

An Autonomous Robot Map Creation and Map Optimization Using Genetic Algorithms and Artificial Neural Networks

INCI CABAR*¹, OSMAN K. EROL¹, SIRMA YAVUZ²

¹Department of Computer Science

Istanbul Technical University

²Department of Computer Engineering

Yildiz Technical University

Istanbul Teknik Universitesi, Elektrik-Elektronik Fakultesi

Bilgisayar Mühendisliği Bolumu, Maslak/Istanbul

TURKEY

incicabar@yahoo.com

okerol@itu.edu.tr <http://www3.itu.edu.tr/~okerol/>

sirma@ce.yildiz.edu.tr <http://www.yildiz.edu.tr/~smyavuz/>

Abstract: - This paper is about an autonomous robot map creation and optimization algorithm. To create the map, calibrated sensor data transferred to the x-y coordinate system were used. Afterwards we tried to optimize the anomalies of the map with different methods as artificial neural networks, genetic algorithms.

Key-Words: - Robot Map, Genetic Algorithm, Artificial Neural Networks, Autonomous Robot, Three-Wheeled Robot, Map Optimization, DBSCAN, Robot Kinematics, Sensor Calibration, Map Creation

1 Introduction

The biggest problem of the autonomous robots is creating an optimized map. Researchers are developing different solutions to handle this problem.

To create a robot map, robot must know its sensor positions. These sensors could be sonar, laser, infrared and so on. But the sensor data are not sufficiently consistent. Furthermore most of the sensor range is limited. For instance, light and sound can't pass through the wall. Because of these constraints for creating a map, the robot must travel around and not stay constant.

There can be problems while the robot creates the environment's map. The most frequently problems that can be occur are as follows[5]:

Sensor Measurement Noise: The basic problem of the mapping is the noisy measurements. If the noises observed from different measurements were statistically independent, mapping problem will be much more easy. However these noises are probabilistically dependent and grow up in course of time. Accordingly a really small rotation fault will augment during the robot tour and finally will cause false mapping of the environment.

Large Map Dimensions: One of the other mapping problems happens if there are many objects in the environment. So the map dimensions will increase and these dimensions will effect the model of the map prediction and the robot position.

Data Association Problem: The third and the most difficult problem of the mapping process is data association problem. We can define this problem as the capability of understanding of the robot that it passes at the same points in different intervals of time. During the robot trip, the probability of passing in the same point grows up exponentially, so the solution of the data association problem becomes more and more difficult.

The Changes in the Environment: If the places where the robot travel are dynamically changing (opening the door, putting the chair in a different place), mapping will get harder. In the experiments, most of the time objects stay stable.

Autonomous Robot Travel: The last most common of the problems is the robot self selection of the travelling road. The chosen strategy must take precautions to the problems encountered during the mapping procedure. The off-line methods of the map prediction ignore this problem.

Because of the problems mentioned above, constructed maps don't give satisfactory results. To achieve the optimized maps different methods have been evolved. One and most of the commons methods is the Kalman Filter[2] which is a probabilistic method. Kalman filter prevent instability and provide robustness to sensor noise. Another one is the Expectation Maximization (EM) algorithm which finds maximum likelihood estimates of parameters in probabilistic models. Today, different expensive solutions to the robot mapping problem have been developed.

2 Infrared Sensor Localization

2.1 Robot Model

Our autonomous robot, Pakize, has two connected traction wheels in rear, and one free steered wheel in front. Consequently, our autonomous three-wheeled robot's kinematic equations differentiate from other wheel constructions' robot. To measure the distance, an encoder is connected to the front wheel. The robot is equipped with 6 Sharp GP2D12 infrared (IR) sensors.

2.2 Sensor Calibration

For sensor localisation, first we applied a calibration to our infrared sensors. The nominal effective range of our infrared sensors are between 10 cm and 80 cm. So we calibrated them between these intervals. For each 5 cm. interval, 1000 values of the IR sensor were read. The minimum value, maximum value, mean and variance for the 1000 sensor readings at each interval were found. The calibration curve is determined using the Piecewise Cubic Hermite Interpolating Polynomial for intervals.

