). lmercab@upo.es

F. Caballero and A. Ollero are with the University of Seville, Escuela Superior de Ingenieros, Camino de los Descubrimientos s/n, 41092 Sevilla (Spain). fcaballero@us.es, aollero@cartuja.us.es

A. Ollero is also with the Center for Advanced Aerospace Technology (CATEC), Parque Tecnológico y Aeronáutico de Andalucía, C. Wilbur y Orville Wright 17-19-21, 41309, La Rinconada (Spain)

paths which are most informative, in the sense of reducing the uncertainty on the nodes' positions. Fig. 1 illustrates the benefit of considering active sensing strategies. Thus, the robot trajectory of Fig. 1.(a) results in two hypotheses with very similar uncertainty (bimodal distribution of the node position). On the other hand, Fig. 1.(b) shows how adapting the robot trajectory benefits the localization of the node.

Active sensing requires a measurement on the information gain obtained when executing a certain task or action. For Bayesian approaches, one possibility is to use the (expected) variation on the entropy of the beliefs on the nodes' positions as a measure of information gain, as for instance in [8] or [9], [10] for exploration and SLAM. In [11], the same ideas about active sensing are applied to a set-theoretic framework used to represent and handle uncertainty. In [12], active sensing strategies are applied to the problem of tracking using only range measurements, where the target is represented by a single Gaussian.

The main contribution of the paper is an extension of the approach presented in [7] for node mapping based on a weighted GMM to represent the non-Gaussian prior distribution of the node position, with an active sensing strategy in order to gather as much information as possible for localizing the nodes. Entropy variation is considered in this paper to measure the information gained by a given robot motion considering all node position hypotheses. As the computation of the entropy of GMM has not a closed form, entropy bounds are used in this approach. The paper will show how these bound provide an effective way for active sensing.

The paper is structured as follows. First, Section II summarizes the multiple hypotheses localization approach using Gaussian Mixtures Models. Section III presents the active sensing strategy employed. Later, Section IV shows results in simulation, followed by the conclusions and future work in Section V.

II. Range-only mapping and SLAM using Gaussian Mixtures

The objective can be summarized as estimating the position of wireless sensor nodes based on the received signal strength on a node attached to the robot. This process is depicted in Fig. 1: when the robot receives the first range information from the node to be localized, the initial position of the node is uniformly distributed around the robot position at the given distance.

The ability of representing arbitrary Non-Gaussian distributions as linear combination of Gaussian distributions allows simplifying the integration of such multi-modal distributions into Gaussian filters like the Kalman Filter and, hence, into the classical approaches for SLAM. This Section will show how GMMs can be included into the SLAM filter with the first measurement and how this model can be easily update with new measurements from the beacon (the reader is refereed to [7] for further details about the localization approach). Of course, the same approach can be used for WSN mapping if the robot pose is known.

A. The state vector

The state vector of the filter will be composed by the estimated 2D position and orientation of the robot, and the estimated 2D position of all the nodes. These positions will be considered as static. Thus, the state vector can be described as follows:

$$\mathbf{x} = [\mathbf{r}^T, \mathbf{b}_1^T, \mathbf{b}_2^T, \dots, \mathbf{b}_n^T]^T \tag{1}$$

where $\mathbf{r} = (x_r, y_r, \theta_r)^T$ represent the Euclidean position and orientation of the robot and $\mathbf{b}_i$ represents the position of the i-th beacon considered in the filter. The beacon position $\mathbf{b}_i$ will be expressed in polar coordinates with respect to the position from which the robot received the very first range information (x_i, y_i). Thus, if the beacon position was completely known, it would be expressed as $\mathbf{b}_i = [x_i, y_i, \rho_i, \theta_i]^T$, where ρ_i is the distance between the robot and the beacon position and θ_i is the angle.

