Journal of Intelligent and Robotic Systems **34:** 135–154, 2002.

A Genetic Algorithm for Mobile Robot Localization Using Ultrasonic Sensors

LUIS MORENO, JOSE M. ARMINGOL, SANTIAGO GARRIDO,
ARTURO DE LA ESCALERA and MIGUEL A. SALICHS
*Departament of Systems Engineering and Automation, Universidad Carlos III de Madrid,
Avda. de la Universidad 30, 28911 Leganés, Madrid, Spain;
e-mail: (moreno,armingol,sgarrido,escalera,salichs)@ing.uc3m.es*

(Received: 14 March 2001; in final form: 23 July 2001)

Abstract. A mobile robot requires the perception of its local environment for position estimation. Ultrasonic range data provide a robust description of the local environment for navigation. This article presents an ultrasonic sensor localization system for autonomous mobile robot navigation in an indoor semi-structured environment. The proposed algorithm is based upon an iterative non-linear filter, which utilizes matches between observed geometric beacons and an *a-priori* map of beacon locations, to correct the position and orientation of the vehicle. A non-linear filter based on a genetic algorithm as an emerging optimization method to search for optimal positions is described. The resulting self-localization module has been integrated successfully in a more complex navigation system. Experiments demonstrate the effectiveness of the proposed method in real world applications.

Key words: mobile robots, localization, ultrasonic sensors, genetic algorithms.

1. Introduction

A prerequisite for geometric navigation of a mobile robot is a position-finding method. A rough estimation of the displacements of the robot (position and orientation) is possible with odometry data. To reduce the uncertainty and the error of the estimated location, re-localization techniques have been developed. Unfortunately, location estimates are typically very sensitive to the event-source attribution. This is particularly true of Kalman filter-based systems. These are very effective at tracking moving parameters as long as their hypotheses are met, but they are extremely susceptible to outliers and systematic model deviations because the Kalman filter update rule depends critically on the assumed error distributions.

Integrated localization systems are absolute positioning methods; which require external absolute references in order to estimate the position and the orientation. This technique estimates the absolute position and orientation of the robot by measuring the direction of incidence or the distances to several signals transmitted or reflected by artificial beacons. The locations of the beacons in the environment have to be known *a priori*. These methods have been used for ship navigation for many years as a very reliable localization method. Traditionally, two approaches are used in mobile robot localization: the matching between certain geometric features such as walls or corners observed by the sensorial system (widely used with ultrasonic sensors, infrared sensors and laser range finders) and a map of the environment, or the detection of natural or artificial landmarks (used mainly with cameras).

A landmark is a localized physical feature that the robot can sense and use to estimate its own position in relation to a map that contains the landmark's absolute position. D'Orazio *et al.* (1994) use a landmark that consists of panels on which two groups of four LEDs were fixed. The camera can see these confidence points at any moment and they can be located easily in the image plane because of their brightness and arrangement. In (Betke and Gurvits, 1997), an efficient method for localizing a mobile robot in an indoor environment is described. The robot can identify the landmarks and measure their bearings relative to each other. Ricotti and Liotta (1995) have used a guide mark (two circles and identification codes) that can be easily characterized by geometric features, which are invariant with respect to scale factors and perspective views. Han and Rhee (1994) use a solid circle in black with two interior white dots to obtain the position of the mobile robot. Salichs *et al.* (1999) present a localization system, which use visual detection of natural and artificial landmarks for re-localization in indoor environments. Artificial landmarks are solid circles placed on the walls and natural landmarks are nameplates fixed at the entrance of each room.

In (Escalera *et al.*, 1996), a combination of a laser diode and a CCD camera is used; the sensorial information is modelled as straight lines that are matched with a prior map of the environment. The matching process is accomplished within an extended Kalman filter. In (Triggs, 1994), a sonar localization system for mobile robot navigation in a known environment is described, where state corrections are applied through a standard Kalman filter. In (Leonard and Durrant-Whyte, 1992), an application of the extended Kalman filter to the problem of mobile robot navigation in a known environment is presented. This navigation algorithm is based on an extended Kalman filter, which matches observed geometric beacons and an a priori map of beacons locations.

This article describes an algorithm based upon an iterative extended Kalman filter, which utilizes matches between observed geometric beacons and an *a priori* map of beacon locations, to correct the position and orientation of the vehicle. A non-linear filter based on a genetic algorithm is presented as an emerging optimization method to search for optimal positions.

Genetic algorithms (GAs) have proven to be effective in different mobile robot applications. In (Chen and Zalzala, 1995) a genetic approach to mobile robot motion with a distance-safety criterion is presented. Hein and Meystel (1994) have developed a GA that finds admissible control trajectories that tend to minimize the vehicle's transit time through the obstacle field. Other applications include the planning of control and trajectories for automated delivery vehicles and the optimization of control for racing vehicles. In Potvin *et al.* (1996) a route construction heuristic for a vehicle routing problem is described. Kang *et al.* (1995) present a GA for global path planning to a goal for a mobile robot in a known environment.

2. Robot Localization Problem

When trying to estimate the state of the robot at the current time-step k, we assume there exist knowledge about the initial state, all the measurements $Z^k = \{z_k, i = 1, \dots, k\}$ up to the current time, the system and the observation models. Due to the fact that the system model and the observation model are corrupted with noise, some kind of state estimation method needs to be used.

$$\begin{aligned} x(k) &= g(k, x(k-1), \varepsilon(k)), \\ z(k) &= h(k, x(k), \eta(k)), \end{aligned} \tag{1}$$

where the first function is the state equation, which represents the model of the system. The function $g(k, x(k-1), \varepsilon(k))$ is called the estimator and the value of this function is the estimate. The second one is the measurement equation and it permits the comparison of the model results with the real ones.

