doi.org/10.32426/engresv4n2-001

Engineering Research

Technical Reports

Volume 4 – Issue 2 – Article 1

ISSN 2179-7625 (online)

APPROXIMATE MOTION MODEL FOR MOBILE ROBOTS LOCALIZATION

Henrique Renno de Azeredo Freitas[1], José Walter Parquet Bizarria[2], Luis Fernando de Almeida[3]

JULY / 2013

Taubaté, São Paulo, Brazil

1 Center for Weather Forecasting and Climate Research.
2 University of Taubate, Department of Informatics.
3 University of Taubate, Department of Informatics.

Engineering Research: Technical Reports

The “Engineering Research” is a publication with purpose of technical and academic knowledge dissemination.

APPROXIMATE MOTION MODEL FOR MOBILE ROBOTS LOCALIZATION

Abstract. *In mobile robotics, most part of the techniques which aim the elaboration of an efficient algorithm to the problem of localization is based on probabilistic approaches. Recently, one have applied algorithms bond to data that are conditioned to uncertainty, like the Monte Carlo Localization (MCL) that is part of a family of probabilistic algorithms dependent on the quality of two models: sensors and motion model. This paper proposes an approach for the generation of the motion model based on a discrete model and from that one searches the adequacy of a function of linear nature in order to represent the motion model in an approximated way, with the purpose of detecting the robot motion's stochastic behavior. The obtained results show that the proposed models work for the problem of localization and they can be alternatives of choice to the MCL algorithm.*

1. INTRODUCTION

Autonomous mobile robotics is a fascinating research field for many reasons, among them is notable its classification as one of the greatest approximations ever conceived of intelligent agents. A mobile autonomous robotic system is a ground, marine, or aerial vehicle consisting of all the integrated components (mobility platform, sensors, computers, and algorithms) required to perceive, learn, and adapt in the environment to make intelligent decisions for navigating, communicating, and accomplishing required tasks [1].

Mobile robots can be applied to several tasks such as transportation, exploration, surveillance, guidance, inspection, etc. Due to the freedom of motion that mobile robots present in their environment, three important questions need to be analyzed [2]: “Where am I?”, “Where am I going?” and “How should I get there?” which the former is of fundamental importance and is the basis to answer the others, being classified as the “localization problem”.

The localization problem can be defined as the problem of finding out the coordinates of a mobile robot relative to its environment. Some authors consider it as one of the most fundamental problems in mobile robotics [3] [4] [5]. The position or pose to be determined in a given instant of time t consists of a tridimensional vector (x, y, θ) called state, where x and y represent its coordinates in the Cartesian plane and θ its orientation as shown in Figure 1.

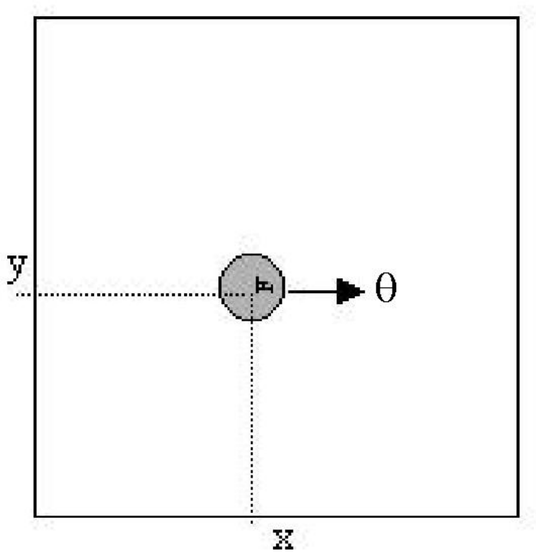

Figure 1. Representation of a circular mobile robot’s pose.

The localization problem can be subdivided into three distinct classes:

- Position tracking: situation in which the robot knows its initial pose and must keep knowledge of future states that it will possibly be in as time goes by, seeking to correct possible errors that will have accumulated related to the robot’s odometry.

- Global localization: it is a more difficult problem that consists of localizing the robot without prior knowledge of its initial pose.
- Kidnapped robot: this problem is an unexpected situation to the robot such that in a given moment its pose is changed or some object in the environment changes, forcing the robot to estimate new states which possibly will not be linked to the latter calculated estimations.

2. LOCALIZATION TECHNIQUES

A basic way of classifying localization techniques is to distinguish those that utilize landmarks or reference points from those that combine the model of data from sensors.