However, this paper assumes non prior information about the angle of arrival of the beacon information, so the value of θ_i is unknown. We propose quantizing the space of possible values of θ_i into k possible hypotheses. Thus, each beacon will be expressed as follows:

$$\mathbf{b}_i = [x_i, y_i, \rho_i, \theta_{i0}, \theta_{i2}, \dots, \theta_{i(k-1)}]^T \tag{2}$$

All the hypotheses θ_{ij}, together with their weights w_{ij}, will compose a GMM such as the probability mass functions of θ_i, $f_{\theta_i}(x)$, will be uniformly distributed from 0 to 2π, that is:

$$f_{\theta_i}(x) = U(0, 2\pi) \simeq \sum_{j=1}^{k} w_{ij} \mathcal{N}(\theta_{ij}, \sigma_{ij}) \tag{3}$$

Then, it can be seen how the state vector presented in (1) will consist of the robot position/orientation estimation and the hypotheses of every beacon considered into the filter. Of course, the number of hypotheses will evolve, reducing its number as range information is integrated from different robot positions.

B. Node initialization

As previously introduced, after the first range information of a beacon is considered, the probability mass function of its position will be uniformly distributed around the robot location. This probability function will be approximated by a GMM using (3). Each of these Gaussians will be considered as an independent hypothesis into the localization filter.

The number k of hypotheses will be manually setup depending on the computational resources, because the length of the state vector depends on the number of nodes n and the number of hypotheses per beacon k through the expression $L = 3 + 3n + nk$. Using a very large number of hypotheses could be overfitting, while reducing this number too much could lead to inconsistencies. In this particular case, experimental results showed that 8 hypotheses are enough to provide a good balance between results and efficiency.

Known the number of hypotheses k, the next step is to estimate the values of w_{ij}, θ_{ij} and σ_{ij} that better approximate the GMM of (3) to an Uniform Distribution between 0 and

2π. The value of the hypothesis weights is equal to all of them, and is set as $w_{ij} = 1/k$.

Estimating the mean value θ_{ij} of each hypothesis is also simple considering that they have to be uniformly distributed from 0 to 2π. Thus, depending on the number of hypotheses, the values of the mean will be defined as:

$$\theta_{ij} = 2\pi j/k, \qquad j = 0, \ldots, k-1 \tag{4}$$

The value of σ_{ij} employed in the filter is:

$$\sigma_{i0} = \sigma_{i1} = \ldots = \sigma_{i(k-1)} = 2\pi/(1.5k) \tag{5}$$

C. Incorporating measurements

Once the beacon has been initialized into the filter with the first range information, next measurements will be used to update the estimation of each hypothesis and also to refine the weights w_{ij} associated to them.

The measurements provided by the system are the distances of the robot to the set of nodes that are in communication range. Thus, let $\hat{\rho}_i$ be the measured distance from the robot to the beacon i and σ_i^2 the measurement error variance. Considering (1), the following measurement equation is applicable for each of the hypotheses j of beacon i:

$$\hat{\rho}_i = \sqrt{(x_i + \rho_i cos(\theta_{ij}) - x_r)^2 + (y_i + \rho_i sin(\theta_{ij}) - y_r)^2} \tag{6}$$

The question now is how to deal with the variance associated with the measurement, σ_i^2. A single measurement is available but it has to be applied to all the existing hypotheses for beacon i. Notice that the measurement cannot be simply applied to all the hypothesis separately because then the same information would be counted k times in the filter which finally would lead to the filter divergence. In [13], the solution to this problem is shown for the case the information comes from a unique source, as it is our case. It is stated that the correction of the estimate of a random variable by a set of measurement pairs $(\mathbf{z}, \mathbf{R}_{ij})$ is equivalent to the unique correction by $(\mathbf{z}, \mathbf{R}_i)$ if:

$$\mathbf{R}_i^{-1} = \sum_{j=0}^{k-1} \mathbf{R}_{ij}^{-1} \tag{7}$$

This means that the original information can be divided into k new measurements with the same mean and with covariances according to (7). Sharing the information according to the likelihood l_{ij} of each hypothesis is proposed in [13]. Thus, if we are able to compute a weight λ_{ij} proportional to the likelihood of each hypothesis of beacon i such as $\sum_{j=0}^{k-1} \lambda_{ij} = 1$, the measurement variance associated to each hypothesis could be computed as $\sigma_{ij}^2 = \sigma_i^2/\lambda_{ij}$. Then, once the likelihood l_{ij} of each of the hypotheses has been computed, it is normalized using the following expression to obtain the values of λ_{ij}:

$$\lambda_{ij} = l_{ij} / \sum_{j=0}^{k-1} l_{ij} \tag{8}$$

Algorithm 1 Build Measures and Update Weights $\{\{\hat{\rho}_i, \sigma_{i0}^2\}, ..., \{\hat{\rho}_i, \sigma_{i(k-1)}^2\}\} \leftarrow \{\{x_r, y_r, \mathbf{b}_i\}, \{\hat{\rho}_i, \sigma_i^2\}\}$

