NEURAL MOBILE ROBOT NAVIGATION BASED ON A 2D LASER RANGE SENSOR

D. Limón Marruedo and J. Gómez Ortega

Dpto. Ingeniería de Sistemas y Automática, Univ. de Sevilla
Escuela Superior de Ingenieros, Avda. reina Mercedes s/n, 41012 Sevilla, Spain.
Fax: +34-5-4555849, email: juango@cartuja.us.es

Abstract: This paper presents a navigation system for an autonomous mobile robot in a partially structured environment. The navigation objective is to follow a reference path avoiding unexpected fixed obstacles which are detected with an on board laser range sensor system. The system is based on two different modules which undertakes the robot control depending on a higher level supervision module. The first module deals with the path tracking problem and is based on a pure pursuit strategy. The second, implemented by a multilayer perceptron neural-network, deals with the obstacle avoidance problem where the sensor measurements are directly used as inputs for the neural network. This avoid a higher level pre-processing of the sensor data, which usually is computer time consuming, allowing real-time execution. The perceptron has been trained off-line in a supervised manner to reproduce the behaviour of an human expert in driving vehicles. Experimental results obtained when applying the navigation system to a LABMATE mobile robot are given in the paper.

Keywords: Mobile robots, navigation, obstacle avoidance, neural networks.

1. INTRODUCTION

Mobile robot navigation is a problem which has been profusely studied in the last years. It consists of driving a mobile platform autonomously avoiding the unexpected obstacles detected with an on board sensor system. If no reference path is specified, the problem is known as pure reactive navigation. On the contrary, when the working environment is structured and the uncertainty is not so high, a previous path planning stage can be considered and the navigation problem is reduced to a classic control problem with the desired path as the reference. This is called the path tracking problem. If the possibility of unexpected obstacles is considered, as it is in this paper, the problem is much more complex, involving sensor data processing and obstacle avoiding strategies.

In this paper an autonomous navigation system based on a neural network has been developed. A partially structured environment has been considered where a reference path is previously specified and an on board laser range sensor system is used to detect unexpected fixed obstacles.

Two general approaches can be encountered when only fixed obstacles are considered. In the first one, the mobile robot motion control and the obstacle avoidance are considered decoupled, using a two levels hierarchical strategy. When an obstacle is encountered, the higher level replans a new reference path which avoids the obstacle. The lower level drives the robot to follow the actual reference path, which can be the original path or the replanned path. In (Foux et al., 1993), (Sinha et al., 1992) and (Lin et al., 1994), this kind of strategy is used. In the second approach, these two problems are coupled. The mobile robot motion
controller is fed with sensor data from the environment. This information is considered in the computation of the control law. In this case, there is no geometrical path replanning. These methods are often based on optimal control techniques, where the control law is obtained from the minimization of a cost function, and where the proximity to any obstacle is also penalized (Feng and Krogh, 1991), (Gómez Ortega and Camacho, 1994) and (Gómez Ortega and Camacho, 1996).

In this paper the second approach has been considered. The neural based navigation system is directly fed with the laser sensor range measurements which are used as inputs for the neural network in order to detect and avoid unforeseen obstacles. Also, a reference path is specified and a pure pursuit strategy (Amidi, 1990) is used as path tracking method when no obstacles are detected near the mobile robot.

This paper is organized as follows: Section 2 describes the navigation system architecture and the neural network utilized. In Section 3, experimental results when applying the neural navigation system to a LABMATE robot are presented. The paper finishes with some conclusions.

2. THE NAVIGATION SYSTEM

The objective of the navigation system developed in this work consists of driving the mobile robot to follow a reference path from an initial point to a final one in a partially structured environment. Unexpected fixed obstacles are considered, which implies the use of an obstacle avoidance module. The navigation system is divided in two modules. The aim of the first one is to follow the reference path when the environment near the robot is free of obstacles. A pure pursuit strategy (Amidi, 1990) has been used for this purpose, where the linear velocity of the robot has been considered constant, being the robot steering velocity the only control signal. This strategy has been chosen because its simplicity, although other strategies could be used.

The second module of the navigation system deals with the unforeseen obstacles avoidance problem. The technique used is based on a reactive strategy and has been implemented by a multilayer perceptron artificial neural network.

The inputs of the neural network are directly the measurements of the laser range sensor system which avoids a high level processing of the sensor system data. This high level processing is usually needed in order to obtain, from the sensor direct measurements, information about the positions and sizes of the obstacles. This process is avoided with the strategy proposed in this paper which also allows to achieve real-time performance. The laser data has been previously coded in order to reduce the number of inputs of the network.

The third task that has to be carried out is to decide which of the two previous modules has to undertake the robot control at each moment. If the environment near to the robot is free of obstacles, then the Pure Pursuit path tracking module will undertake the control. If there is any obstacle, then the neural network will drive the robot. The data needed for making such a decision is just the laser sensor data from the environment, which are also the inputs for the obstacle avoidance Neural Network. Thus, this neural network has also been used as a decision module. The navigation system architecture is shown in figure 1, where \( \theta_{pt} \) is the robot angular velocity given by the path tracking module and \( \theta_{oa} \) is the robot angular velocity given by the obstacle avoidance module, being \( \theta \) the angle of the x axis of the local reference frame attached to the robot respect to the x axis of a global reference frame (see figure 2).

