Robot Mapping and Map Optimization Using Genetic Algorithms and Artificial Neural Networks

INCI CABAR*[1], SIRMA YAVUZ[2], OSMAN EROL[1]
[1]Department of Computer Science
Istanbul Technical University
[2]Department of Computer Engineering
Yildiz Technical University
Istanbul Teknik Universitesi, Elektrik-Elektronik Fakultesi
Bilgisayar Mühendisligi 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 transfered to the x-y coordinate system were used. Afterwards we tried to otimize 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, Mapping

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 faulse 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.

Robot travels the environment using the Simultaneous Localization and Map Building (SLAM) algorithm that was first suggested at the end of 80’s. Using SLAM algorithm the robot creates the map while calculating the new robot location simultaneously.

Some of the algorithms developed for simultaneous localisation and mapping don’t need to know whole data but only the previous step so they can be used in real-time applications. But some simultaneous localisation and mapping algortihms need to know all the passed steps and they can’t be used in real-time applications.

The SLAM methods have one commun characteristic; they are based on statistical methods. During the mapping and localisation processes different problems can be encountered and to solve these problems the statistical prediction methods are used.

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 commun methods is the Kalman Filter which is a probabilistic method[8]. Kalman filter prevent instability and provide robustness to sensor noise. It is an ancient method. Kalman filters are speacial Bayes filters. Here, maps are defined with cartesian coodinates of the objects that the robot finds during its travel. These objects are usually borders as walls.

Another one is the Expectation Maximization algorithm which findes maximum likelihood estimates of parameters in probabilistic models. This method solves the data association problem.

Occupancy Grid algorithm is used with one of the described methods. The aim is obtaining a metric map taking advantage of the noisy data.The map is formed from two-dimensional grids. It is based on the Bayes filters.

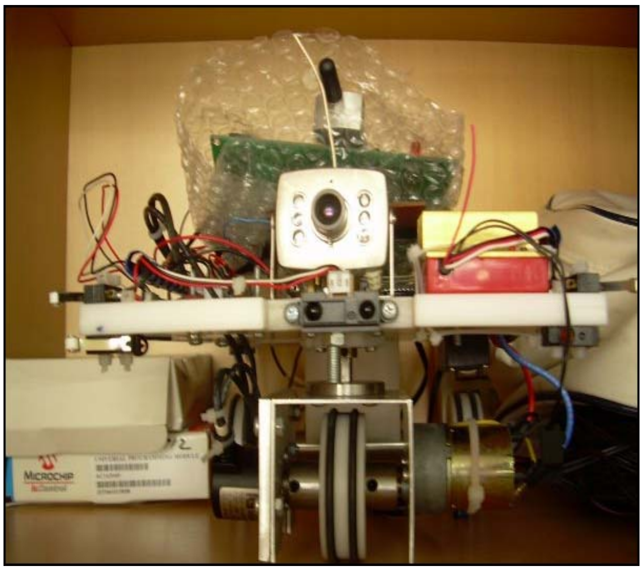
Today, different expensive solutions to the robot mapping problem have been developed.

After the travel of the robot with SLAM algorithm, a map was created using the sensor values.

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 costructions’ robot. To measure the distance, an encoder is connected to the front wheel. The robot is equipped with 6 Sharp GP2D12 infrared (IR) sensors.



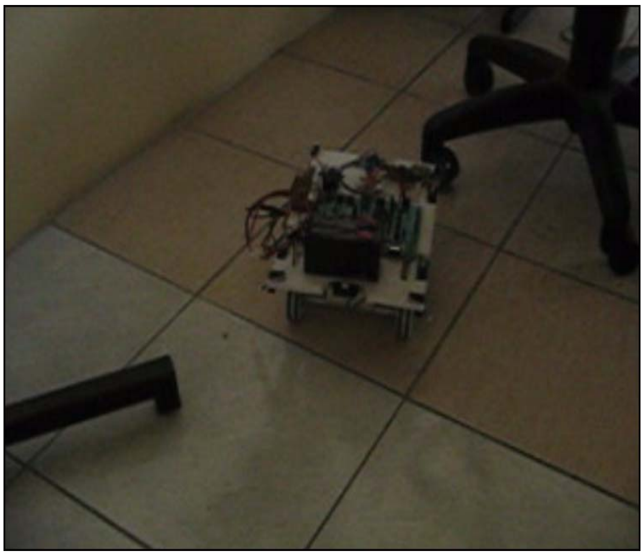


Fig.1 Different Views of The Robot

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.