/*Compute likelihood of each hypothesis */
for $j = 1$ to k **do**
  $l_{ij} = p(\hat{\rho}_i | x_r, y_r, x_i, y_i, \rho_i, \theta_{ij})$
end for
/* Compute measurement variance of each hypothesis */
/* Update weight of each hypothesis */
for $j = 1$ to k **do**
  $\lambda_{ij} = l_{ij} / \sum_{j=0}^{k-1} l_{ij}$
  $\sigma_{ij}^2 = \sigma_i^2 / \lambda_{ij}$
  $w_{ij} = w_{ij} l_{ij}$
end for
Normalize weights w_{ij} such as $\sum_{j=0}^{k-1} w_{ij} = 1$

Following this procedure, all the range measurements are applied to the corresponding beacon hypotheses.

Finally, it is necessary to properly update the weight associated to each hypothesis, w_{ij}. The key idea is to make evolve the weights according to the closeness of the hypotheses with the real beacon position. For doing this, the likelihood is used again according with the following equation:

$$w_{ij} = w_{ij} l_{ij} \tag{9}$$

Later, the new weights are normalized.

The whole procedure is summarized in Algorithm 1 for the beacon i. Once the weights have been updated, all the measurement pairs $\{\hat{\rho}_i, \sigma_{ij}^2\}$ are arranged into the measurement vector and its covariance matrix (which is diagonal), and used to update the hypotheses into the Extended Kalman Filter (EKF) using the standard EKF updating equations. The conditional probability $p(\hat{\rho}|x_r, y_r, x_i, y_i, \rho_i, \theta_{ij})$ is modeled as a Gaussian distribution, with mean obtained evaluating eq. (6) at the current hypothesis j, and propagating the corresponding state covariances through the Jacobian of the cited equation.

D. Pruning hypotheses

The algorithm is completed with a rule to remove hyptoheses from the filter. Loosely speaking, two main rules are employed. Hypotheses with very low weights are eliminated from the filter. Moreover, hypotheses that are closer (in Mahalanobis sense) than a certain threshold are merged. More details can be found in [7].

III. Active sensing for WSN mapping

The benefit of using a mobile robot to estimate the position of a set of nodes is that its motion can be adapted in order to take the most informative actions. In one hand, from the set of possible motions of the robot, it should take those that allows to estimate the position of the nodes more accurately. On the other hand, the robot should try to avoid motions that decrease the observability of the node position.

Our robot uses a combination of behaviors: basically the robot tries to follow a given path as accurately as possible, but at the same time minimizing a combined cost related to obstacle avoidance, etc. The idea is to include, in the computation of the control commands, a cost related to the gain of information. This gain of information is estimated using the expected variation of entropy of the GMM that represents the likelihood of the node position.

A. Entropy-based active sensing strategy

A general measure about the information of a probability distribution is its entropy. The entropy H of a probability distribution $p(x)$ is defined as the expected value of the information $-\log[p(x)]$:

$$H(p(x)) = E_x[-\log p(x)] = -\int p(x) \log p(x) dx \quad (10)$$

The information gain is defined as the variation in the entropy of the distribution after carrying an information gathering action $\mathbf{u}_t$. When executing this action, a new distribution $p(\mathbf{x}_{t+\Delta t}|\mathbf{u}_t, \mathbf{z}_{t+\Delta t})$ will be obtained from the future measurement $\mathbf{z}_{t+\Delta t}$, by using the filtering algorithm described in Section II. The entropy of this new distribution will be denoted by $H(p(\mathbf{x}_{t+\Delta t}|\mathbf{z}_{t+\Delta t}, \mathbf{u}_t))$.

