Position Estimation of Mobile Robots Using Unsupervised Learning Algorithms

Chapter · January 2010

DOI: 10.1007/978-3-642-10781-8_11

2 authors:

Petre Lameski
Ss. Cyril and Methodius University in Skopje

Andrea Kulakov
Ss. Cyril and Methodius University in Skopje Macedonia

Position Estimation of Mobile Robots Using Unsupervised Learning Algorithms

Petre Lameski and Andrea Kulakov

Faculty of Electrical Engineering and Information Technologies, Skopje, R. Macedonia,
Karpos 2 bb, P.O. Box 574, Skopje, R. Macedonia
{lameski,kulak}@feit.ukim.edu.mk

Abstract. Estimating the position of a mobile robot in an environment is a crucial issue. It allows the robot to obtain more precisely the knowledge of its current state and to make the problem of generating command sequences for achieving a certain goal an easier task. The robot learns the environment using an unsupervised learning method and generates a percept – action- percept graph, based on the readings of an ultrasound sensor. The graph is then used in the process of position estimation by matching the current sensory reading category with an existing node category. Our approach allows the robot to generate a set of controls to reach a desired destination. For the learning of the environment, two unsupervised algorithms FuzzyART neural network and GNG network were used. The approach was tested for its ability to recognize previously learnt positions. Both algorithms that were used were compared for their precision.

Keywords: Mobile robots, Position estimation, Unsupervised learning.

1 Introduction

Position estimation is a challenging task in mobile robotics, especially when dealing with a previously unknown environment. The mobile robot must be able to learn the environment by memorizing the states that it has previously visited and in the same time recognize the previously visited positions. Position estimation is also used in the problems of environmental mapping and localization problems. There are many approaches in the literature for both environment learning and mapping problems and localization and position estimation problems. [1][2]

Many approaches for solving the mapping and localization problem are proposed in the literature. For the environmental learning problem, the occupancy grid map is an elegant way of representing the knowledge about the environment. For the problem of localization in a known environment some of the proposed approaches are Grid localization, Montecarlo localization and many other localization approaches [1]. These proposed approaches use the previous knowledge of the environment (a map of the environment) and localize the robot in the known environment.

The problem of Simultaneous Localization and Mapping (SLAM), which urges the robot to learn the environment and estimate its position in the same time, has many different approaches [1]. There are many proposed variations of SLAM algorithms based on Kalman filters, Particle filters and other types of filters that give good results in the SLAM problem and are widely used.

The action planning problem is also an important issue in mobile robotics. One of the most widely used approaches are the Markov decision process (MDP) and the partially observable MD. These methods, however, are computationally expensive for large maps [1].

Artificial neural networks have been used for the problem of environment learning and localization [2]. They give good results in qualitative localization but not in fine localization.

Artificial neural networks are also widely used in the area of object recognition, classification [3] and in control problems [4] where they are used for robot navigational behavior.

For a mobile robot to be able to automatically move and in the same time learn an unknown environment all of these problems must be solved in the same time, thus the need of combining the solutions in a single system. A fully autonomous robot must be able to perform the previously stated actions.

This approach is proposing a solution that combines the environmental learning, position estimation and action planning in a single solution, as it was presented in [5]. In this paper we are using GNG network for the learning of the environment, and comparing it with the FuzzyART network, used for the same purpose. The mobile robot is behaving autonomously and explores the environment by learning the new positions and in the same time recognizing the previously visited position. The obtained knowledge of the environment can be used for generating control sequence for the robot to be able to reach a desired position.

2 System Architecture

In this paper we are using an integrated system for environmental learning, position estimation and planning of actions of a mobile robot [5]. The goal of this system is for the robot to be able to learn a previously unknown environment and be able to use the knowledge about the environment to generate actions for achieving a certain goal or a certain position in the environment.

The system is using a Percept – Action – Percept model [6] [7] of learning, in which, the positions of the robot are evaluated by the sensor readings. Each position is connected to another position by the action that the robot performed or needs to perform in order to reach that position. The Percept – Action –Percept model is shown in Fig 1.

The data about the environment is organized as a connected graph. The nodes contain the information of the robot state, and the links contain the information about the action that is needed for transition from one state into another as in Fig 2. This type of data organization makes the problem of action planning very simple. All the robot needs to do is to use a graph search algorithm and find the action sequence needed to reach the desired position. During the process of executing the actions, the robot must verify that its current position is the one that is desired by the action plan.