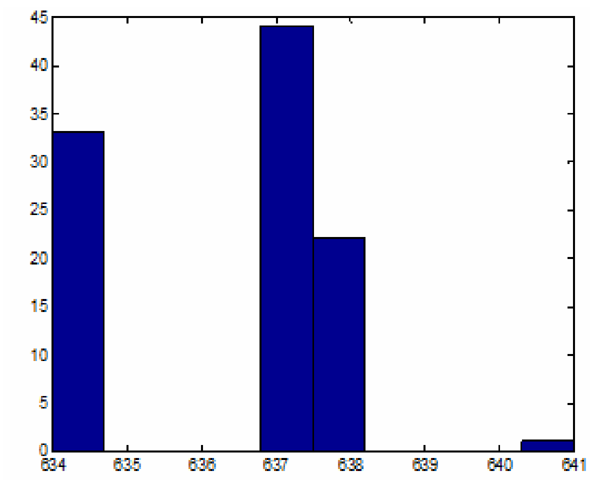


Fig.2 Sensor Values Taken From The 10 cm. to the Wall

Table 1 Sensor Estimations For 10 cm. From The Wall

min	634
max	641
mean	636.27
median	637
mode	637
std	1.6869
range	7

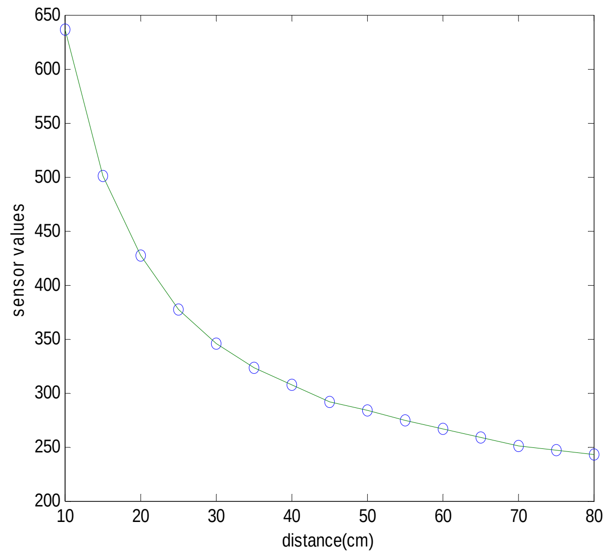


Fig.3 Calibration Curve

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

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

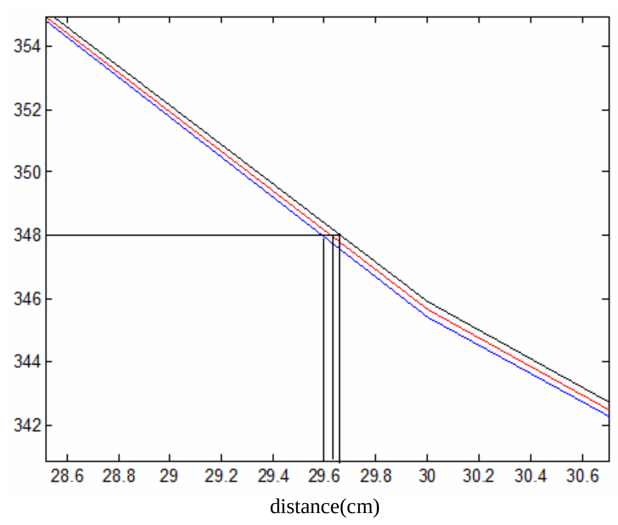


Fig.4 Confidence Intervals

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.