This estimate permits integrate all the previous knowledge about the system with the new data observed by the sensors in order to obtain a precise state estimation of the real state of the system. Due to the stochastical nature introduced by sensors and system noise, it is convenient to use a probabilistic formulation of the estimators to reach reliable values of the state estimate. This estimation problem can be formulated as a Bayesian filtering problem, where the *posterior probability density* $p\{x_k|Z^k\}$ of the current state is constructed taking into account all measurements up to the current time. This density is recursively computed in two steps:

– *Prediction.*
  In this phase the system model is used to predict the state of the system in the next cycle $\hat{x}(k|k-1)$ and to predict the conditional probability density function $p(x_k|Z^{k-1})$.

– *Updating.*
  In the second phase, the measurement model is used to integrate new information from the sensors to obtain the updated state $\hat{x}(k|k)$and the posterior probability density function $p(x_k|z_k, Z^{k-1})$. It is assumed that the new measure z_k is conditionally independent from the previous measurements Z^{k-1}given x_k, and that the measurement model is given in terms of $p(z_k|x_k)$which expresses the likelihood that the robot is at position x_k given that z_k is observed. The posterior density $p(x_k|z_k, Z^{k-1})$ is obtained using the Bayes theorem:

$$p(x_k|Z^k) = \frac{p(z_k|x_k)p(x_k|Z^{k-1})}{p(z_k|Z^{k-1})}, \tag{2}$$

  where $p(z_k|Z^{k-1})$ is defined as:

$$p(z_k|Z^{k-1}) = \int p(z_k|x_k) \cdot p(x_k|Z^{k-1})\, \mathrm{d}x_k. \tag{3}$$

An initial estimate of the state x_1 and the probability density function $p(x_1|Z^0) = p(x_1)$ are necessary for the algorithm to function properly.

3. Existing Approaches

The solution to the robot localization problem is obtained by solving Equations (2) and (3). Depending on the method of representing the probability density function $p(x_k|Z^k)$ different classes of estimators are obtained.

The most traditional methods use the maximum *a posteriori* or the mean as estimator of the state. These approaches work quite well when the probability distribution function is uni-modal. In the case where the probability distribution function is multi-modal this way of estimating the position of the robot is not appropriate. This problem is clearly present when the mobile robot tries to localize its initial position in a global environment model. There exists a high potential number of symmetries in the environment, specially when the information provided by sensors is not sufficient or the *a priori* environment model does not match reasonably the real world. In this situation the probability distribution function is distributed around the different acceptable positions, and can not be considered as uni-modal.

This problem can be addressed in different ways. One possibility consists of increasing the number of features that can be observed by the robot sensors at each possible position in order to decrease or even eliminate the number of possible symmetries. In this form, the assumption of uni-modality in the PDF can be reasonably accepted. The second way of coping with this problem is by using filtering methods where the uni-modal assumption is removed. This is the approach this work uses and it is presented in the next sections. Before presenting the Restricted Genetic Optimization (RGO) method, it is convenient to consider the existing methods of obtaining the *posterior density* $p\{x_k|Z^k\}$ which lead us to substantially different methods with very different properties.

3.1. THE EXTENDED KALMAN FILTER (EKF)

If the equations of the model and the observation are linear and the noise terms are Gaussian, then the least square estimate can be calculated using a Kalman filter (Bar-Shalom *et al.*, 1988). This filter provides us with an optimal estimate from a statistical point of view. The Kalman filter method approach assumes that if both the system and the measurement models can be described using a Gaussian probability density function, and the initial state is also specified as a Gaussian, then the density $p(x_k|Z^k)$ will remains Gaussian. Due to this assumption, the $p(x_k|Z^k)$ can be represented in a very compact way by means of only two parameters (the mean and the covariance matrix of the probability distribution) which leads us to a closed and elegant form of obtaining the state and covariance estimates of the robot positions.

The Kalman filter estimate can be done recursively, based on the new measurement and the last estimate of the state in order to estimate the current state and to predict the state in the next cycle. This recursive solution avoids the necessity of storing previous estimates and measures, which improves greatly the speed and computational requirements of the algorithm.

When the system model has a non-linear dynamics or a non-linear observation model the simplest approach is to expand the model and observation equations in Taylor's series and use the first term of the series into the recursive Kalman filter. This lead us to the Extended Kalman Filter (EKF).

In the linear case, the filtering problem at the state space is given by

$$\begin{aligned} x(k) &= g(k, x(k-1)) + \varepsilon(k), \\ z(k) &= h(k, x(k)) + \eta(k), \end{aligned} \tag{4}$$

where $\varepsilon(k)$ and $\eta(k)$ are the noise at the model and observation, which are supposed to be independent random vectors with a Gaussian distribution of zero mean and a known covariance:

$$\begin{pmatrix} \varepsilon(k) \\ \eta(k) \end{pmatrix} = \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} Q(k) & 0 \\ 0 & R(k) \end{pmatrix} \right). \tag{5}$$

The optimal estimate (the minimum mean square estimate) is evaluated considering the conditional probability density.

$$\hat{x}^{MMSE}(k|k) = E(x(k)|Z^k) = \int x(k)p(x(k)|Z^k)\,\mathrm{d}x. \tag{6}$$

The extended Kalman filter is similar to a linearized Kalman filter with the exception that the linearization is done around the estimated trajectory instead of a previously calculated nominal trajectory. For this reason, the functions $g(k, x(k-1), \varepsilon(k))$ and $h(k, x(k), \eta(k))$ are developed in Taylor's series around $\hat{x}(k|k)$ with first- or second-order terms to obtain the EKF or Second Normal Form (SNF) filters.

A considerable number of researchers have used approaches based on extended Kalman filters to correct the position and orientation of the vehicle using the error between the observed and estimated distances between the robot and the segments of the environment (walls, doors, etc.).

3.2. MONTE CARLO LOCALIZATION

The Gaussian approximation to the density of the underlying state done in the EKF, will often prove to be inadequate, causing the update procedure to become unstable (Aidala, 1979; Kalos and Whitlock, 1986). In addition, the linearization presents problems that tend to appear in the early states of tracking. These linearization errors tend to be self-perpetuating problems and can prevent an accurate estimate ever being found. To avoid this problem, a good knowledge of the initial means and covariances is needed, but in practice such information is frequently unavailable.