A landmark or reference point is a physical characteristic located in the environment used by the robot to help it localize itself. Landmarks may be natural, that is, present in the environment without the purpose of making the task of navigation easier for the robot. Among them are walls, doors, corners, etc [6]. They may also be artificial, as marks placed in the environment, as for instance, geometric figures painted in a defined color.

A group of methods that have been very well spread nowadays is that which searches to combine information acquired from the sensors system assembled with a map of the environment. These methods show very efficient results even when they face highly noisy observations. Notable in this group are: Kalman Filter [7] [8], Markov Localization [9] and Monte Carlo Localization [5] [10] [11]. Among these three techniques the Monte Carlo Localization, also known as MCL, comes as the most recent and efficient solution for the localization problem of mobile robots. Current researches having applications with real robots have demonstrated that this technique has proved to be robust from simpler problems of localization as Position tracking to those of greater complexity as Kidnapped robot.

2.1 Monte Carlo Localization

The Monte Carlo Localization algorithm (MCL) is a probabilistic localization algorithm where its main idea is to represent the belief about the pose of the mobile robot as a set of samples or particles, constructed according to a prior distribution of states [10]. It is a recursive Bayesian filter which from a initial set of N particles one can estimate the posterior distribution of the robot based in sensor data along with an action applied by the robot during its interaction the environment. The basic algorithm is described in Figure 2.

```
MCL(bel(s_{t-1}), o_t, a_{t-1})
begin
  for i = 1 to N do
    select s_{t-1} from bel(s_{t-1}) according to a weight w
    select s_t according to p(s_t | s_{t-1},a_{t-1})
    make w_t = p(o_t | s_t)
    add s_t and w_t in bel(s_t)
   end-for
   normalize bel(s_t)
end-algorithm
```

Figure 2.MCL Algorithm.

The main concept about the MCL algorithm is the fact of representing the belief bel(s_t) in all possible poses that the mobile robot may be as a set of N random samples.

These samples are defined as $\{(x,y,\theta),p\}$, where (x,y,θ) denote the robot's pose and $0 \leq p \leq 1$ a numeric factor of weight. For the Global localization problem, the initial distribution is sampled with uniform probability over the space of all possible states. When the initial pose of the robot is known this distribution is represented as a Gaussian distribution centered in its initial pose. From this distribution, the belief is recursively updated.

In the MCL algorithm showed in the Figure 2, one can notice that the posterior distribution is generated from two predefined probabilistic models: the motion model $p(s_t|s_{t-1},a_{t-1})$ that represents the robot's cinematic behavior, that is, the conditional probabilities that the robot goes to a state given its current state and the action it has executed; and the perception model $p(o_t|s_t)$ which defines the conditional probabilities for every reading from the robot's sensors in each state.

3. PROPOSALS OF MOTION MODELS

In this work, two models will be presented for validity under simulation processes. Firstly, one opted for a discrete motion model and finally the adequacy of a function to represent an approximated motion model. All simulations were made in the Saphira simulator, version 6.2, developed by SRI International's Artificial Intelligence Center. The mobile robot simulated was Magellan Pro (Figure 3) developed ISRobotics. Magellan Pro is 40.6 cm in diameter, 25.4 cm in height, and it has 16 sonar sensors distributed symmetrically, being 8 in its frontal part and 8 in its back part, allowing the robot to have a perception of 360º of the environment. The robot contains two usual DC motors, making it possible forward and backward movements as well as turns along its center.

Figure 3. Magellan Pro Robot.

3.1 Discrete Motion Model

In order to construct a discrete motion model, the robot was placed initially in a known pose and following was applied an action and recorded, at every one second interval, how much it has shifted. The samples were collected for speeds ranging from 100 mm/s to 500 mm/s in steps of 10 mm/s and angular velocities ranging from 0 degrees/s to 30 degrees/s in steps of 10 degrees/s.

The shift of the mobile robot along the x and y axis (namely *des_x* and *des_y*) was calculated by means of Eqs. (1) and (2), where is a shift of the robot's orientation and dist a value of the Euclidian distance. From these values it has been defined a probability for every noise when considering one hundred repetitions of each action.

$$des\_x = dist * \cos(des\_\theta) \tag{1}$$

$$des\_y = dist * sen(des\_\theta) \tag{2}$$

All recorded values were stored in a table in which every row represented the noise resulted from an action applied to the mobile robot. In this way, each action had a different number of registered noises with very distinct probability values.