However, only the action $\mathbf{u}_t$ can be controlled. Then, we should take the expectation of the entropy for all potential measurements $\mathbf{z}_{t+\Delta t}$ that can be obtained after executing the action. Therefore, the (expected) information gain associated to action $\mathbf{u}_t$ is defined as follows:

$$\Delta(\mathbf{u}_t) = H(p(\mathbf{x}_t)) - E_{\mathbf{z}_{t+\Delta t}}[H(p(\mathbf{x}_{t+\Delta t}|\mathbf{z}_{t+\Delta t}, \mathbf{u}_t))] \quad (11)$$

This metric can be used to establish preferences among actions, favoring those that maximize the value $\Delta(\mathbf{u}_t)$.

B. Entropy of a Gaussian Mixture

The entropy, as defined in equation (10), can be obtained analytically for certain distributions, including the Gaussian distribution. However, there is no analytical solution for the case of Gaussian Mixtures, defined by eq. (3).

One option is to numerically integrate (10), for instance using Monte Carlo methods. However, this is computationally demanding, as a high number of samples may be required (the accuracy depends on the number of samples). The proposed approach uses upper bounds of the entropy as an approximation to the actual entropy value. Thus, instead of analyzing the expected variation in entropy for a particular action, the expected variation of the entropy bound will be considered.

In [14], Huber *et al.* derive analytical approximations to the entropy of a Gaussian mixture; moreover, some analytical upper and lower bounds of the entropy of a Gaussian Mixture are presented as well. Among them, the following expression gives an upper bound of the entropy of a Gaussian Mixture $f(\mathbf{x}) = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, \Sigma_i)$, which is very cheap to compute:

$$H(f(\mathbf{x})) \leq \sum_i \omega_i(-\log \omega_i + \frac{1}{2}\log((2\pi e)^N|\Sigma_i|)) \quad (12)$$

Algorithm 2 $\Delta(\phi_k) \leftarrow$ active_searching($p(\mathbf{x}_t)$, Δt)

1: $\Phi = \{\phi_1, \cdots, \phi_k, \cdots\}$ A set of L orientations
2: $H_t \leftarrow$entropy_bound $(p(\mathbf{x}_t))$
3: **for all** $\phi_k \in \Phi$ **do**
4: $\quad \mathbf{r}_{t+\Delta t} \leftarrow$ predict_robot $(\mathbf{r}_t, \phi_k, \Delta t)$
5: $\quad$ **for all** (μ_i, Σ_i) in $f(\mathbf{x}_t)$ **do**
6: $\quad\quad \mathbf{z}_{t+\Delta t} \leftarrow$ simulate_measurement($\mathbf{r}_{t+\Delta t}, \mu_i, \Sigma_i$)
7: $\quad\quad p(\mathbf{x}_{t+\Delta t}|\mathbf{z}_{t+\Delta t}) \leftarrow$ update $(p(\mathbf{x}_t), \mathbf{z}_{t+\Delta t})$
8: $\quad\quad H_{i,k} \leftarrow$ entropy_bound $(p(\mathbf{x}_{t+\Delta t}|\mathbf{z}_{t+\Delta t}))$
9: $\quad$ **end for**
10: $\quad H(\phi_k) = \sum_i \omega_i H_{i,k}$
11: $\quad \Delta(\phi_k) = H_t - H(\phi_k)$
12: **end for**
13: Normalize $\Delta(\phi_k)$

with N the dimension of $\mathbf{x}$.

Moreover, this bound is exact when only one hypothesis remains, or when the hypotheses are separated.Therefore, a possible strategy is to compare actions taking into account how they affect not the entropy itself, but the upper bound. While in theory a decreasing in the bound could not reflect on a decreasing of the actual entropy, in the experiment section it will be seen that the procedure is effective reducing the actual entropy of the distributions.

C. Active sensing controller

The robot considered here is a non-holonomic electric car. The controlled variables are the linear velocity v and the steering angle ϕ of the vehicle. The car uses a combination of distributed behaviors communicating with a centralized arbiter by sending votes in favor of actions that satisfy its objectives (as in [15]). This combination associates a set of weights for all the potential contributions of the different behaviors.

