Takagi-Sugeno Based Controller for Mobile Robot Navigation

S. Kermiche, M.L. Saidi, H.A. Abbassi and H. Ghodbane  
Automatic and Signal Laboratory, Faculty of Engineering,  
Annaba University, B.P.12 Annaba, Algeria

Abstract: This study presents a solution for the problem of learning and controlling a mobile robot in the presence of fixed obstacles. The objective is to move the robot from an initial position (source) to a final position (target) without collision. Thus, we propose an approach based on potential fields principle, we define the target as an attractive pole (given as a vector directly calculated from the target position) and the obstacle as a repulsive pole (a vector derived by using fuzzy logic techniques). The linguistic rules, the linguistic variables, the membership functions... etc are the parameters to be determined for the fuzzy controller conception. A learning method based on gradient descent for the self-tuning of these parameters is introduced. Therefore, it is necessary to have an expert person for moving the robot manually. During this operation of training, the robot moves and memorizes the data (inputs and outputs). This operation is used to find the controller parameters in order to reach the desired outputs for given inputs.

Key words: Robot mobile, fuzzy logic control, obstacle avoidance method, descent method, self tuning

INTRODUCTION

The control of mobile robot involves several tasks such as perception, path planning and path tracking. The estimation of the mobile robot position can be performed using several sources of data including the perception system. The general purpose of a robot path planner is to find a trajectory from a start position to a goal position with no collisions while minimizing a cost measure. The objective is to follow a previously defined path by taking into account the actual position and the constraints imposed by the robot architecture and the environment. Therefore, the autonomous navigation of a mobile robot imposes a perfect knowledge of the environment in order to detect and avoid the eventual obstacles. This knowledge data base is collected during the teaching operation.

The fundamental problem toward of mobile robot control is to guide the motion until a desired target with no collision. Some methods for generating collision-free paths are adapted from mobile robots (Faverjon, 1987; Sharir and Schorr, 1984; Lozano-Perez, 1983; O'Dunlaing and Yap, 1985).

The robot motion control is based on the priori known information or determined at each step.

NAVIGATION PROCEDURE

In a sufficiently large empty space, a mobile robot can be driven to any position with any orientation; hence the robot's configuration space has four dimensions, two for translation, one for rotation and one for the steering angle.

Let \((x, y, \Phi, \theta)\) denote the configuration of the robot, parameterized by the location of the front wheels. The kinematic model of the mobile robot can be represented by Yang et al. (2004), Baturome et al. (2004), Zhao and Bement (1992)

\[
\begin{align*}
\dot{x} &= v \cos \theta \\
\dot{y} &= v \sin \theta \\
\dot{\theta} &= \frac{v}{l} \tan \Phi
\end{align*}
\]  

(1)

Where \(v\) corresponds to the forward velocity for the vehicle body with respect to the horizontal line is \(\theta\), the steering angle with respect to the vehicle body is \(\Phi\),\((x, y)\) is the location of the center point of the front wheels and \(l\) is the length between the front and the rear wheels.

The steering angle of the front wheel is adjusted by using a fuzzy controller. We distinguish three types of wheels:

- Fixed wheel, its position is characterized by four constants \((\alpha, \beta, l, r)\) and the rotation angle \(\Phi(t)\).

The two components of the contact speed are given by the following constraints:

On the wheel plan (Kang et al., 1999; Lallemand and Zeghloul, 1994):

Corresponding Author: H.A. Abbassi, Automatic and signal Laboratory, Faculty of Engineering, Annaba University, B.P.12 Annaba, Algeria
Orthogonal to the wheel plan:

\[
\begin{pmatrix}
\cos(\alpha + \beta) & \sin(\alpha + \beta) & 0
\end{pmatrix}
\begin{pmatrix}
\phi(t) & \zeta(t)
\end{pmatrix}
= 0
\]  

(2)

- Steering wheel, its position is characterised by three constants (l, a, r) and the movement by two variables \(\phi(t)\) and \(\zeta(t)\).
- Castor wheel.

The method used here consists to control the robot motion from an initial position (source) to a final position (target) with no collision.

We suppose in our work that:

- The mobile robot, the obstacle and the target can take any position in the workspace;
- Only one obstacle exists in the robot environment;
- The robot is controlled by its orientation.