3.2 Approximated Motion Model

The approximated motion model considers the set of noises acquired in the simulations as probability distributions. Its construction was based in a linear function that maps a value of probability to a certain noise that is related to ranges of an accumulated probability distribution in which the ranges are defined by the noises over the three variables that represent the state of the mobile robot.

The selected noise to be added to the current state of the mobile robot by the approximated motion model is calculated through a method similar to a transformation of values between two different scales of temperature, which is a linear method quite used in physics where initial and final values of the scale are used as well as a value inside this range in order to obtain a new value in the other scale, as shown in Figure 4.

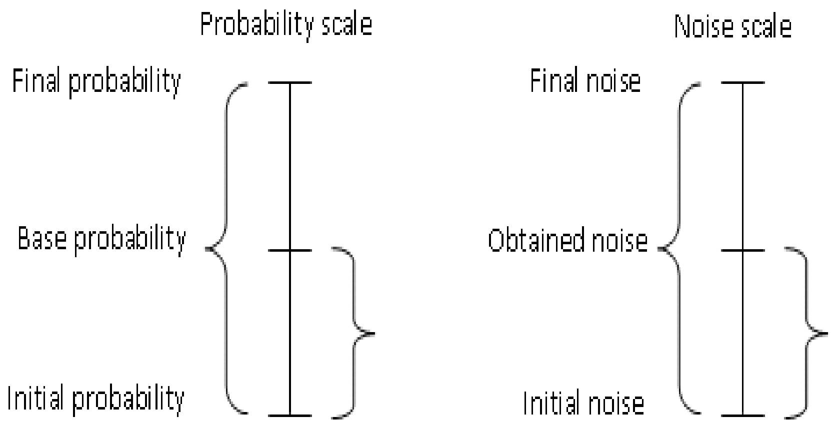

Figure 4. Transformation of distinct scales.

From a probability value generated randomly by the Monte Carlo algorithm, one can find in the set of noises the range that contains this value and apply a linear transformation to obtain a new noise that is going to be added to three variables of the current state of the mobile robot.

The new acquired noise is calculated based on the difference between the extreme values of the probability ranges and the value of each noise as well as on the difference between the random probability value and the initial probability value. In this way, the formula is defined by Eq. (3), where n is the quantity of noises for a determined action. After a simple algebraic manipulation, it can obtain Eq. (4), that is the proposed formulation to be utilized by the Monte Carlo algorithm for the calculation of a new state of the mobile robot.

$$r = \frac{(r_i - r_{i-1})}{(p_i - p_{i-1})}(p - p_{i-1}) + r_{i-1} \tag{3}$$

$$r - r_{i-1} = \frac{r_i - r_{i-1}}{p_i - p_{i-1}}(p - p_{i-1}) \tag{4}$$

3.3 Computational Representation of the Simulations

The environment utilized in the simulations was the same proposed by [11], planned to present some simple obstacles of two kinds: wall and corner. The distance values to obstacles are part of a set of ten possible values, identified by the numbers from “0’ to “9”, where each value represents a region of 25 cm from the environment’s discretization as exemplified in Figure 5.

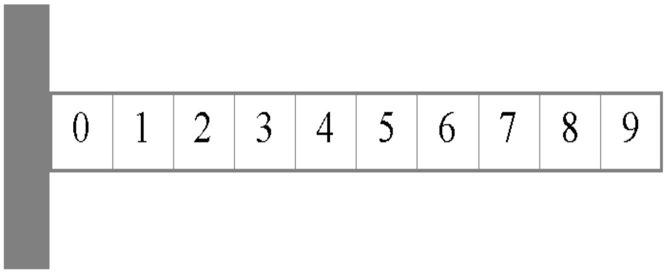

Figure 5. Discretization that indicates the closest obstacle to the robot.

Thus, when one evidences that the robot finds itself in a pose in the environment which the region of reference to the spatial representation has a value of “1” as the distance of the closest obstacle, then it means that the robot is located in a distance ranging from 25.1 cm to 50 cm from this obstacle.

4. TESTS AND RESULTS

The simulations were conducted for a verification of the differences between the obtained results with the use of the discrete and approximated motion models. The efficiency analysis of both motion models studied through the simulations was realized with focus on the Position tracking problem.