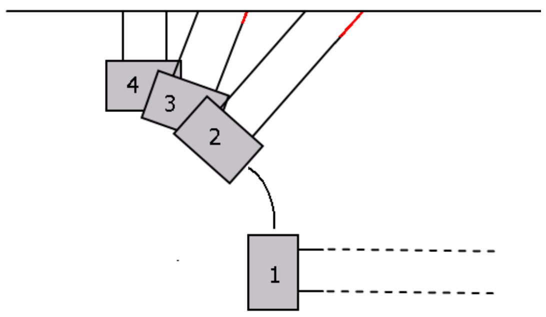


Fig.5 Robot Turning Positions

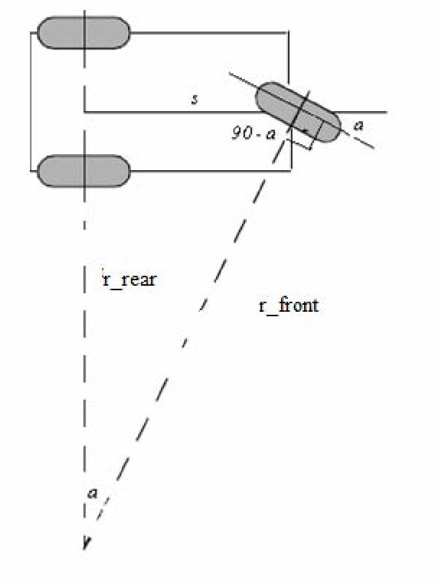


Fig.6 Rotation Center of The Robot

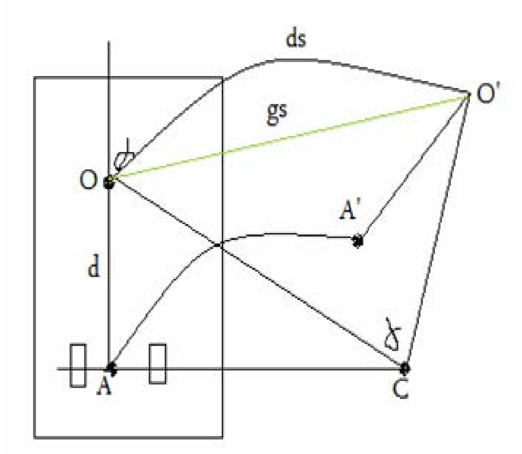


Fig.7 Robot α Degree Rotation in Direction OA to Direction O'A'

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.

So when we know the middle point coordinates of the front and rear wheels at a time t, the equations for estimation of the new coordinates at a time t+1 are as follows:

$$r_{fr} = \frac{|OA|}{\sin\alpha} = \frac{d}{\sin\alpha} \quad (2)$$

d =15 cm.

ds=((Encoder(t+1)-Encoder(t))/512)*wheel_perimeter

The γ angle for the ds arc is:

$$\frac{ds}{2\pi r_{fr}} = \frac{\gamma}{360} \quad (3)$$

Regarding to the robot slope, θ1 and θ2 angles which define new position of the robot are:

$$\theta_1 = 90 - \alpha - \frac{\gamma}{2} \quad (4)$$

$$\theta_2 = 90 - \frac{\gamma}{2} \quad (5)$$

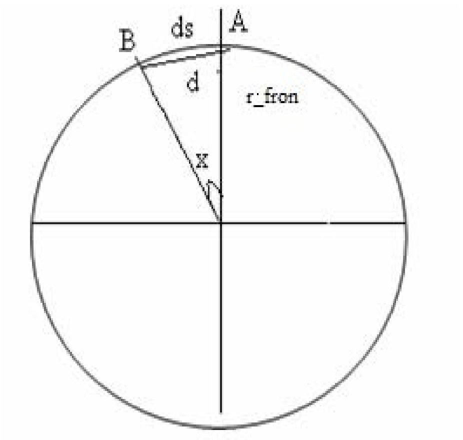


Fig.8 Front Rear Middle Point Arc

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 θ1 and θ2 respectively front and rear wheel angles; the new coordinates can be computed from equations (6) and (7):

$$\begin{bmatrix} x_{1(t+1)} \\ y_{1(t+1)} \\ d \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\cos\theta_1 \\ 0 & 1 & -\sin\theta_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1(t)} \\ y_{1(t)} \\ d \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} x_{2(t+1)} \\ y_{2(t+1)} \\ d \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\cos\theta_2 \\ 0 & 1 & -\sin\theta_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{2(t)} \\ y_{2(t)} \\ d \end{bmatrix} \quad (7)$$

After the robot coordinates' estimations have finished, we also added sensor positions to these calculations. The sensor positions are found by using (x1,y1), (x2,y2) 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.