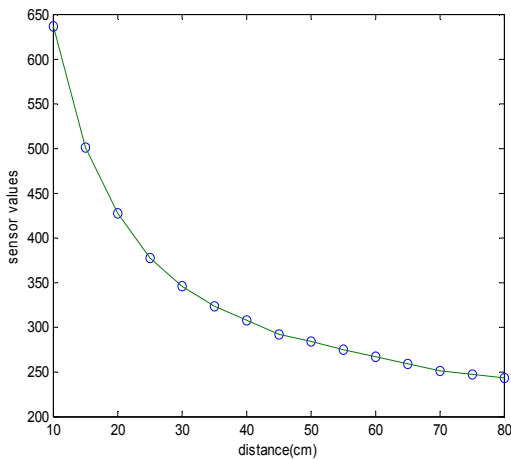


Fig.1 Calibration Curve

To see the effectiveness of these sensor values towards the end of the calibration range, we found the 95% confidence intervals using the equations (1).

$$CI_{95} = \bar{x} - \frac{1.96\sigma}{\sqrt{n}} \leq \mu \leq \bar{x} + \frac{1.96\sigma}{\sqrt{n}} \quad (1)$$

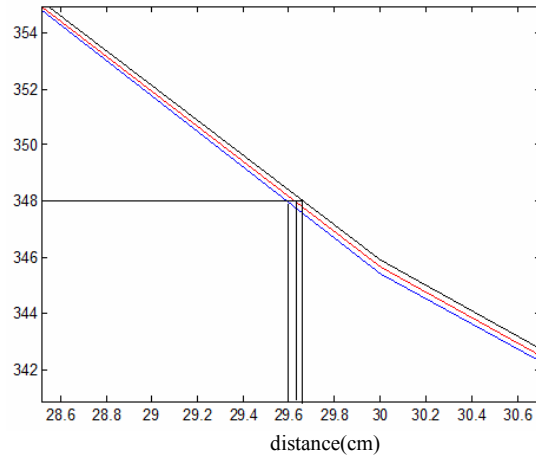


Fig.2 Confidence Intervals

From this curve, we can imply that when the sensor readings decrease, the confidence interval increases and when the confidence interval increases, the sensor readings are imprecise for using at the localization. Accordingly, using this calibration curve the robot is placed to 20 cm away from the wall to create the environment's map.

2.3 Three-Wheeled Autonomous Robot Kinematic

As we know the distance traveled (ds) from the encoder and the angle between the front wheel and the robot body (α) from the potentiometer, the angle between the rear wheel and the robot can be easily estimated.

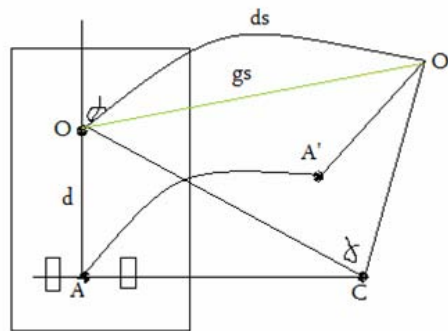


Fig.3 Robot α Degree Rotation in Direction OA to Direction O'A'

If we give a α degree rotation command to the robot when the robot is traveling in direction OA at a time t , after traveling a ds distance, new position $O'A'$ of the robot can be computed from forward kinematic equations.

$$ds = ((\text{Encoder}(t+1) - \text{Encoder}(t)) / 512) * \text{wheel_perimeter} \quad (2)$$

If we consider $O(x_{1(t)}, y_{1(t)})$, $A(x_{2(t)}, y_{2(t)})$ as initial coordinates, $O'(x_{1(t+1)}, y_{1(t+1)})$, $A'(x_{2(t+1)}, y_{2(t+1)})$ as new coordinates and θ_1 and θ_2 respectively front and rear wheel angles; the new coordinates can be computed from equations (4) and (5):

$$\begin{bmatrix} x_{1(t+1)} \\ y_{1(t+1)} \\ d \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\cos \theta_1 \\ 0 & 1 & -\sin \theta_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1(t)} \\ y_{1(t)} \\ d \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} x_{2(t+1)} \\ y_{2(t+1)} \\ d \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\cos \theta_2 \\ 0 & 1 & -\sin \theta_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{2(t)} \\ y_{2(t)} \\ d \end{bmatrix} \quad (4)$$

After the robot coordinates' estimations have finished, we also added sensor positions to these calculations. The sensor positions are found by using (x_1, y_1) , (x_2, y_2) and the sensor locations on the robot.

2.4 Creation of the Robot Map and DBSCAN

At the end of transformation of the sensor data to the x-y coordinate system, the map was created. Below there is the map created by using left-front sensor data obtained after five tours of the robot in the environment.