In the simulations was considered as a quantity N of the set of particles the values 1000, 2000 e 3000. In each simulation one recorded at every interval of 20 seconds the states shown by the simulator and that returned by the Monte Carlo algorithms, where one uses the discrete model and the other the approximated model. In Tables 1, 2 e 3 one can see some results obtained for values of N equal to 1000, 2000 e 3000, respectively. The errors are considered for calculations as the Euclidian distance between each of the values of x and y coordinates, where all values are defined in millimeters.

Table 1. Differences between simulated and estimated for values N = 1000.

Simulation	Simulator			Discrete Model			Approximated Model			Error D	Error A
	TH	X	Y	TH	X	Y	TH	X	Y		
1	359	2.9	0.5	2	2.787	0.54	2	2.851	0.545	0.1199	0.0665
2	357	2.8	0.4	1	2.739	0.501	1	2.886	0.491	0.1180	0.1252
3	356	3.3	0.4	359	3.14	0.459	359	3.418	0.437	0.1705	0.1237
4	357	2.8	0.4	359	2.739	0.48	359	2.697	0.476	0.1006	0.1280
5	2	2.8	0.5	0	2.739	0.494	0	2.895	0.489	0.0613	0.0956
6	357	2.8	0.4	359	2.739	0.457	359	2.959	0.446	0.0835	0.1655
7	357	2.8	0.4	359	2.786	0.487	359	2.970	0.484	0.0881	0.1896
8	357	2.8	0.4	359	2.739	0.463	359	2.929	0.453	0.0877	0.1395
9	357	2.8	0.4	359	2.704	0.502	359	2.991	0.400	0.1401	0.1910
10	357	2.8	0.4	0	2.833	0.495	0	2.882	0.491	0.1006	0.1225

Table 2. Differences between simulated and estimated for values N = 2000.

Simulation	Simulator			Discrete Model			Approximated Model			Error D	Error A
	TH	X	Y	TH	X	Y	TH	X	Y		
1	356	3.0	0.4	1	2.840	0.460	358	3.127	0.414	0.1709	0.1278
2	357	2.8	0.4	3	2.773	0.563	3	2.933	0.466	0.1652	0.1485
3	356	2.8	0.4	357	2.739	0.461	357	2.906	0.430	0.0863	0.1102
4	356	3.0	0.4	1	2.839	0.509	1	3.155	0.516	0.1944	0.1936
5	356	2.8	0.4	0	2.739	0.499	0	2.780	0.499	0.1163	0.1010
6	0	2.9	0.5	0	2.739	0.494	0	2.726	0.489	0.1611	0.1743
7	356	2.8	0.4	1	2.786	0.501	1	2.870	0.502	0.1020	0.1237
8	356	2.8	0.4	0	2.739	0.501	1	2.722	0.502	0.1180	0.1284
9	359	2.8	0.5	2	2.739	0.532	1	2.810	0.507	0.0689	0.0122
10	356	2.8	0.4	3	2.739	0.542	2	2.825	0.540	0.1545	0.1422

Table 3. Differences between simulated and estimated for values N = 3000.

Simulation	Simulator			Discrete Model			Approximated Model			Error D	Error A
	TH	X	Y	TH	X	Y	TH	X	Y		
1	359	2.7	0.4	2	3.000	0.545	0	2.760	0.468	0.1502	0.0907
2	356	3.1	0.4	0	2.985	0.479	359	3.242	0.500	0.1395	0.1737
3	358	2.8	0.4	0	2.786	0.500	0	2.947	0.436	0.1010	0.1513
4	356	2.8	0.4	1	2.739	0.511	0	2.874	0.485	0.1267	0.1127
5	356	2.8	0.4	0	2.739	0.489	0	2.749	0.469	0.1079	0.0858
6	3	2.9	0.5	358	2.739	0.495	356	2.951	0.463	0.1611	0.0630
7	356	2.9	0.4	0	2.787	0.490	1	2.860	0.488	0.1445	0.0967
8	356	2.8	0.4	359	2.739	0.462	0	2.797	0.468	0.0870	0.0681
9	356	2.8	0.4	2	2.739	0.521	0	2.740	0.465	0.1355	0.0885
10	356	2.9	0.4	359	2.739	0.496	356	3.074	0.458	0.1874	0.1834

From the error data shown in Tables 1, 2 e 3 were generated the following error graphs, namely Figures 6, 7 e 8, in which D represents the discrete model and A the approximated model. The errors are practically alike in relation to the form that they behave, although this depends on what action is being applied to the mobile robot. As a consequence of this fact, results from the application of several distinct actions need to be shown to the analysis of each model.