The Monte Carlo localization methods are an alternative class of filters called sampling-based methods in which theoretical probability distributions on the state space are approximated by simulated random measurements. In those methods the probability density function $p(x_k|Z^k)$, where x_k indicates the state at time k, is represented by a set of N randomly sampled points called *particles* $S_k = \{s_k^i; i = 1, \dots, N\}$. Based on the samples it is possible to approximate the probability density (Dellaert *et al.*, 1999; Thrun *et al.*, 2001).

The key point in this group of methods is how to calculate recursively at step k the set of particles S_k based on $p(x_k|Z^k)$.

4. Limitations of Existing Methods

Techniques based on EKF filters have demonstrated to be robust to keep the track of the robot position in a precise way, but they have not been able to solve satisfactorily ambiguities (like symmetries). Moreover there are not able to re-localize the robot in case of localization failure or an initial global localization. Most of these difficulties derive from the assumption of Gaussian constraints in the probability distributions considered in the Kalman filter. Observing carefully the assumptions done when one apply Kalman filtering techniques to the mobile robot localization problem, some weaknesses are noticed:

- The assumption of a Gaussian noise in the system model is not completely true. The errors in the motion model can be originated from mechanical aspects in the design of the robot (transmissions, gears, wheels, encoders) which are systematic or can be originated by slippage of the robot on the floor and then can be considered non-systematic. An important part of the systematic errors can be taken into account through a properly calibration process, but the non-systematic errors can not so easily eliminated (Borenstein and Feng, 1996; Chong and Kleeman, 1997). Depending on the floor conditions and the motion done by the robot the slippage can be substantially different from the Gaussian assumption, and it may be inaccurate.
- The assumption of Gaussian error in the sensor measurements is optimistic in some kind of sensors. In case of using ultrasonic sensors to observe the environment, additional distortions are introduced because of the lobe in the ultrasound sensor which provide us the closer obstacle distance. If the angle between the sensor axis and the surface is not appropriate an erroneous distance is obtained. One can argue that this closer obstacle distance tends to be Gaussian. However, a small change in the observation angle has important effects on the obstacle perceivability. Besides, there is a bias in the observation, except in case of using a complete sensor observation model which includes the ultrasound lobe effect on the observation.
- The extended Kalman filter equations, which define the evolution of the Gaussian conditional probability density of the state, are obtained from a Taylor series expansion up to first-order terms of the filter. This introduces errors in

the equations since the higher-order terms are neglected. This filter can be considered as a class of deterministic estimators.

Kalman filtering techniques applied to mobile robot localization have proven to be robust and accurate for keeping track of the robot's position in relatively open areas. But when robot moves into cluttered environments or in narrow places and the sensorial systems is ultrasound this technique has accuracy limitations.

This techniques works well to keep track of the position if the initial position is well known, this happens because in this case the sensorial symmetries and ambiguities are *a priori* eliminated.

A modified state updating approach can be obtained by using an iterative procedure to optimize locally the error function. Looking at the expression of the measurement prediction:

$$\hat{z}(k+1|k) = h(k, \hat{x}(k+1|k)). \tag{7}$$

It can be noticed that there are prediction errors in using $\hat{x}(k|k)$ for $x(k)$ and also there are other additional errors that are due to measurement nonlinearity. It is possible to alleviate these errors if the updated state is not computed as an approximate conditional mean, but a maximum a posteriori estimate.

With respect to Monte Carlo filtering technique the main limitations comes from the high computational cost required and also from the relatively slow convergence of the algorithm, which restricts the applications of this solution to off line problems.

5. Restricted Genetic Optimization Filter (RGO)

In the literature, the use of Genetic Algorithms as a stochastic optimization method is traditionally done off-line because the computing time is usually considerably long. This high computational effort is due to two main reasons: the first one is that Genetic Algorithms are sampling-based methods and the second one is the difficulty of covering a global solution space with a limited number of samples.

The technique proposed in this paper tries to imitate the Nature: it works online. When Nature uses Genetic Optimization, it uses it locally, that is: at a given moment and to adapt to particular environmental conditions. For this reason, it is possible to achieve fast adaptation rate to changing conditions. It has been demonstrated (Garrido *et al.*, 1998; Garrido and Moreno, 2001), that Genetic Algorithms operating in restricted areas of the solution space can be a fast optimization method for time-varying, non-linear and non-differentiable functions. That is why, the technique proposed is called Restricted Genetic Optimization (RGO).

Usually, GAs are used as a parallel, global search technique. It evaluates many points simultaneously, improving the probability of finding the global optimum.

In Dynamic Optimization, finding the global optimum is useful for the first generations to find the correct basin of attraction. However, it consumes a large computation time. Therefore, a fast semi-local optimization method, such as RGO, is better.

The basic idea of this filtering technique is to combine the extended Kalman filtering approach, which has good properties due to the close form of the algorithm, and their recursive form of integrating the information with a simulation based filtering technique. The technique presented in this work uses the extended Kalman filter to obtain a seed, which is used to estimate a neighborhood where the true value of the state is located. With this information, and inside this neighborhood the most accurate solution is searched. The search of this solution is done stochastically employing a genetic search technique. The genetic optimization technique constitutes a probabilistic search method that avoids derivatives. The use of derivatives present two types of problems:

- Causes strong numerical oscillations when noise to signal ratio is high.
- Requires differentiable functions, because otherwise the derivatives can be discontinuous or even do not exist.

The numerical advantage of being a non gradient-based optimization method (which imitate the natural selection process based on genetic laws) has an important role in non linear systems with high noise to signal ratio.

The set of solutions (the population) is modified according to the natural evolution mechanism: selection, crossover and mutation, in a recursive loop. Each loop iteration is called generation, and represent the set of solutions (population) at this moment. The selection operator tries to improve the medium quality of the set of solutions by giving higher probability to be copied to next generation to the best solutions. This operator has a substantial significance because it focuses on the search of best solutions in the most promising regions of the state space. The quality of an individual solution is measured by means of the fitness function.