![Fig. 1. Navigation system architecture.](image1)

![Fig. 2. Reference frame.](image2)

2.1 The Pure Pursuit Based Path Tracking Module

This strategy is a tracking method in which the robot motion fits a circular arc to a goal point...
in the reference path which is called look ahead point. Therefore the path is tracked by repeatedly fitting different arcs to different look ahead points as the vehicle moves forward. The distance L from the robot to the goal point in the reference path is called look ahead distance, and it has been considered constant (see figure 3).

![Fig. 3. The pure pursuit Strategy.](image)

Then the curvature γ of the robot motion at each sample time is given by:

\[ \gamma = \frac{1}{R} = \frac{2\pi}{L^2} \]

where R is the radius of curvature and x is the abscissa of the look ahead point in a local reference frame fixed to the robot (see figure 3). The steering velocity of the robot is obtained as follows:

\[ \theta = \gamma V \]

being V the linear velocity of the robot, which is considered to be constant.

2.2 The supervision and obstacle avoidance module

As it was mentioned above, an unique module has been utilized for both supervision and obstacle avoidance tasks. This can be done because the inputs needed for both tasks are the same, that is, the sensor data from the environment.

The inputs of this neural network based module are directly the measurements obtained from the laser range system. The output of this sensor system is a set of range measurements uniformly distributed over a range of 180 degrees. 360 measurements are provided by this sensor which gives a resolution of 0.5 degrees in the angular direction.

For the obstacle avoidance strategy implemented in this work, only a coarse information of the obstacles detected in the environment is needed. Considering also that, if each of the 360 range measurements given by the laser system were used as inputs for the neural network, the training stage would be quite complicated, a coarse coding of this sensor data has been included in the navigation module.

For implementing this strategy only a near area of the mobile robot environment is considered for obstacle detection. This area has been called receptive area and includes an angular sector shaped region around the mobile robot front side (see 4). This area is discretized into a virtual grid of cells, where Na is the number of cells in the angular direction and Nr is the number of cells in the radial direction. For coarse coding only completely occupied or empty cells are considered.

![Fig. 4. Receptive area definition.](image)

A binary matrix is then associated with the virtual grid so that each term of the matrix is related to a cell of the grid of the receptive area. A term of the matrix is set to 1 if its related cell is totally or partially occupied by an obstacle, and is set to 0 in other case. The information for this setting is obtained from the set of laser range measurements. All the cells beside an occupied cell in the radial direction are also considered to be occupied (which is considered the worst case). An example is shown in figure 5.

![Fig. 5. Receptive area associated matrix.](image)

Once the associated matrix is formed, a codification procedure is applied to it in order to reduce the number of inputs to the neural network. This procedure consist of two sequential processes: a radial and an angular codification.

The radial codification consists of adding the terms of each column of the matrix. Then, a col-
umn of the matrix represents an angular portion of the receptive area in the radial direction. The value of the sum of each column can be related with the proximity between the robot and the obstacle in the radial direction associated with that column. Thus, the matrix is transformed in a vector of proximity values. For the example shown in figure 5, the result of this codification is the vector \([3 \, 3 \, 3 \, 2 \, 0 \, 0]\), where a value of 3 indicates that in that radial direction the obstacle is nearer to the robot than in the direction with a value of 2, and that in directions with a 0 value no obstacles are detected. This process reduces the sensor information from \(N_a \times N_r\) terms of the matrix to \(N_a\) terms of the vector of proximity obtained, keeping the angular resolution unchanged.

The second codification process is in the angular direction and is based on a distributed representation of the information called Coarse Coding, developed by Hinton et al. (1986). This procedure uses as inputs the proximity vector components obtained in the previous radial codification phase (Meng and Picton, 1992). In order to reduce the number of data needed to describe the obstacle, the data contained in this vector is recoded, joining several of its components into one new value which is obtained as the sum of the original joined components. Because no reduction in the accuracy of an obstacle description is desired, an overlapping strategy is used (see figure 6). Using this method, the number of inputs is reduced to \((N_a+1)/2\). The result of the process applied to the example of figure 5 is shown in figure 6.

2.2.1. Symmetry analysis

The obstacle avoidance problem may be considered as a symmetrical one with respect to the \(y\) axis of the local reference frame fixed to the robot, in such a way that the number of training patterns needed for the neural network training phase is reduced. The symmetry analysis can only be applied to the second output of the network.

An example of the symmetry analysis is shown in figure 7 where the relative position of the obstacle is symmetrical with respect to the \(y\) local axis. When an input pattern is symmetrical to one of the learned patterns, the input data to the network is set to the learned pattern. Afterwards, the outputs generated by the neural network will be symmetrically changed before applying the control actions on the mobile robot (see figure 8). Therefore, the neural network should learn only one of the symmetrical patterns, which simplify the training process.