The mobile robot moves in a field of forces, an attractive force (target) and a repulsive force (obstacle), the resultant of these forces determines the robot orientation (Khatib, 1986).

The Fig. 1 shows the mobile robot, the target position and the obstacle in the workspace.

- \(V_{at}\): attractive vector
- \(V_{rep}\): repulsive vector
- \(V_{or}\): orientation vector

**COLLISION AVOIDANCE STRATEGY**

A fuzzy system is a system based on the concepts of approximate reasoning: linguistic variables, fuzzy propositions, linguistic if-then rules.

The goal is to realize a fuzzy controller able to evaluate the repulsive force (vector) \(V_{rep}\) characterizing the actual relative position of the obstacle (Khatib, 1986; Kermiche and Abbassi, 2004).

The controller has two inputs and one output, the inputs are the observation angle \(\theta_{os}\) and the distance \(d_{os}\) towards the obstacle, the output is the repulsive vector \(V_{rep}\) (Fig. 2).

The orientation angle depending on \(V_{or}\) is the input of the robot and its outputs are the coordinates \((x, y)\) and the direction \(\theta_{r}\).

**Fuzzification:** The fuzzification module performs two tasks:

- Input normalisation, mapping of input values into normalised universes of discourse,
- Transformation of the crisp process state values into fuzzy sets, in order to make them compatible with the antecedent parts of the linguistic rules that will be applied in the fuzzy inference engine.

**Fuzzification of the angle \(\theta_{os}\):** We suppose that the robot can perceive an obstacle in a direction inside the interval \([-90^\circ, 90^\circ]\). The membership function is represented by seven fuzzy subsets of Gaussian ash shown in Fig. 3.
**Fuzzification of the distance \( d_{sr} \):** We admit that the robot can detect an obstacle from a distance of 30 units (Fig. 4).

**Fuzzification of the repulsive angle \( \theta_{rs} \):** The membership function of the repulsive angle has a constant form belonging to the interval \([-135^\circ, 135^\circ]\).

**Inference:** Let \( x_1, x_2, ..., x_n \) be linguistic variables on the input space \( X = X_1 \times X_2 \times ... \times X_n \) and \( y \) be a linguistic variable (or a real variable) on the output space \( Y \); then two forms of fuzzy inference rules by the fuzzy "IF... THEN..." rule model can be described as follows (Shi and Mizumoto, 2000):

**Form (1):** Fuzzy Inference Rules by Product-Sum-Gravity Fuzzy Reasoning Method.

The fuzzy inference rules are defined as:

**Rule 1:** IF \( x_1 \) is \( A_{i1} \) and \( x_2 \) is \( A_{i2} \) and... and \( x_n \) is \( A_{in} \) then \( y \) is \( B_i \) \hspace{1cm} (3)

**Rule 2:** IF \( x_1 \) is \( A_{i1} \) and \( x_2 \) is \( A_{i2} \) and... and \( x_n \) is \( A_{in} \) then \( y \) is \( B_j \) \hspace{1cm} (4)

**Rule \( n \):** IF \( x_1 \) is \( A_{in} \) and \( x_2 \) is \( A_{in} \) and... and \( x_n \) is \( A_{in} \) then \( y \) is \( B_n \) \hspace{1cm} (5)

Where \( A_i (j = 1, 2, ..., m; i = 1, 2, ..., n) \) and \( B_i \) are fuzzy subsets of \( X_i \) and \( Y \), respectively and the subscript \( i \) corresponds to the \( i \)th fuzzy rule.

**Form (2):** Fuzzy Inference Rules by Simplified Fuzzy Reasoning Method.

The fuzzy inference rules are defined as:

**Rule 1:** IF \( x_1 \) is \( A_{i1} \) and \( x_2 \) is \( A_{i2} \) and... and \( x_n \) is \( A_{in} \) then \( y \) is \( Y_i \) \hspace{1cm} (6)

**Rule 2:** IF \( x_1 \) is \( A_{i1} \) and \( x_2 \) is \( A_{i2} \) and... and \( x_n \) is \( A_{in} \) then \( y \) is \( Y_j \) \hspace{1cm} (7)

Where \( A_j (j = 1, 2, ..., m; i = 1, 2, ..., n) \) is a fuzzy subset of \( X_i \) and \( Y_j \) is a real number on \( Y \) (Faerjon, 1987).