It is possible to adapt the genetic algorithm method (Goldberg, 1989) in order to focus the search in a certain neighborhood around the previous estimate by using as a fitness function:

$$f\colon B(\hat{x}(k|k-1), \sigma) \to \Re \tag{8}$$

with

$$\sigma = \|P(k|k-1)\| \tag{9}$$

and

$$\begin{aligned} V(x(k)) \;=\; & \tfrac{1}{2}(z(k) - h(k, x(k)))^T R(k)^{-1}(z(k) - h(k, x(k))) \\ & + \tfrac{1}{2}(x(k) - \hat{x}(k|k-1))^T P(k|k-1)^{-1}(x(k) - \hat{x}(k|k-1)). \end{aligned} \tag{10}$$

The global effect of the algorithm is to update the estimates of the state and the covariance matrix, and then the loop is repeated a number of times.

Each chromosome represents the difference with the best point of the last generation. New generations are oriented in the direction of the steepest slope of the

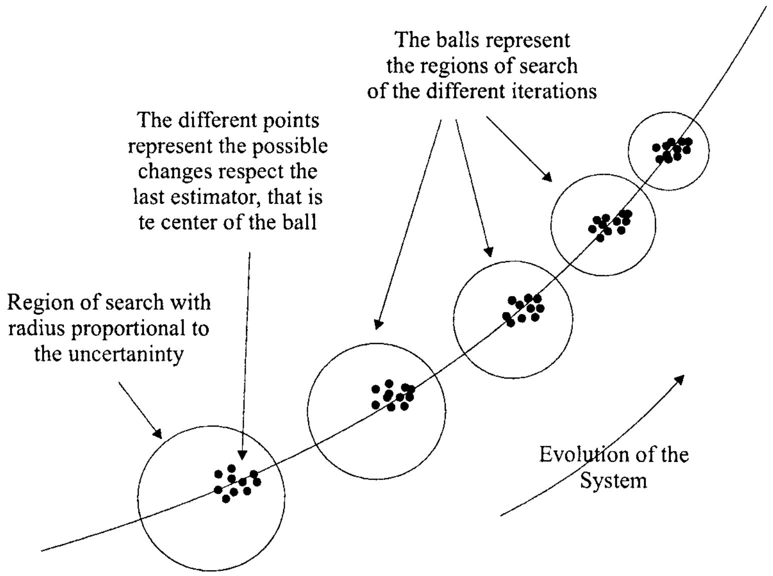

Figure 1. Restricted genetic optimization algorithm.

cost function (steepest descend), and have smaller distance to the correct estimate at each generation. This distance corresponds to the velocity at which the system is changing. This behavior simulates the gradient method without using derivatives and can be used even when signals are noisy.

It is possible to carry out a search in a big neighborhood at the beginning to explore a wide area of the solutions space, which permits to avoid a premature convergence of the algorithm to local minima. Once the space of solutions has been roughly explored, it is convenient to reduce progressively its radius (the radius is taken as proportional to uncertainty) to focus the algorithm to the most promising area of solutions. This improves the stability and accuracy of solutions near the optimum. In this way, the method makes a global search at first and a local search at the end reducing the probability of finding a local minimum and improving the stability and accuracy of solutions near the optimum. Similar effects are obtained in classical non-linear optimization methods by changing from Gradient Descend method to Quasi-Newton method when the algorithm approaches the solution (Dennis and Schnabel, 1996). The algorithm has the following steps:

- *1. Initialization:*

  The initial set of solutions is calculated and the fitness value associated to each of the points in the state space is evaluated.

- *2. Genetic Search:*

  The genetic operators: selection, cross mutation, elitism and immigration are used to build the new population (set of points in the state space), and the fitness function is evaluated for the new population. After a certain number of iterations, the best value of the whole population is selected as the most promising estimate. An important point to be considered is to determine the adequate population of solutions and the number of iterations to achieve good results. There is no an exact value, because it depends strongly on the dimensions of the state space, the variability of the system (noise to signal ratio) and the maximum response time required to be used on-line. For these reasons the populations and the number of iterations should be adapted to each specific problem. In the problem under consideration, the state space has dimension 3 and we have obtained good results with a population of 250 and a number of generations from 5 to 10.

– *Updating:*
  The best element of the population is used as the updated state and then used in an EKF to predict the new state. Then, a new Mahalanobis distance is evaluated to determine the new search radius for the next genetic search.

The steps of the algorithm are illustrated in Figure 2.

The maximization of the previous function done by the algorithm is equivalent to calculating a Maximum a Posteriori estimate (MAP). That is also equivalent to minimizing $V(x(k))$; that is to maximize the fitness function $f(J)$.

The standard fitness function (that is the fitness function divided by the sum of the fitness) is an approximation of the conditional probability density function (PDF).

$$p(x(k)|Z^k) = \frac{p(z(k)|x(k))p(x(k)|Z^{k-1})}{\int p(z(k)|x(k))p(x(k)|Z^{k-1})}. \tag{11}$$

To determine the radius of the search area, the Mahalanobis distance obtained from the EKF is used,

$$d = (\hat{x}(k|k-1) - \hat{x}(k-1|k-1)^T P^{-1}(k|k)(\hat{x}(k|k-1) - \hat{x}(k-1|k-1). \tag{12}$$

This distance provides information about the associated uncertainty of the linear estimate $x(k)$.

The most interesting characteristic of this approach is that it combines the advantages of the EKF with the interesting properties of the genetic search mechanism. The EKF is used to limit the search area (otherwise the genetic search becomes too costly from a computational point of view) and the genetic search area is used to improve the estimate (which is very interesting if the system is non-linear or the Gaussianity of the errors is not true or when the noise to signal ratio is high).

The results of the RGO filter shown in the next section have been obtained by using a population of 250 elements and 5–10 generations in the search of the best solution. The population size depends on the size of the area where the solutions are looked for, and can be decreased by iterating the algorithm more frequently if the sensors and the computational cost permit it.