Fig. 7. Symmetry analysis.

Fig. 8. Neural network inputs/outputs generation.

2.2.2. Pattern generation and neural network training

The training of the neural network has been carried out off-line in a supervised manner. The set of training patterns has been obtained using an heuristical procedure, with a simulation module where the user selects, for different situations of
the reference path and obstacles positions, the proper control actions. The following non-linear model of a differential drive vehicle kinematics has been used for the simulations:

\[
\begin{align*}
\theta(k+1) &= \theta(k) + \frac{\dot{\theta}(k-1)T}{\cos(\theta(k) + \dot{\theta}(k-1)T)}
\end{align*}
\]

\[
\begin{align*}
x(k+1) &= x(k) + \frac{V(k-1)}{\theta(k-1)} \{ \cos(\theta(k) + \dot{\theta}(k-1)T) - \cos(\theta(k)) \}
\end{align*}
\]

\[
\begin{align*}
y(k+1) &= y(k) + \frac{V(k-1)}{\theta(k-1)} \{ \sin(\theta(k) + \dot{\theta}(k-1)T) - \sin(\theta(k)) \}
\end{align*}
\]

\[
\begin{align*}
\dot{\theta}(k-1) &= R \frac{\omega_r(k-1) - \omega_l(k-1)}{2W}
\end{align*}
\]

\[
\begin{align*}
V(k-1) &= R \frac{\omega_r(k-1) + \omega_l(k-1)}{2}
\end{align*}
\]

where \( x, y, \theta \) are defined as shown in figure 2. \( T \) is the sample interval and \( W \) is the half-distance between the wheels, which value has been estimated to be 168 mm. \( V \) is the linear velocity of the mobile robot, \( \dot{\theta} \) is the steering speed, and \( \omega_r, \omega_l \) are the right and left wheel angular velocities (which are considered to be constant for each sample interval) and the wheel radius, respectively. These equations are valid in the case of a steering speed \( \dot{\theta}(k-1) \neq 0 \). In the case of a linear trajectory, the equations of motion are given by:

\[
\begin{align*}
\theta(k+1) &= \theta(k)
\end{align*}
\]

\[
\begin{align*}
x(k+1) &= x(k) - VT \sin \theta(k)
\end{align*}
\]

\[
\begin{align*}
y(k+1) &= y(k) + VT \cos \theta(k)
\end{align*}
\]

The neural network has been trained using a Resilient Propagation (Riedmiller and Braun, 1993) algorithm.

3. RESULTS

The proposed navigation system has been tested with the LABMATE mobile robot, shown in figure 9, designed for indoor environments. Its dimensions are 0.8 x 0.8 m, with a weight of 50 Kg. The maximum attainable linear velocity is 1000 mm/s and the maximum acceleration is 1000 mm/s². A serial link RS-232 is used for the communication between the host computer, where the neural network navigation system is executed, and the lower level processors located onboard the mobile robot which control the wheels angular velocities \( \omega_r, \omega_l \).

A SICK PLS 200 laser range sensor has been used for obtaining the distance measurements. This device scans 180° in 80 ms, with an angular accuracy of 0.5° and a maximum distance range of 50 m. For the experiments, a receptive area with an angle of 120°, a maximum radius of 2 m, a minimum radius of 0.5 m and a grid of 7 x 3 cells have been considered. A sample time of 0.5 seconds, a linear velocity of 150 mm/sec and a maximum steering velocity of 16°/sec have been chosen for the experiments. Using the off-line simulation module, a set of 77 training patterns were obtained. A multilayer perceptron with 3 layers and 40 units in the hidden layer has been used for implementing the obstacle avoidance module. Figures 10 and 11 show two experiments carried out in the laboratory.

![LABMATE mobile robot](image)

Fig. 9. LABMATE mobile robot.

![Experimental result (a)](image)

Fig. 10. Experimental result (a).
The environment depicted in both figures have been drawn from the laser range sensor measurements. Figure 10 shows an experiment where the mobile robot has to follow a reference path along two narrow corridors joined by a turn of small curvature radius. Several unexpected fixed obstacles have been placed in the environment. For this test, a set of via points joined by linear segments has been used as the reference path. In figure 11, only one goal point has been defined which leads to an almost pure reactive behaviour for the mobile robot. Both tests demonstrate that the navigation system proposed in this work gives quite acceptable results and a high degree of robustness.

Fig. 11. Experimental result (b).

4. CONCLUSIONS

In this paper, a navigation system for an autonomous mobile robot has been presented. This system has two modules, one based on a pure pursuit path tracking strategy which is enabled when no obstacles are near the robot, and another based on a neural network which deals with the obstacle avoidance task. Also, a supervision module, which selects the control mode between the two mentioned above, has been developed. A laser range sensor system has been used for distance measurements. The module based on a neural network has been trained using a set of patterns obtained off-line directly from the control actions given by the user in a simulation module.

The navigation system has been experimentally tested with a LABMATE mobile robot. The experiments presented in the paper demonstrate the good performance of the proposed mobile robot navigator.

5. REFERENCES