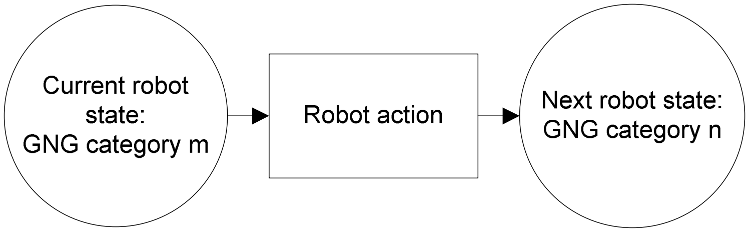

Fig. 1. Percept – action – percept model

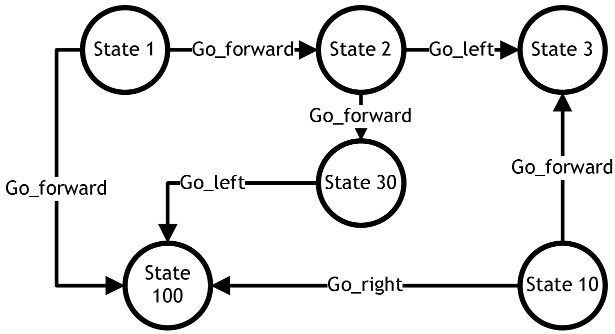

Fig. 2. State graph representation of the environment with typical actions

Each percept at the robot's position is categorized by using an unsupervised learning algorithm. Two algorithms, which are used for that task, are the FuzzyART neural network [8] [9] in the first approach (already considered in [5]) and the GNG network [10] [11] in the second approach (considered in this paper). A Lego Mindstorms NXT robot is used as a robot platform and its ultrasound sensor is used for perceiving the environment. The robot uses two servo motors for the movement in the environment and one servo motor to rotate the ultra-sound sensor in different directions. For each robot position, 12 measurements in different directions are taken. These measurements define the percept of the position.

The GNG network is used for remembering and estimating the positions of the mobile robot in the environment. Each percept acquired by the robot is normalized and then used as input in the GNG neural network. After a category is selected, the graph is checked for an existing node. If none exists, the new category is added to the graph as a new state and a new action link is created from the previous category. If the category exists, then if necessary, a new action link, containing the necessary action as in Fig. 2, is created to connect the previous one with this category. The state represents the position in which the robot is at the moment. Different state means that the robot estimated it is as a different position. The percept in a robot position is used for determining the state. In Fig. 3 is given the activity diagram of the developed processes.

Adaptive resonance theory (ART) networks develop stable recognition codes by self-organization in response to arbitrary sequences of input patterns. They are able to continue to learn from new events without forgetting previously learned information. ART networks model several features such as robustness to variations in intensity, detection of signals mixed with noise, and both short- and long-term memory to accommodate variable rates of change in the environment. There are several variations of ART-based networks, but we have used FuzzyART, which has analog inputs, as most appropriate for these kinds of tasks [5].