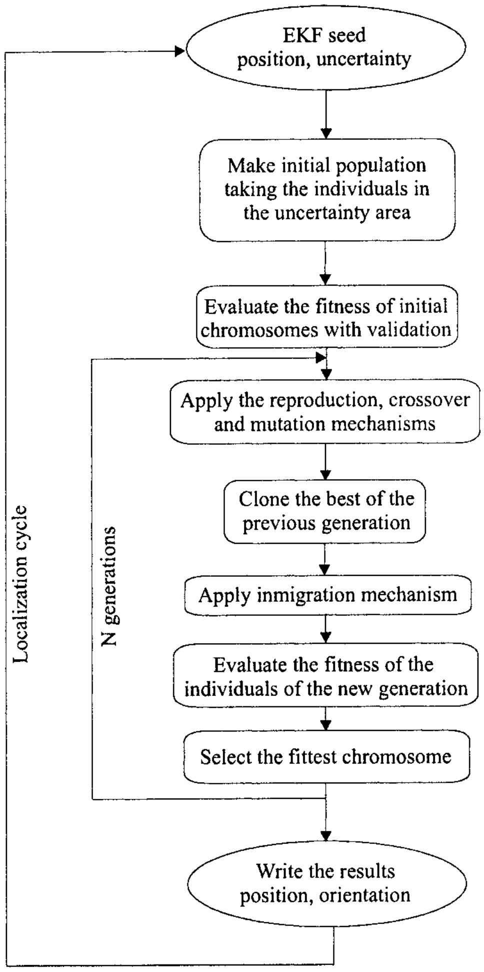
EKF seed
position, uncertainty
Make initial population taking the individuals in the uncertainty area
Evaluate the fitness of initial chromosomes with validation
Apply the reproduction, crossover and mutation mechanisms
Clone the best of the previous generation
Apply inmigration mechanism
Evaluate the fitness of the individuals of the new generation
Select the fittest chromosome
Write the results
position, orientation
Localization cycle
N generations

Figure 2. Restricted genetic optimization flowchart.

6. Experimental Results

All experiments have been developed in an indoor environment – University laboratories, offices and corridors (Figure 3), and conducted on a B21-RWI mobile vehicle, equipped with a peripheral ring of 24 Polaroid ultrasonic sensors.

The map of the environment is based on rooms and corridors modelled as segments. Segments of the environment are lines in 2-D, and are defined by the parameter vector $p_s = (O_s, D_s, \theta_s)$, where O_s is the origin of the local coordinate system. D_s is the length of the segment and θ_s is the angle with respect to the global coordinate system. A set of segments defines a specific room or a corridor.

The ultrasonic sensors from Polaroid Corporation have been adopted as a standard for most mobile robot platforms. There are several reasons for its popularity: robustness, range accuracy and low cost. However, this sensor presents some draw-

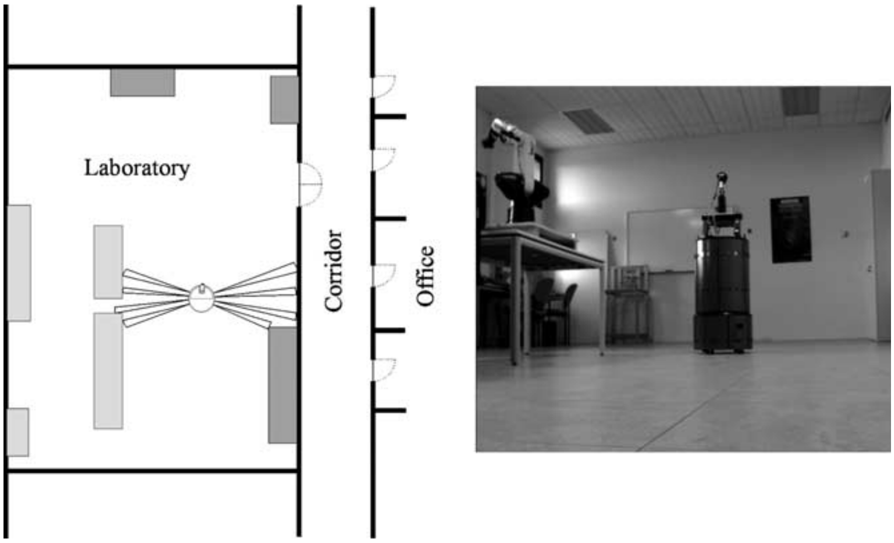

Figure 3. Experimental indoor environment.

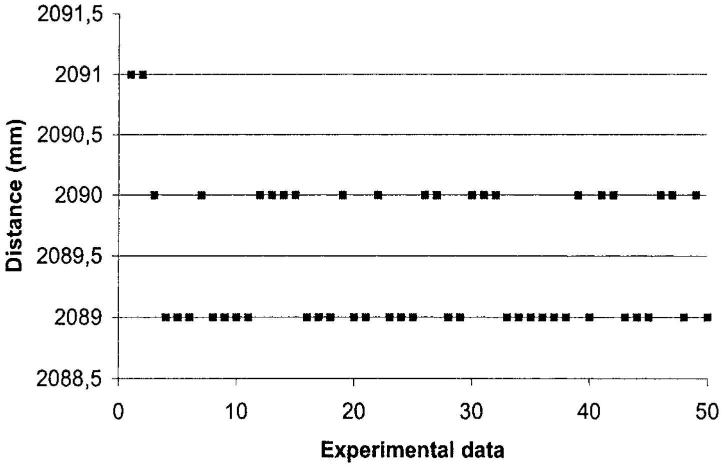

Figure 4. Ultrasonic sensor target.

backs such as reflection problems and low angular resolution. A detected target can reside anywhere inside a conical region with an angle of ± 15°. When the target is oriented at unfavorable angles, it may not be detected. As a result of these issues, a probabilistic model is required to capture the behavior of the sensor. Most of the actual implementations on mobile robots assume a Gaussian distribution.

A calibration step to determine ultrasonic sensor parameters is necessary in order to obtain accurate results. Figure 4 shows a detected target at 2250 mm along 50 measurement cycles; with a mean distance of 2089.5 mm and a variance of 0.335 mm.