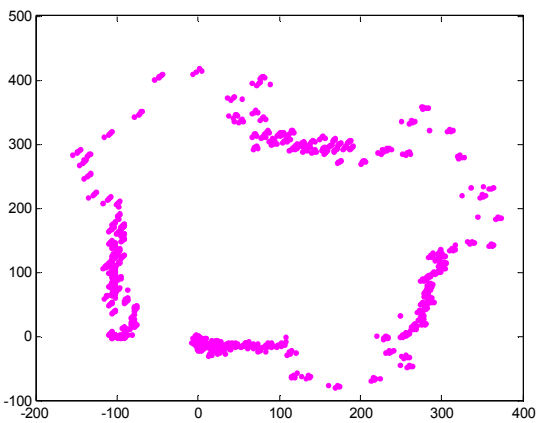


Fig.4 Map after 5 Tours in the Environment (x and y in cm)

The robot sensor data are noisy. So we applied DBSCAN, a density-based algorithm to discover clusters of noise and find out the outliers. These outliers occur especially at the corners. So map corners were modeled separately.

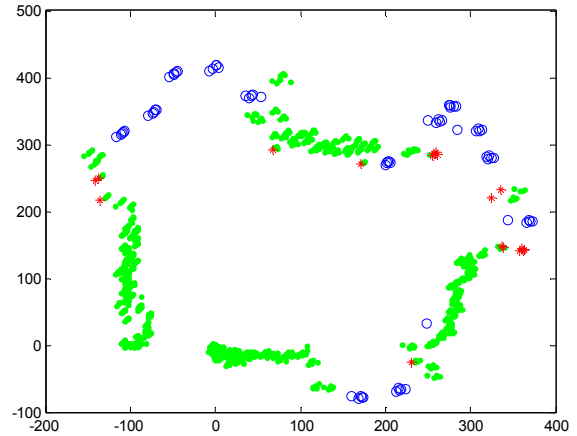


Fig.5 Map with DBSCAN (x and y in cm)

The average of these five tours were taken and we the map was recreated.

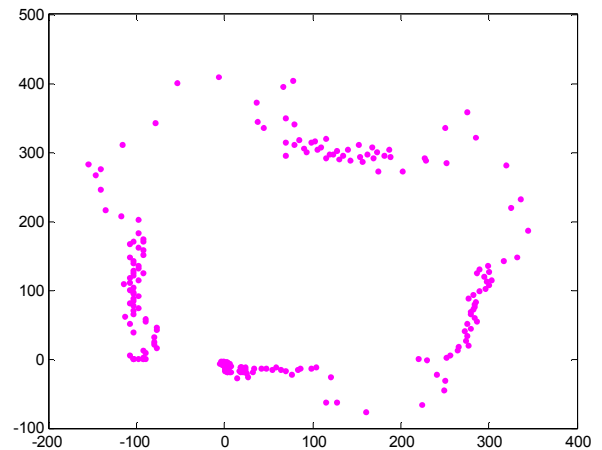


Fig.6 Averaging Map (x and y in cm)

The same process was applied for DBSCAN. The figure below show averaging map with DBSCAN.

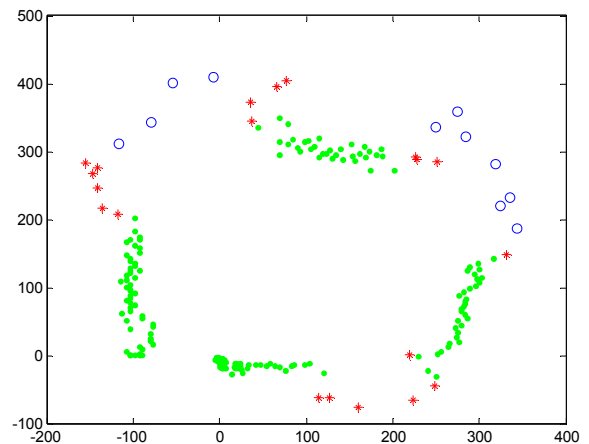


Fig.7 Averaging Map with DBSCAN(x and y in cm)

The environment where the robot was traveled has the shape of a rectangle. But as seen from the figure, our map did not give the correct shape because of the problems encountered at the introduction part. So the map must be optimized. Although the exact correctness of the map isn't possible, different methods to ameliorate the map were performed.

3 Map Optimization

In our study we used artificial neural networks and genetic algorithms for optimizing the map.