Algorithm 2 shows the strategy to compute the votes associated to the active sensing behavior. Only the steering angle will be considered, which is discretized into a set of L orientations $\{\phi_1, \cdots, \phi_L\}$. For each potential angle ϕ_k, it is possible to predict the future position of the robot for a certain time horizon Δt. At that future position, the potential range measurements to the known nodes are considered. The basis of the algorithm is given by lines 6, 7 and 8. Within the for loop, each hypothesis within the Gaussian mixture about the position of the known nodes is considered correct, and a measurement $\mathbf{z}_{t+\Delta t}$ is simulated for that hypothesis at line 6. Then, the filter described in Section II is applied by the function `update` to estimate the future belief, and the upper bound of the entropy is computed.

The final expected upper bound is computed as the mean of these upper bounds, taking into account the weight ω_i associated to each hypothesis. That is, taking the expectation with respect to all the potential measurements, which corresponds to the second term of the right hand side of (11).

Although not depicted in Algorithm 2, the final algorithm applies the same operation for all the currently known

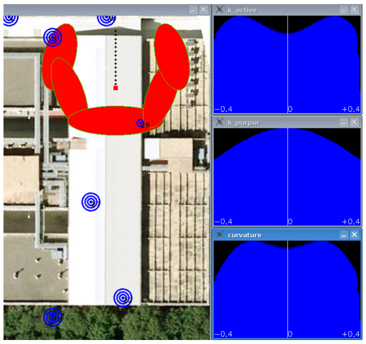
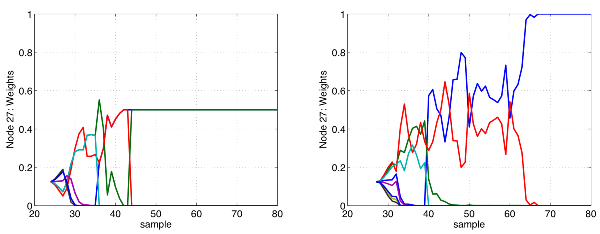


Fig. 2: Behaviors combination. On the left, it can be seen a simple graphical representation of the hypotheses (ellipses) and the robot position (red small square). On the right, it can be seen the votes on the steering angle imposed by the active sensing (top) and path tracking (middle) behaviors, and the resultant combination. The active sensing behavior presents two maxima that will deviate the robot from the straight line path.



(a) (b)

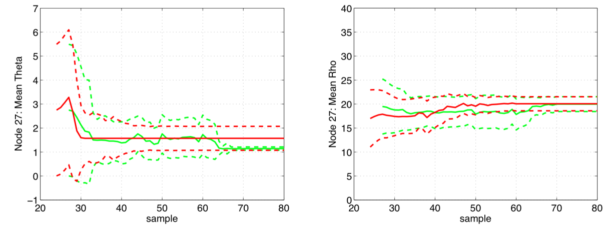
Fig. 3: Values of the weights associated to the hypotheses of node 27. The weight value range from 0 to 1. (a) No active controller. (b) Active contoller is considered. Notice how usually after some measuremens only two hypotheses remain. The active controller is able to converge to a single solution.

beacons that are within communication range. Therefore, the final vote $\Delta(\phi_k)$ for a particular angle is the sum of the variations of the entropy for each of these beacons.

The final votes for all steering angles are normalized. These votes are then combined with the votes indicated by other behaviors. Figure 2 shows an example of particular interest. It shows how the strategy not only can lead to reductions on the uncertainty, but also to avoid non-observable motions, like straight lines. In this example, it can be seen how there are two symmetric entropy variation maxima when computing the active sensing votes; both of them are far from the steering angle of 0 degrees that would maintain the car on a straight line.