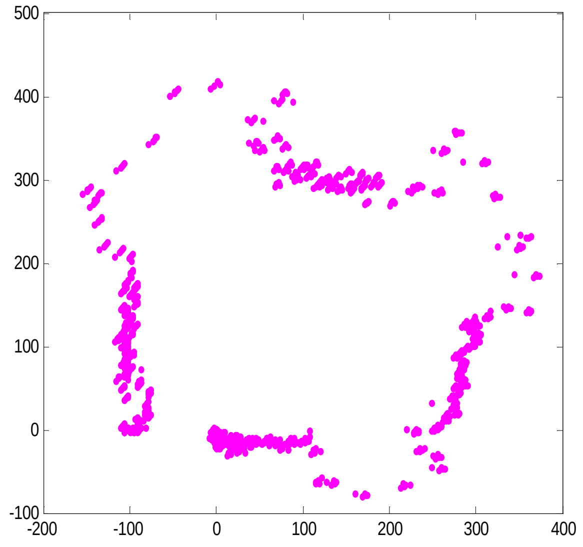
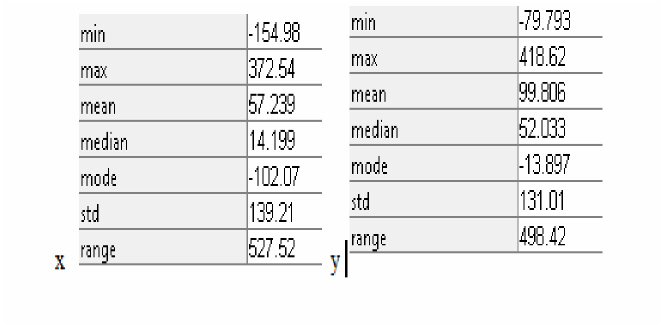


Fig.9 Map after 5 Tours in the Environment (x and y in cm)

Table 2 Five Tours' Estimations



x

min	-154.98
max	372.54
mean	57.239
median	14.199
mode	-102.07
std	139.21
range	527.52

y

min	-79.793
max	418.62
mean	99.806
median	52.033
mode	-13.897
std	131.01
range	498.42

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.

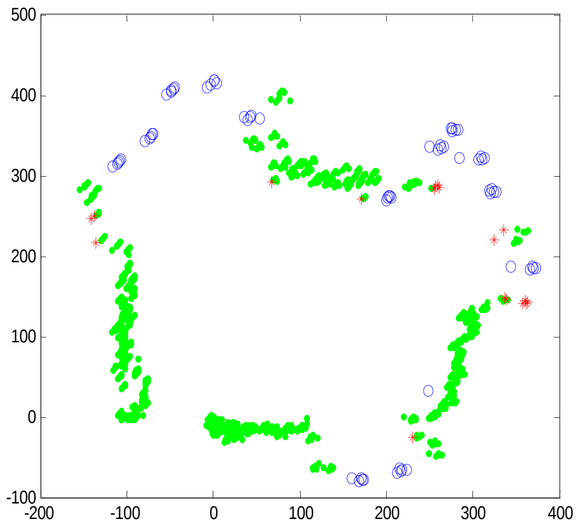
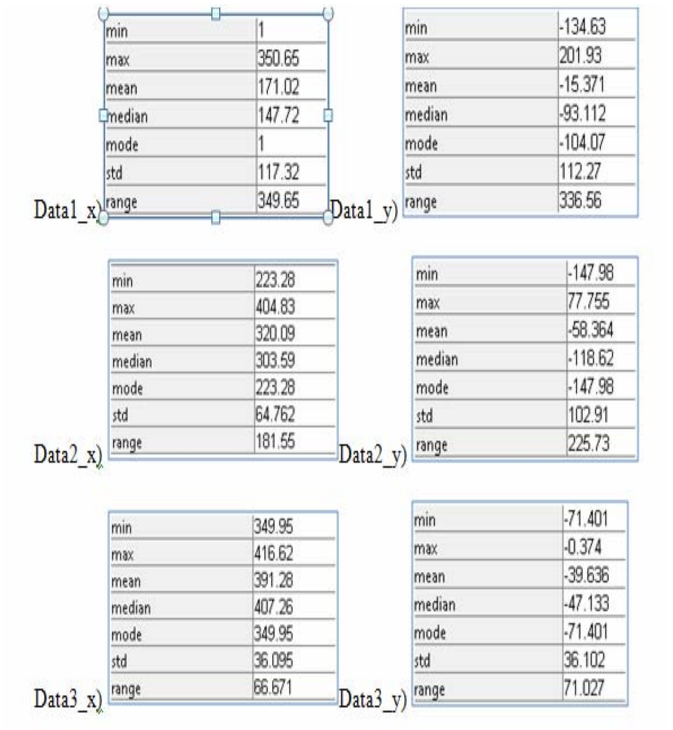