3.1 Artificial Neural Networks

The artificial neural networks (ANN) consist of a set of neurons as the human neurological system. In fact these neurons are processing elements that interact by sending signals to one another along weighted connections. To obtain the activation function, learning is necessary.

There are three types of learning: supervised, unsupervised and reinforced. Error signals used to train the weights in the network define all of the three types of learning.

In our research supervised learning was used due to its characteristic because inputs and targets are given to obtain outputs. Two-layer feedforward network is suitable for our application [6].

Levenberg–Marquardt algorithm is a learning algorithm for finding the weight that makes the performance minimum. The algorithm includes the best sides of back-propagation which is a generalization of the Widrow-Hoff error correction rule and Newton algorithms [3].

In the Back-Propagation algorithm, the performance index is defined as the sum of squared errors between the targets and the outputs.

$$F(w) = e^T e \quad (5)$$

In equation (6), the weights of the network are represented. e is the error vector for all of the training examples.[4]

When we train with Levenberg –Marquardt algorithm, the increment of weights is

$$\Delta w = [J^T + \lambda I]^{-1} J^T e \quad (6)$$

Here J is the Jacobian matrix, and λ is the training parameter.

We practised our studies on a standard two-layer feed-forward neural network trained with Levenberg-Marquardt. Hidden layer neuron size was 20. 60% of

our samples are used for training, 20% for the validation and 20% for testing.

Below there are the map optimized by artificial neural networks. Because of the need of the modeling the corners separately from the walls first figure show only the walls excluding the corners and the second-one all together walls and corners.

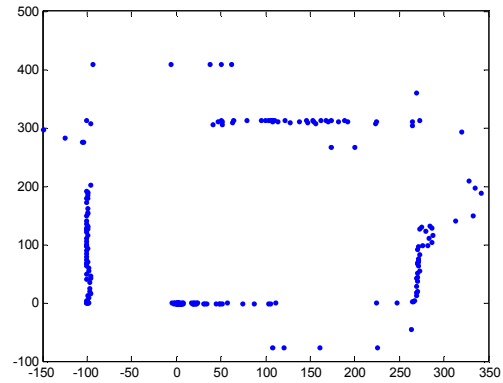


Fig.8 Map Optimized with Two-Layer Feedforward ANN,excluding the corners.

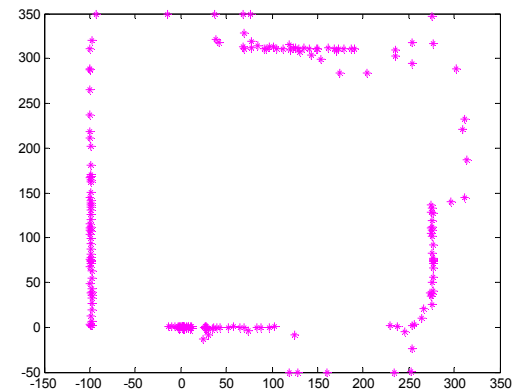


Fig.9 Map Optimized with Two-Layer Feedforward ANN, after modeling corners.

3.2 Genetic Algorithms

Genetic algorithms are global search techniques based on natural selection, biological evolution and genetics [1].

The parameters of genetic algorithms are number of generation, population size, chromosome length, and the probability of applying some operators. Fitness function is our objective function.

In genetic algorithms, the chromosomes represent the population of candidate solutions of the problem. The generations select the fittest solutions.

Three main process of the genetic algorithms are selection, crossover and mutation. Selection is the process that selects an individual to go into the mating pool. Using the crossover operator with mutation that causes diversity in the population; the selected

individuals in the mating pool are combined to produce offsprings that contain parent's genetic code [2].

In our study, an elitist genetic algorithm was used. The aim of using elitism is to preserve the best individuals of each iteration. The solution was described by $(x_1, x_2, x_3, \dots, x_n)$, n representing decimal data points. Each gene was encoded showing the knot points that symbolize the points where the cubic spline data interpolation curve is passing by. Model performance was calculated using the root mean square error.

As selection ranking selection was used and as crossover, one-point crossover was used for exchanging parent string segments and recombining them to produce offsprings.

