Open Archive Toulouse Archive Ouverte (OATAO)
OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: http://oatao.univ-toulouse.fr/
Eprints ID: 11734


Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@inp-toulouse.fr
A Neural Network Strategy Applied in Autonomous Mobile Localization

Andrê Scolari Conceição, Caroline Ponzoni Carvalho, Eduardo Rath Rohr,
Daniel Porath, Diego Eckhard and Luís Fernando Alves Pereira

Abstract—In this article, a new approach to the problem of indoor navigation based on ultrasonic sensors is presented, where artificial neural networks (ANN) are used to estimate the position and orientation of a mobile robot. This approach proposes the use of three Radial Basis Function (RBF) Networks, where environment maps from an ultrasonic sensor and maps synthetically generated are used to estimate the robot localization. The mobile robot is mainly characterized by its real-time operation based on the Matlab/Simulink environment, where the whole necessary tasks for an autonomous navigation are done in a hierarchical and easy reprogramming way. Finally, practical results of real time navigation related to robot localization in a known indoor environment are shown.

I. INTRODUCTION

Methods for navigation and localization of mobile robots have been widely investigated in the control community, mainly in the robotic area in which mobile robots have the necessity to move themselves in an autonomous way. For autonomous navigation, robots must be able to interact with the environment around them whether familiar or unrecognized. In other words, they have to navigate, steer, and position themselves.

The localization problem has been receiving special attention over the years, with a focus on either a priori knowledge or completely unknown navigation environments, remaining an active topic of interest and allowing the proposition of new methods and techniques for the problem solution [1], [2], [3]. According to the characteristics of the navigation environment, different kinds of sensors have been employed to perform the localization task. For outdoor navigation, as an example, the use of a Global Positioning System (GPS) is common, since the accurate transmission/reception conditions between the mobile robot and the set of satellites responsible for the global coordinate determination could be provided [4], [5]. For indoor navigation, the mobile robot localization task can be realized through odometers placed on the traction wheels, or through special sensors embedded on the robot, as a bar code reader, video camera, among others, used for the recognition of natural or artificial landmarks spread in the navigation environment [6], [3], [7]. The localization techniques based on odometer sensors present a cumulative error drawback, while the main disadvantage concerning artificial landmarks is the previous arrangement of a set of marks on the navigation environment. Also, the integrity of each landmark must be preserved in order to assure a good localization performance. An alternative localization technique employs ultrasonic sensors or laser scanners to create distance maps from the robot to the environment’s obstacles. While the distance maps created by the laser scanners present reliable information, which is useful for localization tasks, ultrasonic sensors are commonly used for obstacle avoidance [8]. The usefulness of lasers scanners is partially justified by the high directionality and small wavelength of the lasers, making possible a precise distance measurement in a large number of reflective surfaces considering a wide range of incidence angles. Despite of the specular characteristic of most surfaces and the large aperture angle characteristics, intrinsically related with the physical properties of acoustic sensors, there are some researches which consider ultrasonic sensors as effective and low cost alternatives for the autonomous mobile robot global localization problem. A seminal article written by Elfes [9] presents a sonar-based mapping and navigation system employing a probabilistic approach to represent occupancy maps, observing the necessity of a sonar data preprocessing step to remove easily detectable incorrect readings. To improve the sonar readings reliability, some articles have been written exploiting some properties of the acoustic sensors. In [10], [11], a theoretical formulation for interpreting the sonar data based on the physical principles of acoustic propagation and reflection was presented.

Other authors prefer to handle the task of processing sonar data by using artificial neural networks (ANN) [12], [13]. There are two main approaches in which the research using neural network can be classified. In one hand, some researchers [12] use ANN to handle low level sensor information, such as differentiating the shape of the obstacle through the sonar readings. On the other hand, the ANN is also used to compare and match the mapping provided by the sensor’s readings and the environment map recorded in the robot.

In this article, an alternative methodology for localization
of a mobile robot is presented. The proposed technique is based on ANN to perform the matching between the mapping obtained by the raw sensor data and a set of maps previously known from some positions along the navigation environment.

This paper is organized as follows. In Section II the description of hardware and software of the mobile robot is presented. Section III presents the localization strategy elements: system localization scheme and the RBF training method. Experimental results of indoor navigation and localization are presented in section IV. Finally, the conclusions are drawn in section V.

II. MOBILE ROBOT ARCHITECTURE

A. Hardware

The mobile robot has a differential drive configuration. The two aligned wheels are driven by independent DC motors, while the third one is a free wheel just used for support, as shown in Fig. 1(a).