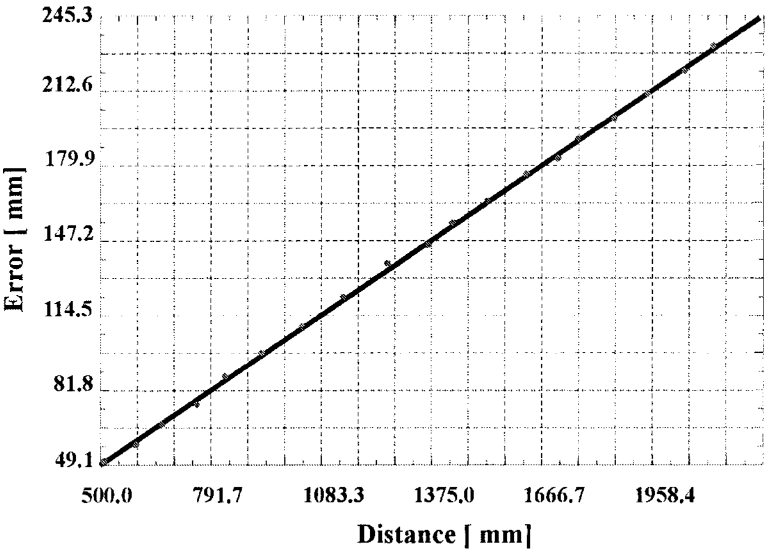

Figure 5. Linear model of the US error.

Figure 5 shows a linear model of the ultrasonic sensor errors working in a corridor (2.6 meters wide). The maximum range to be expected in an indoor environment is about 10 meters, with a mean error of 5 cm per meter (calibration results are used in the observation model).

To obtain the real position of the mobile robot, a Geodimeter 600 (Theodolite) was used. The accuracy of the distance measurement is ± 5 mm.

To test the localization method proposed in this article, different experiments were performed with the mobile robot stopped. The first test tries to determine the statical accuracy and the convergence speed of the proposed method. Initially, a position and orientation error was introduced in the location of the mobile robot, and then maintaining the mobile robot stopped the localization algorithm tries to reduce the position and orientation errors introduced artificially. Figure 6 shows the evolution of the position in the two first generations, with an initial error of 20 cm in position and 10° in orientation.

In Table I the evolution of the statical error at each generation of the RGO algorithm is shown (with 250 population elements and 5 iterations). In this experiment the number of ultrasonic sensors selected are eight and only one data acquisition is done. The convergence behavior of the algorithm is remarkable and it has great importance because a fast convergence in statical results facilitates a good dynamical error correction.

A second test to evaluate the effect of a statical localization with unexpected obstacles was developed. In this test there are some obstacles which occlude part of the expected environment and only 4 lateral sensors match the expected observations. To compare the accuracy, sensor measurements are integrated with EKF and RGO methods. Figures 7 and 8 depict the convergence of EKF and RGO methods

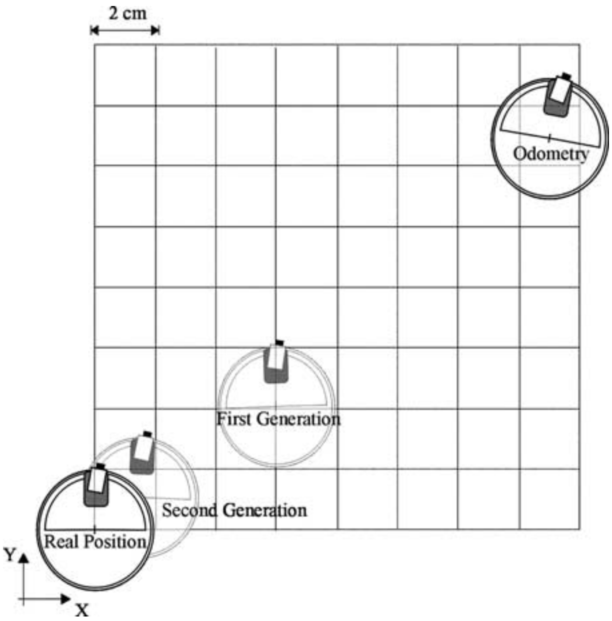

Figure 6. Localization evolution.

Table I. Evolution of the error

	Total error	*X* error	*Y* error	Heading error
Position 1 (Odometry)	198.5 mm	150 mm	130 mm	10°
Position 2 (Generation 1)	75 mm	60 mm	45 mm	2°
Position 3 (Generation 2)	27 mm	25 mm	10 mm	1.5°
Position 4 (Generation 3)	23 mm	21 mm	9 mm	1.1°
Position 4 (Generation 4)	18 mm	16 mm	8 mm	0.9°
Position 5 (Generation 5)	15 mm	14 mm	6 mm	0.6°

in position and orientation to the real values after some re-localization cycles. It can be noticed that the error after one statical re-localization is bigger than in the case of having 8 positive observations of the expected environment. Besides, a higher number of re-localization cycles are required to reach an accuracy similar to the full observation case. The lack of information affects to both methods similarly, but RGO method achieves considerably better accuracy than EKF method.

To test the validity of our method in a real dynamical situation, the mobile robot moved along two different typical indoor environments. In Figure 9, the mobile robot estimates its current position in a laboratory along a given path. In Table II a comparison between EKF and RGO errors at some intermediate points is shown. The results let us notice that the RGO method achieves systematically better

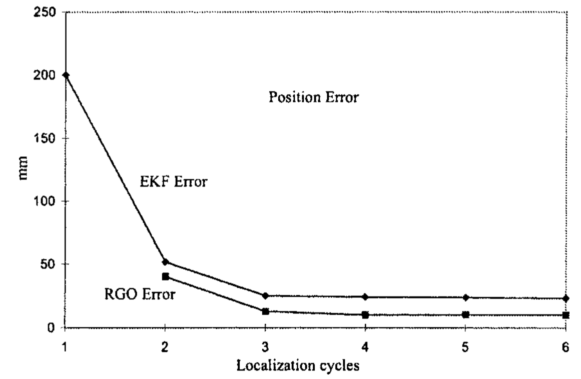

Figure 7. Position error evolution.