Observing Figure 6, one can verify that for 1000 particles the discrete motion model showed only two errors considered big while the approximated model also showed small errors although they were more accentuated. Thus, the discrete model seemed more proper when reducing the quantity of particles.

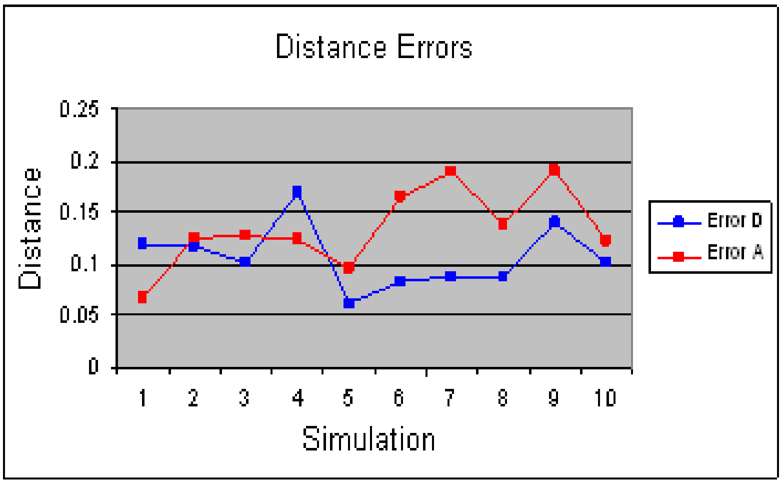

Figure 6. Distance errors for N = 1000.

In Figures 7 and 8, one can observe that for 2000 and 3000 particles, respectively, the approximated motion model resulted less distance errors with higher frequency while the discrete model presented some errors near 10 cm. In this way, the approximated model demonstrated to be a possibility as an application for a greater quantity of particles, what represents a practical result of the statistical concept that the greater is the quantity of sample data the higher is the fidelity of the modeled results in relation to what has been observed.

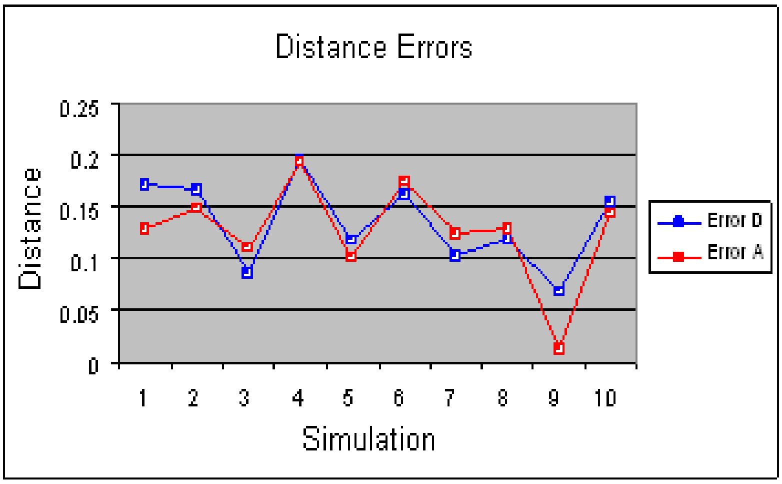

Figure 7. Distance errors for N = 2000.

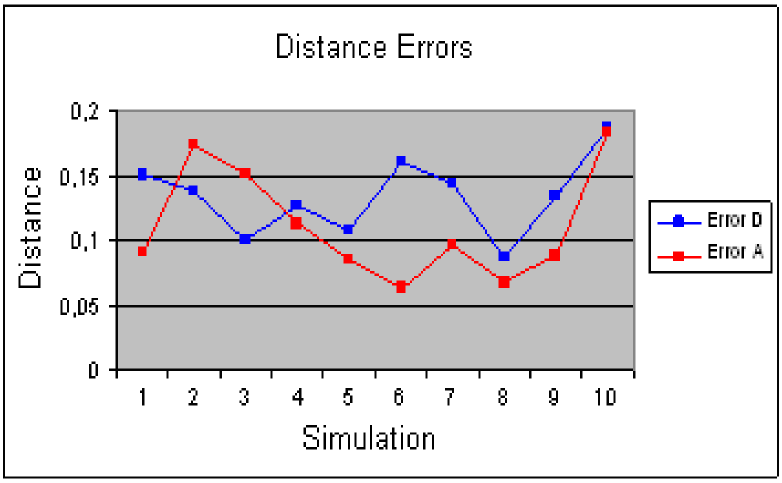

Figure 8. Distance errors for N = 3000.