The world frame (Oxy) and the robot’s body frame are shown in Fig. 1(b). The mobile robot is described on a 2D plane in which a global cartesian coordinate system is defined. The position of robot in the plane is described by point \( P(x, y) \), between the two driving wheels, and by orientation angle \( \phi \). The Robot’s motion is controlled by its linear velocity \( (v) \) and rotational velocity \( (w) \). The motion of the robot can be described by the following kinematical model,

\[
\begin{align*}
\dot{x} &= v \cos(\phi) \\
\dot{y} &= v \sin(\phi) \\
\dot{\phi} &= w
\end{align*}
\]

Tasks performed by autonomous robots are usually complex; a large quantity of information must be collected about the environment in which the robot is working. For this reason it is necessary to use various sensors with different characteristics [2], [14]. In this context, the developed mobile platform uses ultrasonic range sensors and encoders to generate the information required by the operations of guidance, control, navigation and localization. The incremental optical encoders, coupled to the axle of each motor, provide measurements of the velocity and angular displacement of each driven wheel independently, giving information about the relative position of the vehicle. Information on the platform’s absolute position in the environment in which it will navigate is obtained by means of a sonar that makes periodic 360° scans of the navigation environment. This sonar is of simple construction, consisting of an ultrasonic range sensor assembled over a stepper motor that provides readings of the surrounding displaced environment in steps of 1.8 degrees. In the philosophy adopted for the electrical design, an ARM7 microcontroller manages the low level tasks, such as interfacing sonar and odometry measurements and implementing a PID controller in each driving wheel. A notebook disposed over the robot is responsible for the higher level tasks, i.e. motion planning and localization.

The Universal Serial Bus (USB) was used to exchange data between computer and microcontroller. The use of a USB allows a fast and reliable communication, while the compatibility with newer computers is ensured.

B. Software

The notebook over the robot runs MATLAB, which is worldwide recognized as a simulation environment with tools capable of reproducing an important class of dynamic processes, allowing users to perform the simplest mathematical calculations whilst extending to the possibility of reproducing industrial processes. MATLAB/Simulink is used to be mostly regarded as an environment for simulation, using tools whose function was to reproduce a physical model in graphical or descriptive form. However, tools for real-time execution and interface to hardware are also available, allowing the designer to model, simulate and execute a project in the same environment. The block diagram of the main navigation functions and the control loop of the robot are shown in Fig. 2.

The robot control environment based on MATLAB/Simulink is shown in Fig. 3.

The main blocks of the control are (see the numbers in Fig. 3):

1) Localization and navigation \( \leftrightarrow \) algorithms of localization and algorithms of planning and generation of trajectories;
2) Kinematic and inverse kinematic model \( \leftrightarrow \) calculate reference speeds and robot position;
3) Communication tools and controller \( \leftrightarrow \) receive sensor information and send control signs to the robot (PID controller);
4) Data debugging visualization and log of the robot state (position and velocities) and sensor data (ultrasonic sensor).

C. The Neural Network toolbox of Matlab/Simulink

This work proposes the use of Matlab/Simulink not only as a simulation tool, but principally for the real-time implementation of control techniques applied to mobile robots. In this philosophy, the user or designer makes use of all the resources for simulation and design of controllers that are built into Matlab/Simulink, and validates them directly in real time.

The localization strategy proposed in this paper uses Neural Network Toolbox of Matlab/Simulink for mobile robot localization in an indoor environment. We chose Artificial Neural Networks based on Radial Basis Functions (RBF) for having a faster training and to be an approximator of functions.

Radial Basis Functions (RBFs) are a universal approximator in that it can approximate arbitrarily well any multivariate continuous function [15]. The construction of a radial-basis function (RBF) network in its most basic form involves three entirely different layers. The input layer is made up of source nodes. The second layer is a hidden layer of high enough dimension, which serves a different purpose from that in a multilayer perceptron. The output layer supplies the response of the network to the activation patterns applied to the input layer [16].

III. STRATEGY OF LOCALIZATION

The strategy of absolute localization of this article is destined to indoor navigation, where the objective is to estimate the robot position \((x, y)\) and the robot orientation \((\varphi)\) in a known environment (see Fig. 4). The strategy of localization consists of four steps, see Fig. 5:

1) Environment mapping using an embedded ultrasonic sensor;
2) Estimate robot positions and robot orientations using three RBF networks, where its input data is the environment map and its output data are the estimations \((x_{1..3}, y_{1..3}, \varphi_{1..3})\);
3) Build synthetic maps composed from the estimated positions \((x_{1..3}, y_{1..3}, \varphi_{1..3})\);
4) Compare the environment map with synthetic maps.

The average errors between the synthetic maps and the environment map is calculated, in order to choose the best estimation for \((x_f, y_f, \varphi_f)\).

The average error is calculated as follows:

\[
Error(i)_{1..3} = \frac{1}{200} \sum_{j=1}^{200} \sqrt{(D_j - d_{i,j})^2}, \quad (4)
\]
where \( D_j \) is a vector that represents the sonar measurements. The synthetic maps composed from the estimated positions \((x_{1..3}, y_{1..3}, \varphi_{1..3})\) are represented by the \( d_{i,j} \) matrix. The best estimation for \((x_f, y_f, \varphi_f)\) is obtained through the comparison between the real and the synthetic maps which results in the smallest error.

A. RBF Networks Training

The training method of the RBF networks is based on environment maps from the ultrasonic sensor of the robot. These maps were built at 54 equally spaced positions around the environment, see Fig. 6, called reference grid.