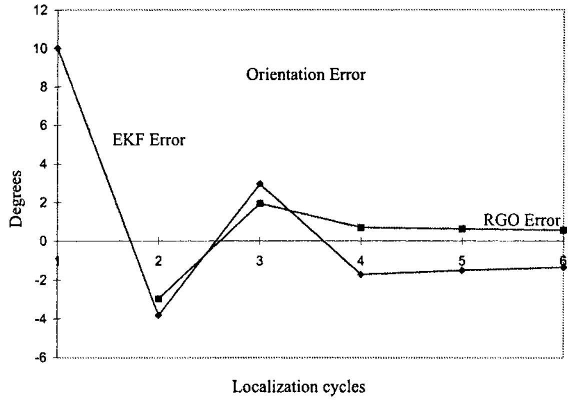

Figure 8. Orientation error evolution.

results than EKF. The improvement is of about 2 to 3 centimeters in the different dynamical test done. An RGO method with a population size of 250 elements and 5 generations has been used. The computational time is 0.1 s and 0.4 s for both algorithms respectively on a Pentium processor at 300 MHz.

The mean absolute error accumulated at the final position is about 3.5 cm in position and 1.0° in orientation with a population size of 250 elements and 5 generations. The results obtained with a population size 250 elements and 10 generations are 2.5 cm in position error and 0.5° in orientation error.

In the next figure, Figure 10, the mobile robot moves along a corridor and estimates its current position at some intermediate points. In Table III a comparison between results obtained with EKF an RGO algorithms, at some intermediate points of the trajectory, is shown. These results have been obtained with a population size of 250 elements and 10 generations. It can be noticed that the error has been reduced significatively compared with the error provided by the EKF method.

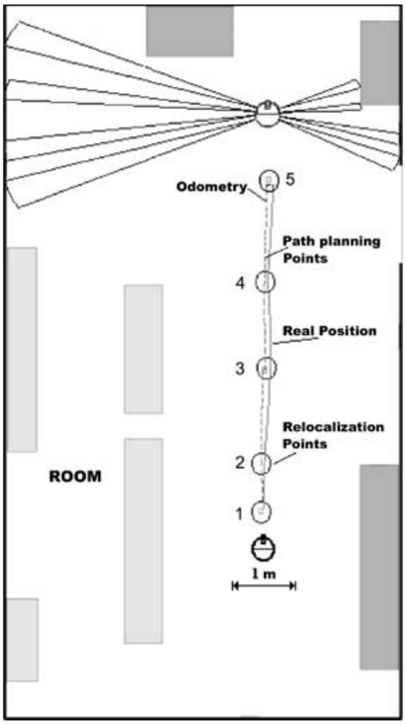

Figure 9. Travel path in a laboratory.

Table II. List of errors at each laboratory re-localization point

250,5	Sensors	EKF error (cm)	RGO error (cm)
Point 1	4	4.9	3.4
Point 2	4	5.5	3.8
Point 3	4	6.7	4.8
Point 4	5	7.3	4.5
Point 5	6	6.7	3.7

In Table IV the effect of the number of iterations executed by the RGO method in the accuracy at final position can be noticed. In both cases the results are clearly superior to the results of EKF method.

The re-localization points are marked with a continuous circle in Figures 9 and 10. At each point a different number of ultrasonic sensors is selected; re-localization process is developed with the vehicle moving at a maximum velocity set at 30–40 cm/s. The experiments were performed in a 12 × 7 m^2 room (Figure 9) and in a 20 m corridor (Figure 10).

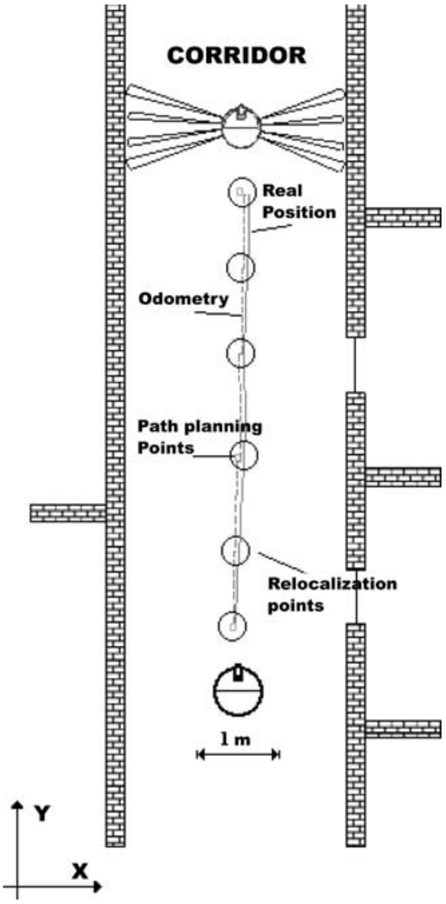

Figure 10. Travel path in a corridor.

7. Limitations of RGO

The most important limitations of the algorithm are related to the genetic nature of the stochastic search:

1. As most of sampling based methods the number of samples affect the accuracy of the estimate. With population sizes between 100 and 250 chromosomes, in problems with state space of dimension 3, good results have been founded. If the dimension of the state space increases the population required to obtain good results would be substantially augmented.
2. The convergence speed of the algorithm is considerably fast because the search area is restricted. In our tests the algorithm converges in around 3 to 5 generations and over 10 there are no significative improvement in the estimate.
3. Another important aspect to be considered is that the area where the solutions are looked up has to be bounded. We use a Mahalanobis distance with 95% of probability (but limited between a maximum and minimum values) to deter-

Table III. List of errors at each re-localization point at the corridor

250,10	Sensors	EKF error (cm)	RGO error (cm)
Point 1	6	5.7	3.9
Point 2	5	6.9	4.6
Point 3	4	7.6	4.8
Point 4	5	6.1	4.4
Final	5	6.7	3.8

Table IV. Effect of generation number on RGO errors at the corridor's final point position

	Final Error	*X* error	*Y* error	Orientation error
(250,10)	3.8 cm	1.5 cm	3.5 cm	0.5°
(250,5)	5 cm	2 cm	4.5 cm	1°

mine the radius of the sampling area and an EKF to estimate the center of the searching area.