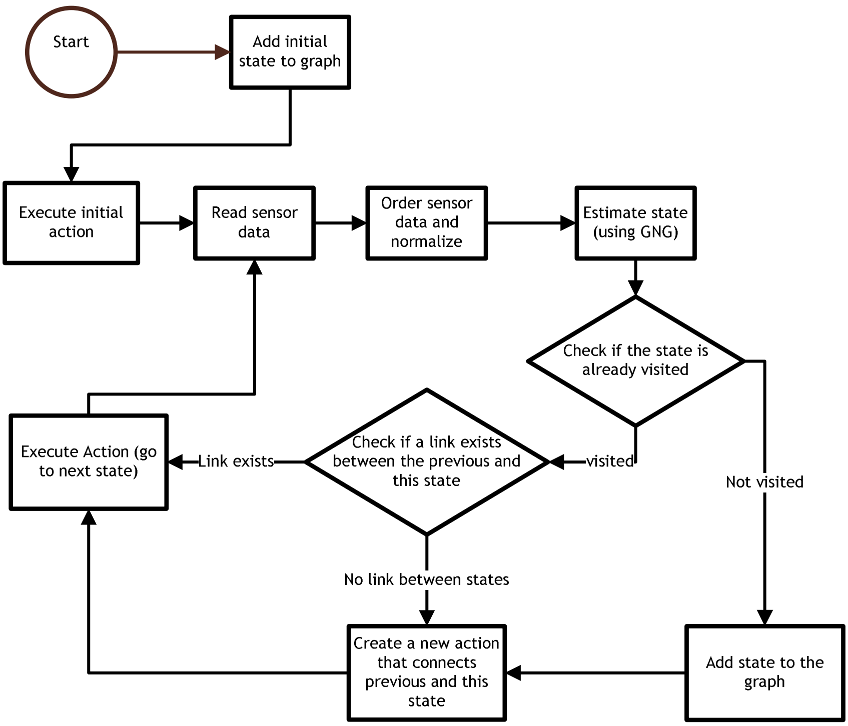

Fig. 3. UML Activity diagram

The GNG algorithm is an unsupervised incremental clustering algorithm. Given some input distribution in the input space R^n, GNG incrementally creates a graph, or network of nodes, where each node in the graph has a position in R^n. GNG is an adaptive algorithm in the sense that if the input distribution slowly changes over time, GNG is able to adapt, that is to move the internal nodes, so as to cover the new distribution. The algorithm constructs a graph in which nodes are considered neighbours if they are connected by an edge. The neighbour information is maintained throughout the execution by a variant of competitive Hebbian learning [11]. The nodes of the GNG graph tend to follow the distribution of the signals. Each node in the GNG algorithm is consisted of the following:

- a reference vector, in R^n,
- a local accumulated error variable, and
- a set of edges defining the topological neighbours of the node k.

The reference vector represents the node in the signal space. The local accumulated error is a statistical parameter that is used for determination of the place where a new node should be inserted. Each edge between the nodes has an aging parameter used for removing old and inactive edges. This is necessary since the nodes are moved.

The GNG algorithm first initializes two nodes, and links them. For each signal that is received, a winner node (the node that is closest to the signal in R^n space) is found and the second closest node is found. The winner node is moved towards the signal and all topological neighbours are moved towards the signal too. The error variable is updated for the winner node and its topological members. Then the age of the edges of the winner node are reset to 0. If there is an edge between the winner node and the second closest node, it is reset to 0, if an edge does not exist, it is created. If a creation criterion is met, usually when the number of iterations reaches a certain multiple of a constant number λ, a new node is created between the nodes that have the largest accumulated error. The error of these nodes is updated. Edges that reach a certain pre-specified age and nodes that don't have topological neighbours are deleted. In this way, the algorithm creates nodes and clusters of nodes in the areas where the input signal distribution is highest. The algorithm stops when certain stopping criterion is met.

In our case each node cluster is a position of the robot. The robot therefore remembers the positions by placing nodes, or clusters of nodes in the R^n space of the GNG network. Each node or node cluster represents the information about the percept of the robot. The GNG algorithm does not have stopping criteria in our case, as the robot needs to learn the environment constantly.

3 Experiments and Evaluation Results

The system was implemented as in [5], on a PC and tested on a Lego Mindstorms NXT robot with rotating ultrasound sensor. The sensor readings were sent by the Lego NXT robot on the PC and the necessary calculations were done on the PC. This was done because of the limited memory and data structure organization that were available on the brick alone.

The system was tested in an indoor environment. The robot was given a task to explore an unknown environment by moving in a loop and memorizing the new and in the same time, recognizing the learnt positions. The robot also actively memorizes the sequence of actions it took in order to arrive at a certain position. After the learning process, the robot is expected execute a command sequence that would take it to a desired position.

The robot starts at an initial state. For each state the robot takes 12 readings from the ultrasound sensor in 12 different directions. These directions are then reordered according to the angle they were taken from. In absolute angles, from the initial direction (the direction of the initial pose) of the robot, the readings are taken from -180, -150, -120, -90, -60, -30, 0, 30, 60, 90, 120 and 150 angle degrees. If the robot direction is different from the initial direction, the relative angle of the reading directions is changed respectively. These values are then normalized for the GNG input in the presented approach and thus become the feature of the robots state (position). The GNG is then used to decide whether the robot's state is a new one or an old one, as in Fig 4. The obtained category identifies the state of the robot and is invariant of the robots direction angle. Which means that the sensor readings are ordered so that the array of the readings always start with the reading taken at -180 degrees and ends with 150 degrees in a global coordinate system for the environment. For simplicity the robot was allowed to rotate for 90 degrees in any direction and to move for a fixed length forward or backward.