The localization strategy proposes the use of three(3) RBFs. The idea is to divide the rotational space of 360 degrees in three sub-spaces of 120 degrees, where the environment maps are not similar. In this way, each RBF is responsible for the angular robot position in a sub-space of 120 degrees. This approach avoids the problem of similarity of the maps for angles near to 0 and 360 degrees. For RBF training, similar inputs demand similar outputs, in this case similar mappings do not have similar angular positions. This characteristic of the rotational space compromises the capacity of generalization of the RBF. For example, in Fig. 7 we can see the similarity of the maps at positions (210, 180, 0°) and (210, 180, 350°).

The training data consist of environment maps as input, and the robot localization \((x, y, \varphi)\) as output. Each environment map has 200 readings with step of 1.8 degrees. The maps used for the training had been organized in the following way:

- for each position of the reference grid, its environment map was rotated between 0 and 360 degrees, with step of 3 degrees (54 positions x 120 angles per position = 6480 maps);
- The maps with angle between 0 and 120 degrees were used for training the first RBF. The second one used maps with angle between 120 and 240 degrees, and the third one used maps with angle between 240 and 360 degrees (54 positions x 40 angles per position = 2160 maps for each RBF);
- the input data for each RBF has a dimension of 2160x200 and the output data has a dimension of 2160x3.

After the training phase the RBFs have the following configuration, shown in Table I:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>RBF 1</th>
<th>RBF 2</th>
<th>RBF 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>totalInputSize</td>
<td>200</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>totalLayerSize</td>
<td>1895</td>
<td>1897</td>
<td>1906</td>
</tr>
<tr>
<td>totalOutputSize</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Some considerations about the training strategy can be made. Preliminary, the training was tested with 5 and 4 networks (5 and 4 groups of maps), obtaining very similar results when compared with 3 networks (3 groups of maps). The training with 2 networks did not have good results, because for central points of the environment the RCDs were almost ambiguous for maps of different positions and orientations. Another test was conducted to verify the quantity of data to training the networks. The environment maps were made with steps of 6 and 10 degrees, but the errors the positions and orientation increased substantially.

IV. EXPERIMENTAL RESULTS

In this section, experimental results of real time navigation in the known indoor environment (see Fig. 4) are presented,
as well, the results of the strategy of localization and results of the internal odometry of the robot.

In most mobile robots, odometry is implemented by means of optical encoders that monitor the wheel revolutions of the robot’s wheels. The encoder data is then used to compute the robot’s offset from a known starting position. Odometry is simple, inexpensive, and easy to accomplish in real-time. The disadvantage of odometry is its unbounded accumulation of errors [17]. Because of the accumulation of errors, absolute position corrections are often necessary after as little as 10 meter of travel, and they are usually based on external measurements from localization systems. With this purpose, the proposed localization strategy is used to reset odometry errors along the robot path. So, the path was divided into seven parts, and the localization procedure was applied at the end of each part. Obviously, the robot can not move during the mapping process.

Fig. 8 shows the odometry of the robot, estimated robot locations, and real robot locations. The real values of the robot location were carefully measured by hand with a taped measure. The odometry correction is shown in Fig. 8(b), where after each correction of the robot position \((x, y)\) and orientation \((\phi)\), the odometry errors disappear. It reduces the navigation errors related to localization, and internal odometry of the robot.

Table II summarizes the robot locations, besides, the error between real and estimated positions \((x, y)\) and orientations \((\phi)\). The biggest errors obtained in the experiment are 0.13 (m) in \(X\) axis, 0.05 (m) in \(Y\) axis and 9 degrees of orientation.

<table>
<thead>
<tr>
<th>Table II</th>
<th>Localization Summary.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X) (m)</td>
<td>(Y) (m)</td>
</tr>
<tr>
<td>1.03</td>
<td>0.66</td>
</tr>
<tr>
<td>1.14</td>
<td>0.90</td>
</tr>
<tr>
<td>1.79</td>
<td>0.85</td>
</tr>
<tr>
<td>2.50</td>
<td>1.20</td>
</tr>
<tr>
<td>1.84</td>
<td>1.58</td>
</tr>
<tr>
<td>1.35</td>
<td>1.93</td>
</tr>
<tr>
<td>1.53</td>
<td>2.68</td>
</tr>
<tr>
<td>1.40</td>
<td>2.91</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS AND FUTURE WORKS

This article shows an alternative algorithm for localization of mobile robots in a structured environment using artificial neural networks. A set of ultrasonic measurements is acquired in different places of the navigation environment and used for training three different neural networks. Each neural network is responsible to cover a range of 120 degrees around the training points. Despite of the physical characteristics of the ultrasonic sensor, which sometimes results in erroneous distance measurements, the obtained results show the effectiveness of the proposed localization strategy.

The results were obtained using an experimental mobile robot with an embedded notebook, with all the control and localization features performed by a real-time Matlab/Simulink platform. The generality of the proposed methodology supports the use of different kinds of sensors based in a 360 degrees scan measurements. As a future work, the authors intend to replace the sonar sensor by a laser scanner, ameliorating the absolute localization results.

VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support of FAPESB, FAPERGS and CNPq.

REFERENCES