Fig.10 Map with DBSCAN (x and y in cm)

Table 3 DBSCAN Estimations



Data1_x)

min	1
max	350.65
mean	171.02
median	147.72
mode	1
std	117.32
range	349.65

Data1_y)

min	-134.63
max	201.93
mean	-15.371
median	-93.112
mode	-104.07
std	112.27
range	336.56

Data2_x)

min	223.28
max	404.83
mean	320.09
median	303.59
mode	223.28
std	64.762
range	181.55

Data2_y)

min	-147.98
max	77.755
mean	-58.364
median	-118.62
mode	-147.98
std	102.91
range	225.73

Data3_x)

min	349.95
max	416.62
mean	391.28
median	407.26
mode	349.95
std	36.095
range	66.671

Data3_y)

min	-71.401
max	-0.374
mean	-39.636
median	-47.133
mode	-71.401
std	36.102
range	71.027

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

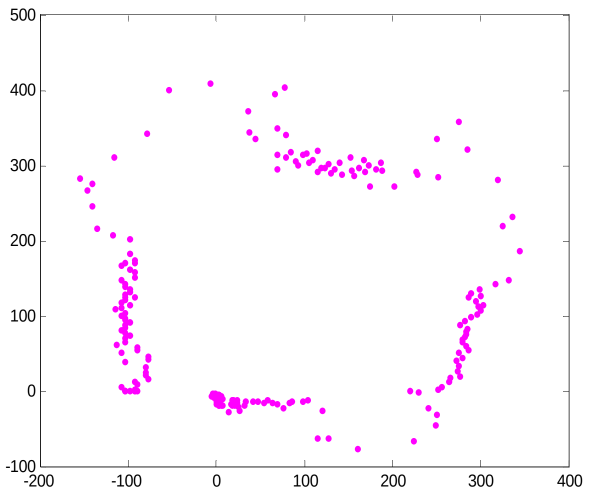


Fig.11 Averaging Map (x and y in cm)

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

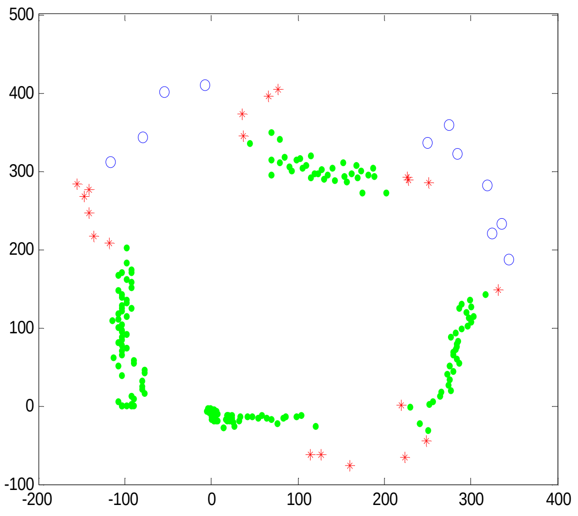


Fig.12 Averaging Map with DBSCAN(x and y in cm)

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[7].