Tests were made to evaluate the ability of the approach to recognize already visited states with the GNG algorithm. This allowed an estimation of the precision that the system has in learning the states. Due to the errors in the movement of the ultrasound sensor used, the robot tends to get different readings in the same state and map it as a different one. It also recognizes a new state as a state that it has already visited. For testing of the approach, the robot was ordered to move in a closed squared loop, as in Fig 5. The goal was for the robot to learn the new states and to recognize an old state when needed to. In the first loop the robot initializes the graph and remembers the states and the actions performed.

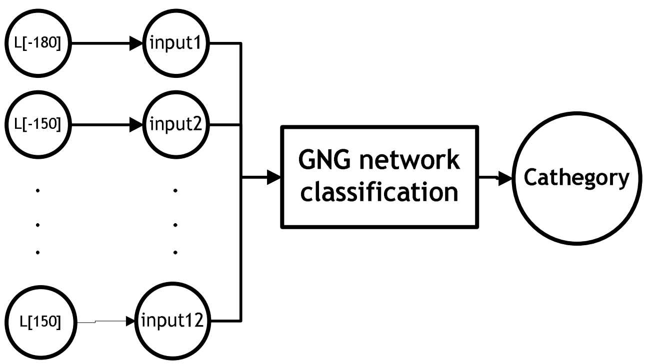

Fig. 4. Category determination using GNG network

The states are on a single step distance from each other. For each state the robot takes the measurements using the ultrasound sensor and the GNG network estimates the state that the robot is into. The robot was expected to recognize the states it has visited. The example labeling of the states is given in Fig 5. In this way we measured the precision of this method for the purpose of state recognition of the robot.

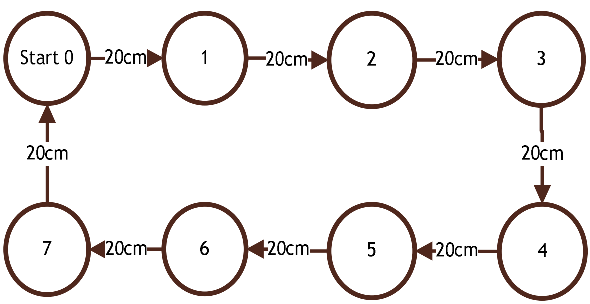

Fig. 5. States of the robot used for testing

The action planning for reaching a certain state using this method is a simple graph search algorithm that extracts the actions needed for reaching a certain position from a given position both represented as states.

The results of the tests are given in Fig 6. The experiment showed that both of the algorithms give similar results in the position estimation task. The FuzzyART is, however, slightly better than the GNG algorithm in the estimation of the state of the robot. The FuzzyART algorithm also tends to learn the positions faster, giving a stable position even after only one loop. The GNG algorithm learns the positions a bit slower and tends to recognize old states as new ones and continue to recognize them as the newer state in the next loops.

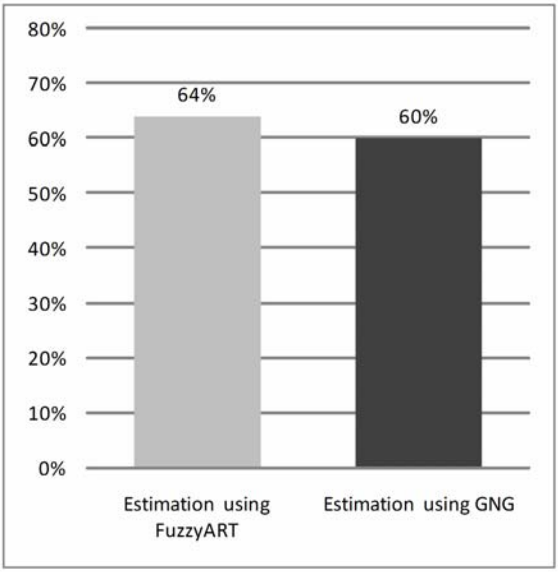

Fig. 6. Percentage of guessed categories

4 Conclusion

The obtained results showed errors in the state estimation. The reasons for these errors are the imprecision of the ultrasound sensor that is prone to miss some readings of the distance, especially if the object is a surface that has large angle with the ultrasound waves. This reason, combined with the angle errors of the rotation of the ultrasound sensor, tended to give imprecise readings that contained significant errors. These errors led the system to miss a state or to recognize two different states as the same category. The results shown in this paper are similar to the ones shown in [5]. The GNG approach however, was slightly less accurate.