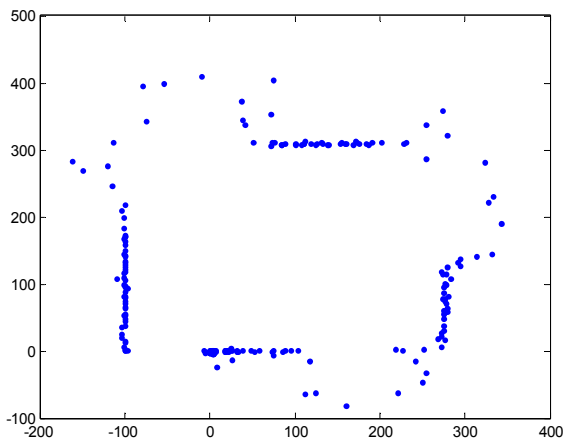


Fig.11 Map Optimized with GA, excluding the corners.

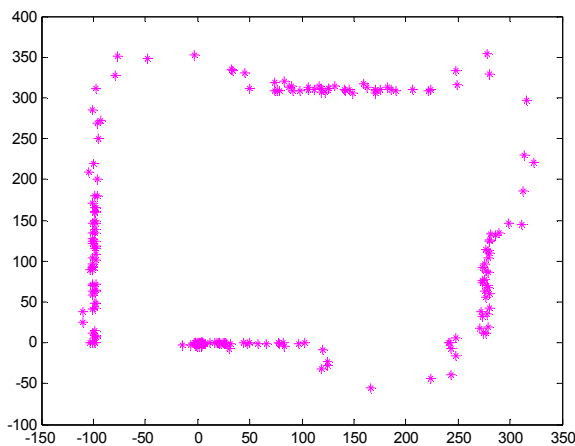


Fig.12 Map Optimized with GA, after modeling corners.

4 Conclusion

We have described a robot map creation and optimization algorithms for the autonomous three-wheeled mobile robot.

The algorithms were tested with the data recorded by an autonomous robot. The genetic algorithm was run for 250 generations with a population size of 80. The

notebook used for testing algorithm has 2.00 GHz Core 2 Duo T7200 processor. The computational time for the genetic algorithm is 21 seconds for 250 generations.

We chose genetic algorithms because the structure of genetic algorithms is helpful to solve environmental problems. With genetic algorithms, environments can be easily decomposed into rooms, corridors, so on. The major disadvantage of the genetic algorithm is its computation time.

The second optimization method used in this study is a two-layer feedforward artificial neural network with Levenberg-Marquardt algorithm. Since the performance function will reduce at each iteration of the algorithm, using Levenberg-Marquardt algorithm as learning algorithm is more efficient than Gauss-Newton methods. The other advantage of Levenberg-Marquardt learning algorithm is that since the performance function will be reduced at each iteration of the algorithm, it is possible to get minimum error with less iteration. And it is also less complex because it uses the approximate value of the Hessian matrix. For obtaining better results from artificial neural networks, we need to train the ANN with more data.

To compare the results, first we applied Kruskal-Wallis to see if there is a significant difference between the two results. And we found that there is a difference between two maps. Afterwards we compared root mean square errors of the algorithms. The RMSE for ANN was 13.2931 and for GA 4.0354. From these results we can imply that using the genetic algorithms gives better solutions than using artificial neural networks.

As future work more detailed theoretical and empirical analysis of the algorithm in different environments is planned.

References:

- [1] Duckett T., A Genetic Algorithm for Simultaneous Localization and Mapping, ICRA-2003, *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, May 12-17, 2003.
- [2] Armingol J. M., Moreno L. E., de la Escalera A., Salichs M. A., A Genetic Algorithm for Mobile Robot Localization Using Ultrasonic Sensors, *14th World Congress of IFAC International Federation of Automatic Control. Beijing, P. R. China. July 5-9, (1999).*
- [3] Chakravorti S., Mukherjee P. K., Application of Artificial Neural Networks for Optimization of Electrode Contour, *IEEE Transactions on Dielectric and Electrical Insulation*, Vol. 1, No. 2, pp. 254-263, 1994.
- [4] Chen T., Han D.J., Au F.T.K. and Tham L.G., Acceleration of Levenberg-Marquardt training of neural networks with variable decay rate,

Proceedings of the International Conference on Neural Networks, 2003, 3: 1873-1878.

- [5] Kurt Z., *Development of Intelligent Algorithms for Simultaneous Localization and Map Building Problem*, Master's Thesis, Yildiz Technical University, Istanbul 2007.
- [6] Demuth H. and M. Beale M., *Neural Network TOOLBOX User's Guide*. For use with MATLAB. The MathWorks Inc.. 1998.