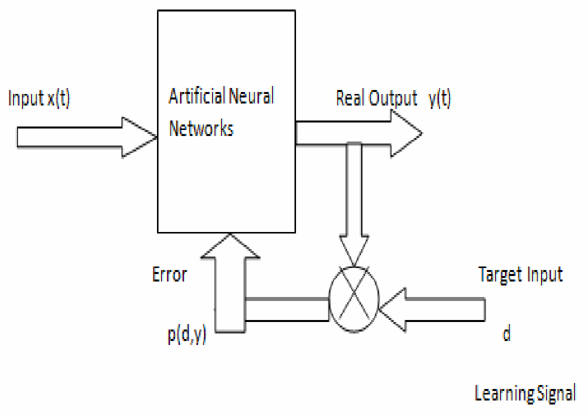


Fig. 13 Supervised Learning

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,p].

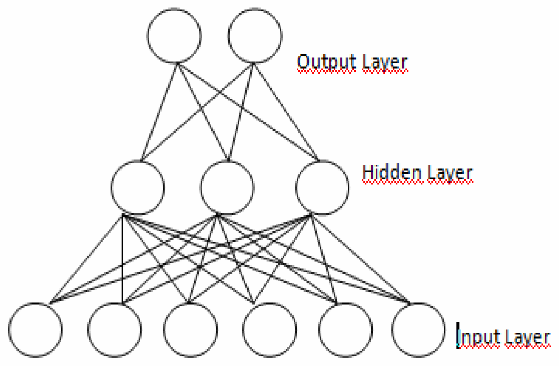


Fig. 14 Two-Layer Feedforward ANN

Levenberg–Marquardt algorithm is a learning algorithm for finding the weight that makes the performance minimum. The algortihm includes the best sides of back-popagation 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.

$$F(w)= e^{T} e \qquad (8)$$

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

$$\Delta w= [J^{T} J + \lambda I]^{-1} J^{T} Je \qquad (9)$$

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.

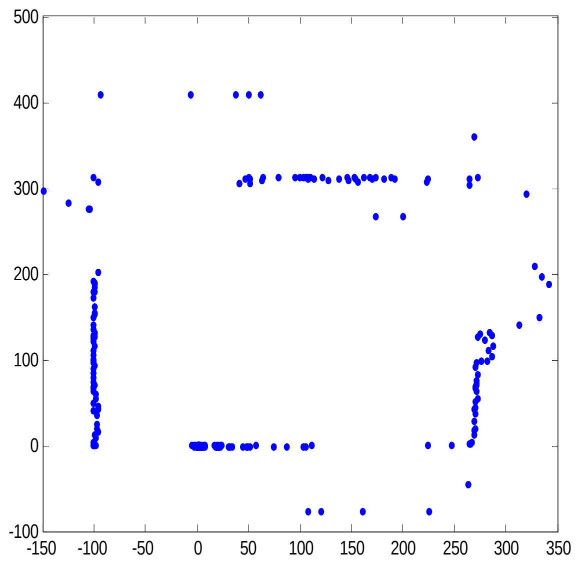


Fig.15 Map Optimized with Two-Layer Feedforward ANN,excluding the corners.

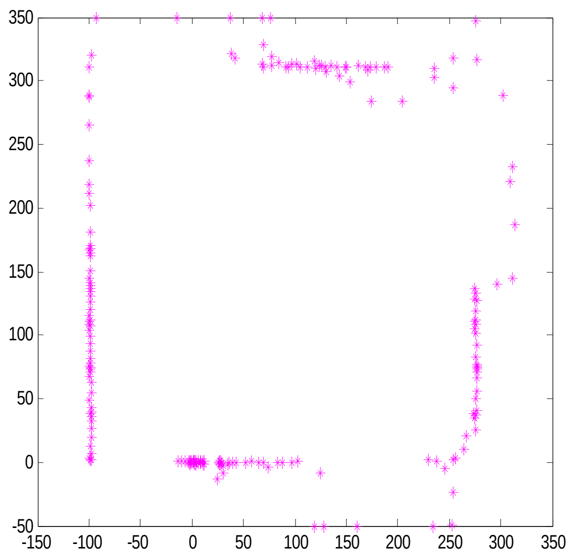


Fig.16 Map Optimized with Two-Layer Feedforward ANN, after modeling corners.