IV. Results

A set of simulations has been carried out in order to test the approach for active mapping using range-only information. The setup consists of a car-like mobile robot equipped with a wireless sensor node. The robot moves in an area in



(a) (b)

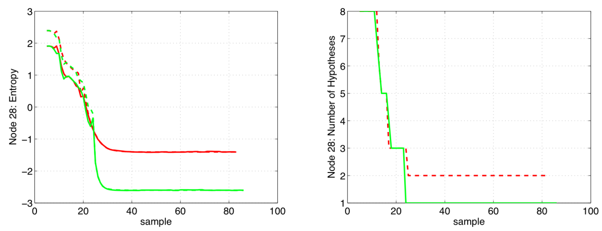
Fig. 4: (a) Estimated value of θ (solid line) and its 3σ confidence interval (dashed) for node 27. (b) Estimated value of ρ (solid line) and its 3σ confidence interval (dashed line) for node 27. The active sensing strategy in green, and the non-active in red.

which a wireless sensor network composed by other fifteen static nodes has been deployed. The robot is assumed to be localized with a certain accuracy. Each time a message arrives at the sensor node onboard the robot, the distance to the emitter is calculated based on the Received Signal Strength Information (RSSI), and this information is used to update the node position into the filter. In the simulation, the signal propagation model described in [16] has been used to generate random samples of the distance between the robot and the sensor node. The maximum transmission range and the rate of messages sent by each node of the network (about one per second) have been also considered into the simulation to be as realistic as possible. The robot is commanded a predefined trajectory.

The simulations are carried out using the active controller and without the active controler. The same path tracking behavior is considered in both cases. Fig. 3 shows the typical evolution of the weights associated to each of the hypotheses related to a node (node 27) of the WSN. Three main stages can be seen in the evolution of the hypotheses' weights: First, most of the hypotheses are removed quickly after the integration of the new measurements. Later, another hypothesis is removed and only two possible solutions remain in the filter. In this case, the active controller moves the robot in favor of a better triangulation and the wrong hypothesis can be removed from the filter, converging to a single solution of the node localization.

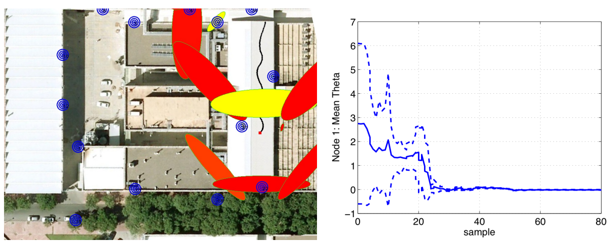
For the case of node 27, Fig. 4 shows the estimated mean and standard deviation of θ and ρ. It can be seen how the estimated θ slowly converges to the correct solution together with the estimated range ρ. The final uncertainty in ρ is due to the noisy distance measurements obtained from the RSSI. The active strategy is able to discard all but one of the hypotheses regarding θ.

Figure 5 shows the evolution of the entropy of the GMM, and the number of hypotheses for a different node. The entropy is estimated by Monte Carlo integration, using 10000 samples. Moreover, the evolution of the entropy bound (12) is also shown. Interestingly enough, the simulations show how the entropy bound converges to the actual entropy value



(a) (b)

Fig. 5: (a) Estimated value of the entropy (solid lines) and entropy bounds (dashed lines) for Node 28. (b) Evolution on the number of hypothesis for Node 28. The active sensing strategy in green, and the non-active in red.



(a) (b)

Fig. 6: (a) Trajectory and estimations on the nodes for a real experiment. (b) Estimated value of θ (solid line) and its 3σ confidence interval (dashed) for node 1.

when the number of hypotheses is reduced. Therefore, it is possible to use the variations on the bound for active sensing.

Finally, Figure 6 shows the results obtained in a actual experiment using our autonomous car Romeo. In this experiment, the nodes are simulated, while the active controller is running on a laptop (1 GB of RAM, 1,2 GHz processor) and actually controlling the real robot. The figure shows the actual trajectory followed, and the estimated angle θ for one the simulated nodes in the environment. The active sensing controller was running at 20 Hz with a temporal horizon (Δt in Algorithm 2) of 5 seconds.

V. Conclusions and Future Work

This paper presented a Gaussian Mixture approach to solve the mapping problem in presence of radio signal-based range-only measurements. The Mixture Model can be integrated into a Kalman Filter, leading to a multiple hypotheses filter able to deal with the range-only mapping problem, and allowing undelayed use of the information.