For examples the representations of these rules would then be constructed as follows:

IF \( \theta_{rs} \) is \( LL \) and \( d_{sr} \) is \( S \) then \( \theta_{rs} \) is \( APP \) OR

IF \( \theta_{rs} \) is \( LL \) and \( d_{sr} \) is \( M \) then \( \theta_{rs} \) is \( TPP \) OR

IF \( \theta_{rs} \) is \( LR \) and \( d_{sr} \) is \( L \) then \( \theta_{rs} \) is \( EZ \).

The rules are shown in Table 1.

**DEFFUZZIFICATION**

The defuzzification module performs the conversion of the union of modified fuzzy sets into a crisp output value followed by the denormalisation of this value.

The height method is the simplest and fastest one because only peak values of the modified fuzzy sets are taken into consideration. The resulting crisp output is the weighted sum of the peak values with respect to the heights of the modified fuzzy sets.

It is interesting to notice that for this type of defuzzification, we do not need to define the widths of the membership functions. It follows that a set of output membership functions can be defined as shown in Fig. 5. This type of membership functions is called singletons. This definition corresponds to the special case of Takagi and Sugeno's controller.

![Membership function of the repulsive angle \( \theta_{rs} \)](image)

---

**Table 1: Fuzzy rule table**

<table>
<thead>
<tr>
<th>( \theta_{rs} )</th>
<th>LL</th>
<th>ML</th>
<th>SL</th>
<th>EZ</th>
<th>SR</th>
<th>MR</th>
<th>LR</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d_{sr} )</td>
<td>S</td>
<td>APP</td>
<td>MP</td>
<td>AGP</td>
<td>TGN</td>
<td>AGN</td>
<td>MN</td>
</tr>
<tr>
<td>L</td>
<td>TPP</td>
<td>APP</td>
<td>MP</td>
<td>GN</td>
<td>MN</td>
<td>APN</td>
<td>TPN</td>
</tr>
</tbody>
</table>

**Rule \( n \):** IF \( x_1 \) is \( A_{i1} \) and \( x_2 \) is \( A_{i2} \) and... and \( x_n \) is \( A_{in} \) then \( y \) is \( y_n \) \hspace{1cm} (8)

---

1840
ADJUSTABLE FUZZY CONTROLLER

The controller is based on Sugeno inference method and the defuzzification uses height method. The controller’s output is function of linguistic variables (Foulloy and Galichet, 1995; Buhlner, 1997).

**Linguistic rules**: We suppose that the controller has M+K linguistic variables, M inputs, K outputs and N linguistic rules (6, 7 and 8).

**Learning method**: It is used to determine the controller parameters values in order to reach the desired outputs for given inputs (Fig. 6).

**General algorithm**: The model is described by the following equations (Godjevac, 1997):

The parameters estimation consists to minimize the criterion V:

\[ V = E \left( \left\langle \hat{e}(t) \right\rangle \right)^2 \]  

Or

\[ V = E \left( \frac{1}{N} \sum_{i=1}^{N} \left\langle \hat{e}(t) \right\rangle \right)^2 \]  

E: means  
N: number of iterations  
e(t): learning error vector

In order to minimise the learning error, we will find the minimum of the criterion function V. It can be found by solving:

\[ -V_{c} V = \begin{bmatrix} \frac{\partial V}{\partial z_{1}}, \ldots, \frac{\partial V}{\partial z_{p}} \end{bmatrix} = 0 \]  

\[ -\nabla_{V} V = \begin{bmatrix} -\frac{\partial V}{\partial z_{1}}, \ldots, -\frac{\partial V}{\partial z_{p}} \end{bmatrix} = 0 \]  

Then

\[ \tilde{z} = (z_{1}, \ldots, z_{M}, b_{1}, \ldots, b_{K}, c_{1}, \ldots, c_{K}) \]

The number of parameters to adapt is:

\[ P = 2N \times M + K \times N \]

The vector which minimize the criterion function is given by:

\[ \frac{\partial V}{\partial a_{ni}}, \frac{\partial V}{\partial a_{nim}}, \frac{\partial V}{\partial b_{ni}}, \frac{\partial V}{\partial b_{nim}}, \frac{\partial V}{\partial c_{ni}}, \frac{\partial V}{\partial c_{nim}} = 0 \]