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.

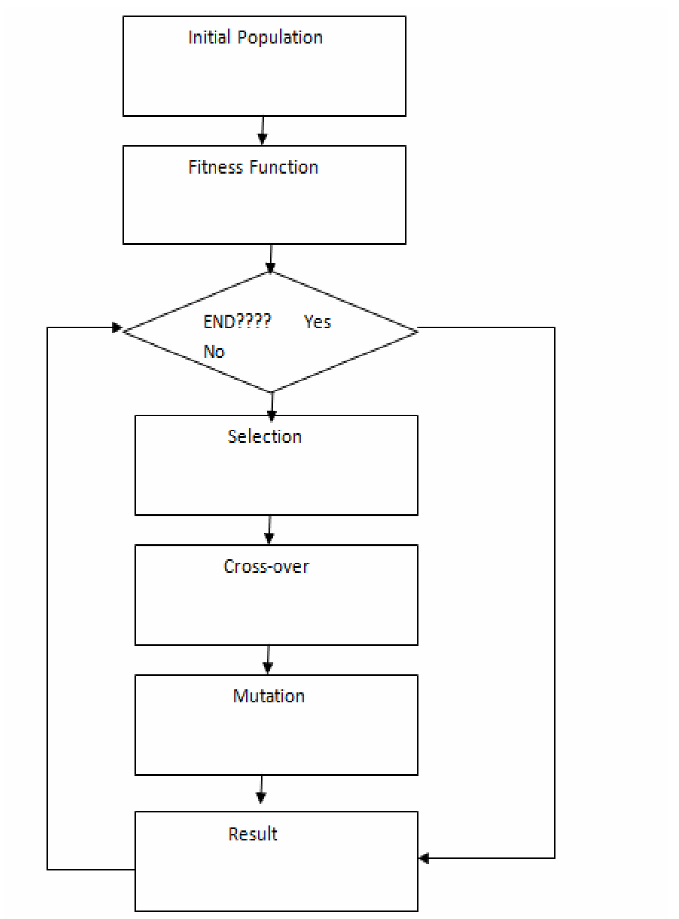


Fig.17 Genetic Algorithm Flow Chart

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.......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.

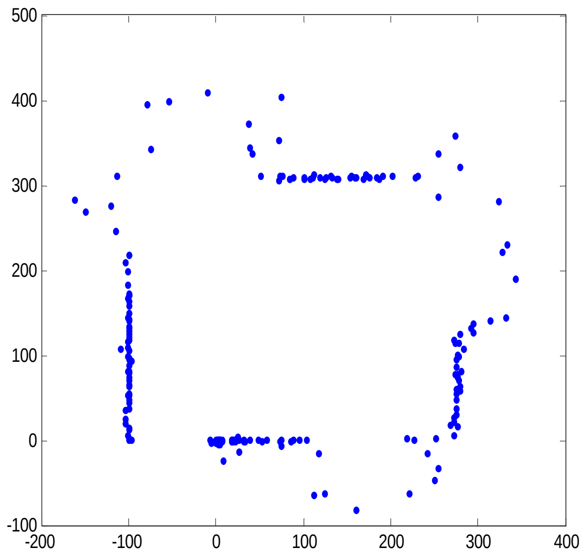


Fig.18 Map Optimized with GA, excluding the corners.

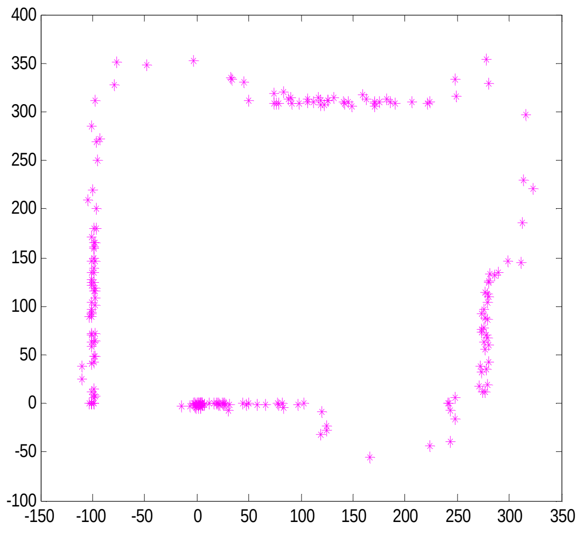


Fig.19 Map Optimized with GA, after modeling corners.

