Self-Trained Automated Parking System

R.J. Oentaryo and M. Pasquier

Centre for Computational Intelligence, Nanyang Technological University,
School of Computer Engineering, Blk N4 #02a-32, Nanyang Avenue, Singapore 639798.
Email: pasquier@pmail.ntu.edu.sg

Abstract

This paper presents part of the research work carried out at the Centre for Computational Intelligence at NTU to develop novel technologies for the routing, navigation, and control of intelligent cars. One objective is to endow the cars with the ability to autonomously drive on various types of roads and realize manoeuvres such as reverse and parallel parking, three-point turns, etc. Our approach is to design a self-training system that makes use of human expertise to automatically derive a working car control system. A new neuro-fuzzy architecture known as the GenSoYager Fuzzy Neural Network has been realized and integrated with our car-driving simulator for training and testing purposes. The GenSoYagerFNN has proven so far superior to other trained networks in detecting parking slots and accomplishing reverse parking manoeuvres. The approach described has also been validated using a microprocessor controlled model car.

1. Introduction

The dramatic growth in automotive technology during the past century has culminated in today’s society of motorization, the safety of which is more than ever an imperative concern. Because human errors remain the primary cause in most traffic accidents, the development of in-car technologies for monitoring, prevention, and guidance, has become a major research area, with the objective of reducing the burden of the human driver, increasing traffic capacity, and providing both safe and smooth vehicle operations. Our group at the Centre for Computational Intelligence (formerly Intelligent Systems Laboratory) at NTU has long been working on this topic and particularly the realization of autonomous driving systems for road vehicles [2][3][8].

This paper describes our latest system, the Generic Self-Organising Yager Fuzzy Neural Network (GenSoYagerFNN) and its application to autonomously drive a car on a road track and realize manoeuvres such as reverse parking and three-point turns. Our approach consists of designing a self-training system that can make use of human expertise to automatically extract objective rules for controlling the car. The assumption is that driving is a continuous decision-making process that can be articulated as a set of rules relating sensory input to control output. A fuzzy control system is selected to model the intrinsic vagueness of the information available (speed, distance, dynamic environment variables). This fuzzy rule-based system is then implemented atop a neural network architecture, which provides the ability to learn, recall, generalize and adapt from training data. The resulting fuzzy neural network, or neuro-fuzzy system, possesses the merits of both techniques, that is, in the aspect of learning and optimization capabilities as well as connectionist structure, the human reasoning ability and ease of incorporating expert knowledge. At the same time, the shortcomings of each approach are alleviated, namely the design issue for the fuzzy rule-based system (selection of membership functions, fuzzy rules identification) and the network’s black-box nature (opacity of the intermediate layers).

Inference in the GenSoYagerFNN is modelled after the Yager reasoning scheme [10], which computes the dissimilarity of the inputs with the rule antecedents to derive the degree of dissimilarity with the rule consequent, thus arriving at the output. The main benefit over the classic Compositional Rule of Inference (CRI) [11], which we used previously [3][4][9], is that when the input matches the antecedent exactly, the resultant output matches the consequent exactly. Intuitively, the Yager inference rule is closer to human reasoning, and it emerges superior to existing techniques. Indeed the GenSoYagerFNN has proven so far superior to other trained networks in detecting parking slots and accomplishing reverse parking manoeuvres. Lastly, it must be noted that, while the work reported is done in simulation, the approach has also been validated recently using a microprocessor controlled model car.

2. GenSoYager Fuzzy Neural Network

The proposed GenSoYagerFNN is based on another in-house connectionist structure, namely the Generic Self-Organizing Fuzzy Neural Network (GenSoFNN) [7][9]. It is able to automatically formulate fuzzy rules from given numerical training data and to maintain a consistent rule set by ensuring that each fuzzy label in the input/output dimensions is uniquely represented by only one cluster (fuzzy set). Each input fuzzy set can still contribute to the antecedent of more than one fuzzy rule. The GenSoYagerFNN has strong noise tolerance
capability by employing a new clustering technique known as Discrete Incremental Clustering (DIC) [7]. Under this framework, noisy/spurious data that is found to have poor correlation to genuine or valid data will have separate clusters created for them. In addition, DIC does not require any prior knowledge of the number of clusters in the problem domain. These features are the reason why GenSoYagerFNN was selected over other implementations of the Yager inference [10][5].