These experiments showed that relying on a single ultrasound sensor for state estimation gives imprecise estimates. A good approach would be to use 12 or more different distance sensors attached in the appropriate angles in order to avoid the rotation errors of the single sensor. The nearly same error of both unsupervised learning algorithms confirmed the above statement.

Another reason for the obtained results is the ambiguity of the environment. The positions, which the robot was expected to recognize, were too close to each other and the errors of the sensor measurements of +-3cm were more than 10% of the distance between the positions, which also contributed to the error of the estimation. Furthermore the small distance between different positions narrows the gap between them in the state space also. Both the FuzzyART and the GNG networks occasionally tend to classify neighboring positions as the same one.

Another issue to be taken in consideration is that the robot learns the environment on the fly. This gives additional uncertainties in the state estimation. The main advantage of this approach however, is that the path planning of the mobile robot becomes a typical graph search problem that is easily solved.

In the future the same experiment would be repeated with 12 or more distance sensors including different types of equipment that would remove or at least minimize the errors that were made due to the imperfectness of the equipment used for these experiments.

Since the ultrasound sensors have shown to be imprecise and the learning of new positions and recognizing old ones, based on the data acquired from them, prone to errors, further consideration would be done to use different kinds of sensors for acquiring the environment data.

Acknowledgments

The work described in this article has been partially funded by the European Commission's Sixth Framework Programme under contract no. 029427 as part of the Specific Targeted Research Project XPERO ("Robotic Learning by Experimentation") and partially by the Ministry of Education and Science of Macedonia.

References

[1] Thrun, S., Burgard, W., Fox, D.: Probabalistic Robotics, 1st edn. The MIT Press, Cambridge (2005)

[2] Racz, J., Dubrawski, A.: Qualitative Pose Estimation Using An Artificial Neural Network. In: ICAR 1995 (1995)

[3] Mayer, G., Kaufmann, U., Kraetzschmar, G.: Neural robot detection in robocup. In: Wermter, S., Palm, G., Elshaw, M. (eds.) Biomimetic Neural Learning for Intelligent Robots. LNCS (LNAI), vol. 3575, pp. 349–361. Springer, Heidelberg (2005)

[4] Omidvar, O., Van der Smagt, P.: Neural Systems for Robotics. Academic Press, New York (1997)

[5] Lameski, P., Kulakov, A., Davcev, D.: Learning and position estimation of a mobile robot using FuzzyART neural network. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics (2009)

[6] Kulakov, A., Stojanov, G.: Structure, Inner Values, Hierarchies, and Stages: Essentials for Developmental Robot Architecture. In: Proceedings of the 2nd International Workshop on Epigenetic Robotics, pp. 63–70. Lund University Cognitive Studies (2002)

[7] Stojanov, G., Trajkovski, G., Kulakov, A.: Interactivism in artificial intelligence (AI) and intelligent robotics. New Ideas in Psychology 24, 163–185 (2006)
[8] Grossberg, S.: Adaptive Resonance Theory. Encyclopedia of Cognitive Science. Macmillan Reference Ltd., Basingstoke (2000)
[9] Carpenter, G.A., Grossberg, S., Rosen, D.B.: Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks 4, 759–771 (1991)
[10] Fritzke, B.: A self-organizing network that can follow non-stationary distributions. In: Gerstner, W., Hasler, M., Germond, A., Nicoud, J.-D. (eds.) ICANN 1997. LNCS, vol. 1327, pp. 613–618. Springer, Heidelberg (1997)
[11] Fritzke, B.: A growing neural gas network learns topologies. In: Advances in Neural Information Processing Systems (NIPS 1994), vol. 7, pp. 625–632. MIT Press, Cambridge (1995)
[12] Martinetz, T.M.: Competitive Hebbian learning rule forms perfectly topology preserving maps. In: ICANN 1993: International Conference on Artificial Neural Networks, Amsterdam, pp. 427–434. Springer, Heidelberg (1993)