And the recursive (learning) rules:

\[ a_{nm}(t+1) = a_{nm}(t) - \Gamma_{a} \frac{\partial V(z)}{\partial a_{nm}} \]  

\[ b_{nm}(t+1) = b_{nm}(t) - \Gamma_{b} \frac{\partial V(z)}{\partial b_{nm}} \]  

\[ c_{nk}(t+1) = c_{nk}(t) - \Gamma_{c} \frac{\partial V(z)}{\partial c_{nk}} \]

If the membership functions of the controller are Gaussians, then the partial derivatives of the criterion V are:

\[ \frac{\partial V}{\partial a_{nm}} = \sum_{k=1}^{K} \left( y_{k} - y_{nk} \right) \frac{u_{nk}}{b_{nm}} (c_{nk} - y_{nk}) x_{nk} - a_{nm} \]
\[
\frac{\partial V}{\partial w_{mn}} = \sum_{k=1}^{K} (y_k - y_{ak}) \frac{u_n}{\sum_{n=1}^{N} u_n} (c_{ak} - y_k) (x_m - a_{mn})^{2} b_{mn}^{-1}
\]

\[
\frac{\partial V}{\partial c_{ak}} = (y_k - y_{ak}) \frac{u_n}{\sum_{n=1}^{N} u_n} \tag{17}
\]

The adaptation of the parameters of the Gaussians and weights is done by

\[
a_{mn}(t+1) = a_{mn}(t) - \gamma_s \frac{u_n}{\sum_{n=1}^{N} u_n} \frac{x_m(t) - a_{mn}(t)}{b_{mn}(t)^2}
\]

\[
\sum_{k=1}^{K} (y_k(t) - y_{ak}(t)) (c_{ak}(t) - y_k(t))
\]

\[
b_{mn}(t+1) = b_{mn}(t) - \gamma_b \frac{u_n}{\sum_{n=1}^{N} u_n} \frac{(x_m(t) - a_{mn}(t))^2}{b_{mn}(t)^3}
\]

\[
\sum_{k=1}^{K} (y_k(t) - y_{ak}(t)) (c_{ak}(t) - y_k(t))
\]

\[
c_{ak}(t+1) = c_{ak}(t) - \gamma_c \frac{u_n}{\sum_{n=1}^{N} u_n} (y_k(t) - y_{ak}(t))
\]

**Linguistic rules extraction:** The problem of the linguistic rules extraction is to convert the parameters \((a_{mn}, b_{mn}, c_{ak})\) at the end of the adjustment to linguistic values.

To solve this problem we compare the membership functions defined by \(a_{mn}\) and \(b_{mn}\) with preset ones whose linguistic terms belong to a set of trajectories provided by an expert person.

The controller parameters are identified by an off-line procedure based on the memorized data and the initial parameters.

- Number of inputs \(M = 2\)
- Number of outputs \(K = 1\)
- Number of rules \(N = 21\)

**SIMULATION**

The source, the target and the obstacle positions are specified.

Test:

Before training the robot moves from a start configuration to a goal configuration without collision (Fig. 7).
Figure 9 describes the robot trajectory with collision avoidance after training.

The Fig. 10 and 11 represent the adjustment of the center and the width of each membership function after the training.

We note that the trajectory after training is optimal compared to the one before training (Fig. 11).

We have observed that after training, the concentration of singletons is located on left side. This is due to the avoidance of obstacle is done on the right side (Fig. 12).

**CONCLUSIONS**

In this study, we have presented a solution to the problem of trajectory tracking without collision.

The robot motion depends on the potential field approach.

The robot moves in a field of forces where the goal position is an attractive pole and where the obstacle is a repulsive pole. The attractive force is calculated from the goal position and the repulsive force is determined by a fuzzy logic.

The robot has to follow a trajectory specified by the operator from a start configuration to a goal configuration, which goes through a fixed obstacle.

When a potential collision with the obstacle is detected, the collision avoidance redirects the robot motion thanks to the repulsive force in order to generate a new collision free path.

The method has been tested on robot and the results are very satisfactory.

**REFERENCES**