3.3 Genetic Algorithms and Artificial Neural Networks

At the last stage of our study , we applied consecutively artificial neural networks and genetic algorithms for optimizing the map.

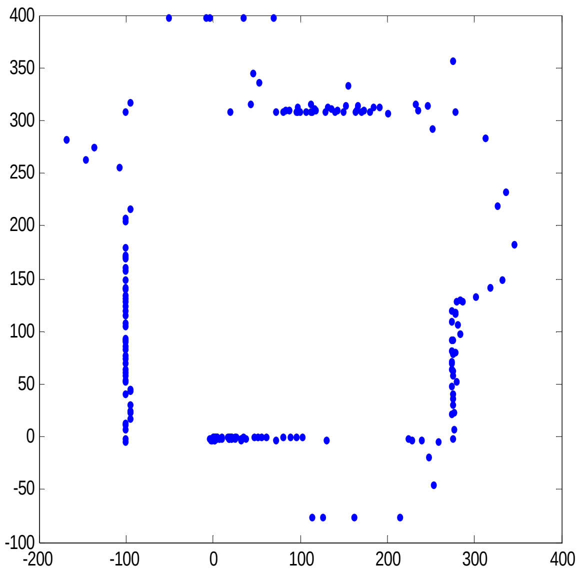


Fig. 20 Map Optimized with ANN and GA, excluding the corners.

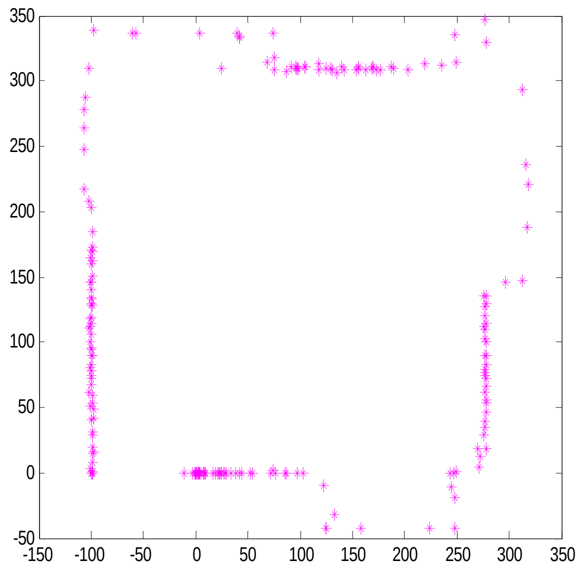


Fig.21 Map Optimized with ANN and GA, after modeling corners.

4 Conclusion

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

The algorithms were tested with the data recorded by an autonomuos 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 choosed genetic algorithms because the structure of genetic algortihms is helpfull 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 the results. And we found that there is a difference between the 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 algortihms 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. Bcale M., *Neural Network TOOLBOX User's Guide.* For use with MATLAB. The MathWorks Inc.. 1998.

[7] Sawarkar S.D.,Ghatol A.A.,Pnade A. P, Neural Network Aided Breast Cancer Detection and Diagnosis Using Support Vector Machine, *7th WSEAS International Conference on Neural Networks*, Cavtat, Croatia, June 12-14, 2006

[8] Kim Du Y., Shin V.,Optimal Receding Horizon Filter for Continuous-Time Nonlinear Stochastic Systems, *6th WSEAS International Conference on SIGNAL PROCESSING*, Dallas, Texas, USA, March 22-24, 2007

[9] Rodriguez A., Carricajo I.,Hybrid Approach to MK Classification of Stars Neural Networks and Knowledge-based Systems, *6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*, Corfu Island, Greece, February 16-19, 2007