4. The computational cost of the RGO method, for the problem took into consideration here, is approximately four times the computational time required for the EKF method to integrate 8 ultrasound sensor readings. In spite that the computational cost is higher than conventional EKF method, the RGO method can be executed on line. Monte Carlo filter with 5,000 sampling points is 20 times slower than RGO method and can not be implemented on line. Monte Carlo method is particularly slow because it estimate the probability density function, and this requires a high number of samples to be statistically efficient.

8. Conclusions

This article addresses the problem of localization for autonomous mobile robots. It has been shown that the effectiveness of traditional methods based on the use of EKF depends strongly on the assumption of the correctness of the knowledge about the error distribution (Gaussian and uni-modal). The practical difficulties to implement algorithms based on Monte Carlo techniques which do not require any assumptions about the error distribution have been commented.

To overcome these limitations, a new algorithm for mobile robot localization is proposed in this article. The new technique proposed here combines the use of an EKF to focus the area where the best estimate is looked for, and a genetic algorithm which operates in this restricted area to obtain the best estimated solution. The genetic search is iterated in the method to refine the final solution.

The algorithm has been implemented for the case where the environment is modelled in a semi-structured way and the robot knows an *a priori* map of it. A peripheral ring of 24 ultrasonic sensors provides the sensorial information; although the ideas of the algorithm can be implemented for other localization methods such as computer vision, laser range finder, etc.

The solution obtained with this method has been compared with those obtained with a classical EKF, and it shows a remarkable improvement in the results. The algorithm is computationally more expensive than the classical EKF, but less expensive than the Monte Carlo method. In addition, the proposed method has been successfully implemented and tested under real conditions in a B-21 mobile robot.

Acknowledgements

The authors gratefully acknowledge the funds provided by the Spanish Government through the CICYT projects TAP96-657 and TAP97-0296, and thank A. Novillo and M. Vázquez for their experimental work with the B21 robot.

References

Aidala, V. J.: 1979. Kalman filter behaviour in bearings-only tracking applications, *IEEE Transactions on Aerospace and Electronic Systems*, AES-**15**(1), 29–39, January.

Bar-Shalom, Y. and Fortmann, T. E.: 1988. *Tracking and Data Association*, Academic Press.

Betke, M. and Gurvits, L.: 1997. Mobile robot localization using landmarks, *IEEE Transactions on Robotics and Automation* **13**(2), 251–263.

Borenstein, J. and Feng, L.: 1996. Measurement and correction of systematic odometry errors in mobile robots, *IEEE Transactions on Robotics and Automation* **12**.

Chen, M. and Zalzala, A.: 1995. Safety considerations in the optimization of paths for mobile robots using genetic algorithms, in: *Proceedings of First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pp. 299–304.

Chong K. and Kleeman L.: 1992. Accurate odometry and error modelling for a mobile robot, in: *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, New Mexico, pp. 2783–2788.

Dellaert, F., Fox, D., Burgard, W., and Thrun, S.: 1999. Monte Carlo Localization for Mobile Robots, in: *Proceedings of the 1999 International Conference on Robotics and Automation*, pp. 1322–1328.

Dennis, J. E. and Schnabel, R. B.: 1996. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM Classics in Applied Mathematics.

D'Orazio, T., Lovergine, F. P., Ianigro, M., Stella, E., and Distante, A.: 1994. Mobile robot position determination using visual landmarks, *IEEE Transactions on Industrial Electronics* **41**(6), 654–662.

Escalera, A., Moreno, L., Salichs, M. A., and Armingol, J. M.: 1996. Continuous mobile robot localization by using structured light and a geometric map, *International Journal of Systems Science* **27**(8), 771–782.

Garrido, S., Moreno, L., and Salichs, M. A.: 1998. Non linear on line identification of dynamic systems with restricted genetic optimization, in: *Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing EUFIT*, pp. 423–428.

Garrido, S. and Moreno, L.: 2001. Learning adaptive parameters with restricted genetic optimization, in: *Bio-inspired Applications of Connectionism: 6th International Work-Conference on Artifical and Natural Neural Networks, IWANN2001*, Springer-Verlag, pp. 612–620.

Goldberg, D. E.: 1989. *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison-Wesley.

Han, M. and Rhee, S.: 1994. Navigation control for a mobile robot, *Journal of Robotics Systems* **11**(3), 169–179.

Hein, C. and Meystel, A.: 1994. A genetic technique for robotic trajectory planning, *Telematics and Informatics* **11**, 351–364.

Kang, D., Hashimoto, H., and Harashima, F.: 1995. Path generation for mobile robot navigation using genetic algorithm, in: *Proceedings of the 1995 IEEE IECON. First International Conference on Industrial Electronics, Control and Instrumentation*, Vol. 1, pp. 167–172.

Kalos, M. H. and Whitlock, P. A.: 1986. *Monte Carlo Methods. Volume I: Basics*, Wiley.

Leonard, J. J. and Durrant-White, H. F.: 1992. *Directed Sonar Sensing for Mobile Robot Navigation*, Kluwer Acad. Publ.

Potvin, J., Duhamel, C., and Guertin, F.: 1996. A genetic algorithm for vehicle routing with backhauling, *Applied Intelligence* **6**, 345–355.

Ricotti, M. and Liotta, A.: 1995. Real time landmarks detection for the mobile robot PARIDE, in: *Proceedings of the SPIE*, Vol. 2423, pp. 60–71.

Salichs, M. A., Armingol, J. M., Moreno, L. E., and De la Escalera, A.: 1999. Localization system for mobile robots in indoor environments, *Integrated Computer-Aided Engineering* **6**, 303–318.

Thrun, S., Fox, D., Burgard, W., and Dellaert, F.: 2001. Robust Monte Carlo localization for mobile robots, *Artificial Intelligence* **128**, 99–141.

Triggs, B.: 1994. Model-based sonar localization for mobile robots, *Robotics and Autonomous Systems* **12**, 173–186.