The main contribution of the paper is an active sensing strategy to control the mapping robot, in order to select the most informative control actions. Variation in entropy is considered as a measure of information gain. However, as there is no analytical expression for computing the entropy of a GMM, entropy bounds are used instead to obtain an estimation of the information gain. Simulations show that the approach is feasible for localizing wireless sensor nodes based on range measurements. The simulations also show that this procedure is effective producing efficient motions for node mapping. Moreover, the controller has been tested in a real robot.

Actual results with a WSN of Mica2 nodes will be carried out in the short future. Moreover, future research will consider an analytical study of the non-observable motions that can arise in this problem; for instance, pure straight lines are non-observable motions (in the sense that two hypothesis can generate the same sequence of measurements and robot positions). The active sensing strategy presented allows to avoid this, but a rigorous theoretical study is required.

References

[1] A. M. Ladd, K. E. Bekris, G. Marceau, A. Rudys, D. S. Wallach, and L. E. Kavraki, "Using wireless Ethernet for localization," in *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2002*, vol. 1, Lausanne, Switzerland, September 2002, pp. 402–408.

[2] D. Kurth, G. A. Kantor, and S. Singh, "Experimental results in range-only localization with radio," in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '03)*, vol. 1, October 2003, pp. 974 – 979.

[3] R. Moses, D. Krishnamurthy, and R. Patterson, "A self-localization method for wireless sensor networks," *EURASIP Journal on Applied Signal Processing*, no. 4, pp. 348–358, 2003.

[4] E. Menegatti, A. Zanella, S. Zilli, F. Zorzi, and E. Pagello, "Range-only slam with a mobile robot and a wireless sensor networks," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, 2009, pp. 8–14.

[5] F. Caballero, L. Merino, I. Maza, and A. Ollero, "A particle filter method for wireless sensor network localization with an aerial robot beacon," in *Proceedings of the International Conference on Robotics and Automation*, Pasadena, CA, USA, May 19-23 2008, pp. 596–601.

[6] J. Djugash and S. Singh, "A robust method of localization and mapping using only range," in *International Symposium on Experimental Robotics*, July 2008.

[7] F. Caballero, L. Merino, and A. Ollero, "A general gaussian-mixture approach for range-only mapping using multiple hypotheses," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, Anchorage, Alaska (USA), May 2010, pp. 4404–4409.

[8] J. Djugash, S. Singh, and B. P. Grocholsky, "Decentralized mapping of robot-aided sensor networks," in *IEEE International Conference on Robotics and Automation*, May 2008.

[9] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte, "Information based adaptive robotic exploration," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2002, pp. 540–545.

[10] C. Stachniss, G. Grisetti, and W. Burgard, "Information-gain based exploration using rao-blackwellized particle filters," in *Proceedins of Robotics: Sciene and Systems*, 2005.

[11] E. Stump, V. Kumar, B. Grocholsky, and P. Shiroma, "Control for localization of targets using range-only sensors," *International Journal of Robotics Research*, vol. 28, no. 6, pp. 743–757, 2009.

[12] K. Zhou and S. Roumeliotis, "Optimal motion strategies for range-only constrained multisensor target tracking," *IEEE Transactions on Robotics*, vol. 24, pp. 1168–1185, 2008.

[13] J. Sola, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only slam," in *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, August 2005, pp. 2499–2504.

[14] M. F. Huber, T. Bailey, H. Durrant-Whyte, and U. D. Hanebeck, "On entropy approximation for gaussian mixture random vectors," in *Multisensor Fusion and Integration for Intelligent Systems, MFI. IEEE International Conference on*, 2008, pp. 181–188.

[15] J. Rosenblatt, "Damn: A distributed architecture for mobile navigation," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, January 1997.

[16] F. Caballero, L. Merino, P. Gil, I. Maza, and A. Ollero, "A Probabilistic Framework for Entire WSN Localization Using a Mobile Robot," *Journal of Robotics and Autonomous Systems*, vol. 56, no. 10, pp. 798–806, 2008.