For comparison purposes between the obtained results, with a quantity of particles being different in each simulation, Table 4 presents the mean and standard deviation for the localization errors for every configuration. In that table one can also observe the percentage in relation to how many times one model presented to be better than the other as well as the difference between means.

Table 4. Analysis of results for different quantities of particles.

	Discrete Model		Approximated Model		Relation between Models		
N	Mean	σ	Mean	σ	Discrete	Approximated	Difference
3000	0.1341	0.0299	0.1114	0.0431	20%	80%	0.02269
2000	0.1338	0.0412	0.1262	0.0489	40%	60%	0.00757
1000	0.1070	0.0314	0.1347	0.0391	70%	30%	0.02769

Although the obtained means of distance errors have presented very close values in a precision of two or three decimal places, one can observe that the standard deviation of the discrete motion model is lesser than the standard deviation of the approximated motion model. However, the approximated model exhibited an improvement when the quantity of particles increased, resulting in a frequency of less errors in relation to the discrete model.

7. CONCLUSIONS

This work presented a study about motion models applied to the MCL algorithm used for the localization problem. It has been proposed two alternatives of motion model: a discrete and an approximated model.

A first observation is that the MCL algorithm really presents as a very good solution to the localization problem in mobile robotics. Meanwhile, its efficiency is totally dependent on the quality of the available probabilistic models. The motion models studied in this work presented good results reaching a small error range for the navigation theme.

In this way, one can consider that both the discrete and the approximated models are options to be applied as a solution along with the MCL algorithm. However, it is important to notice that these models were defined based on simulations, and, therefore, it would be interesting to future works a deeper analysis with a real mobile robot.

Another important aspect is that the proposed motion models do not take into consideration the influence the distinct kinds of floors could have over the robot motion (friction and sliding). Thus, for more complex models, one can suggest a study that may incorporate a coefficient related to the kind of floor where the robot acts.

At last, in future stages of this work, it would also be interesting to realize several tests in order to generate estimates for the Global localization problem, which is a more difficult problem and that demand more time and computational resources for the mobile robot to present a more adequate localization and possibly even to the Kidnapped robot problem.

REFERENCES

[1] W. L. Fehlman II, M. K. Hinders, *Mobile Robot Navigation with Intelligent Infrared Image Interpretation*, Springer-Verlag London, 2009.

[2] J. J. Leonard, H. F. Durrant-Whyte, *Directed sonar sensing for mobile robot navigation*, Kluwer Academic Publishers, 1992.

[3] I. J., Cox, G. T. Wilfong, *Autonomous robot vehicles*. New York: Springer Verlag. 1990.

[4] J. Zhu, N. Zheng, Z. Yuan, An improved technique for robot global localization in indoor environments, International Journal of Advanced Robotic Systems, V.8, N.1, pp. 21-28, 2011.

[5] S. Thrun, D. Fox, W. Burgard, F. Dellaert, Robust Monte Carlo localization for mobile robots, Artificial Intelligence, N.128, pp. 99-141, 2001.

[6] D. Meyer-Delius, M. Beinhofer, A. Kleiner, W. Burgard, Using artificial landmarks to reduce the ambiguity in the environment of a mobile robot, 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 5173-6178, 2011.

[7] R. Smith, M. Self, P. Cheeseman, Estimating uncertain spatial relationship in robotics. In I. J. Cox, G. T. Wilfong (Eds), Autonomous robot vehicles, New York, Springer Verlag, pp. 167-193, 1990.

[8] S. Thrun, W. Burgard, D. Fox, *Probabilistic Robotics*, MIT Press, London, 2005.

[9] D. Fox, W. Burgard, S. Thrun, *Markov Localization for Mobile Robots in Dynamic Environments*, Pittsburgh, Carnegie Mellon University, 1999.

[10] D. Fox, Adapting the sample size in particle filters through KLD-sampling, International Journal of Robotics Research IJRR, V. 22, N. 12, pp. 985-1003, 2003.

[11] L. F. Almeida, C. H. C. Ribeiro, Mobile robot localization based on monte carlo method and genetic algorithms, In Proceeding of V IFAC EURON Symposium on Intelligent Autonomous Vehicles, Portugal, 2004.

[12] M. Han, S. Rhee, Navigation Control for a Mobile Robot, Journal of Robotics Systems, V.11, N.3, pp.169-179, 1994.