The training cycle of the GenSoYagerFNN consists of three phases: self-organizing, rule mapping, and parameter learning, which all take place in a single pass of the training data, thus allowing for online use of the system. The well-known back-propagation learning [6] based on the negative gradient descent is employed during the last phase to tune the network parameters. As shown in Figure 1, the GenSoYagerFNN consists of five layers of nodes, each of which has finite “fan-in” and finite “fan-out” connections. The number of nodes in each layer is denoted as “n_l”, where \( l \in \{1, \ldots, 5\} \). Each of the input nodes \( IV_l \) in layer 1, \( l \in \{1, \ldots, n_1\} \) has a single input \( x_i \). The number of input attributes of the data set used to train the network determines the number of input nodes “n_1”, which are represented as a vector \( X^{n_1} \). Similarly, each of the output nodes \( OV_m \), where \( m \in \{1, \ldots, n_5\} \), computes a single output \( y_m \), and all outputs with respect to \( X \) are represented as a vector \( Y^{n_5} \).

The GenSoYagerFNN uses a supervised training scheme to automatically formulate the fuzzy rules from the training data and to tune the system parameters. The vector \( D = [d_1, \ldots, d_m, \ldots, d_n] \) represents the desired network outputs. Prior to training, the GenSoYagerFNN only has layer 1 and layer 5 nodes. The hidden layers that contain the input term nodes (layer 2), the rule nodes (layer 3) and the output term nodes (layer 4) are created and tuned as training progresses. The trainable weights of the network (enclosed in rectangular blocks in Figure 1) can be found in layers 2 and 5, which characterize the input and output fuzzy sets, respectively. The weights of the remaining network connections are unity. The input nodes in layer 1 may have different number of input terms. For input node \( IV_l \), the number of input term nodes is denoted as \( J_l \), and the total number of layer 2 nodes is \( n_2 = \sum_{l=1}^{5} J_l \).

Each node \( R_k \) in layer 3 is a rule node, where \( k = \{1, \ldots, n_3\} \) and “n_3” is thus the total number of fuzzy rules in the GenSoFNN. Each output node \( OV_m \) in layer 5 can have a different number \( L_m \) of output terms, hence the total number of layer 4 nodes is given as \( n_4 = \sum_{m=1}^{5} L_m \).

3. Yager Reasoning Scheme

Fuzzy inference extends the conventional modus ponens rule [1], which states that the proposition \( Y \) is \( B \) can be deduced from the propositions:

- \( P_1: \text{IF } X \text{ is } A \text{ THEN } Y \text{ is } B \),
- \( P_2: X \text{ is } A \)

The proposition \( P_1 \) concerns the joint fuzzy variable \((X, Y)\) and is characterized over the cross product space \( U \times V \) by a possibility distribution, as per Equation (1).

\[
\prod_{(X,Y)} = R
\]

(1)

Figure 1: Structure of the GenSoYagerFNN
There are two possible interpretations of the fuzzy relation $R$, based on the conjunctive and implication-based (i.e., disjunctive) models, respectively. The Compositional Rule of Inference (CRI) \[11\] scheme adopts the first approach, while the Yager rule of inference \[10\] adopts the other. Accordingly, the membership function for $R$ is computed using either of Equations 2 or 3, which correspond to the logical transition of $P_1$ interpreted in the two different ways. The $T$-norm “$T$” and $T$-conorm “$S$” stand for general definitions of the conjunctive and disjunctive fuzzy set operators, respectively, most common of which are the $\min$ and $\max$ operators.

\[
\mu_R(x, y) = T\{\mu_A(x), \mu_B(y)\} 
\]

\[
\mu_R(x, y) = S\{(1 - \mu_A(x)), \mu_B(y)\} 
\]

It can be noticed that Equation 3 corresponds to the statement ‘$\neg A \cup B$’, which is essentially the same as ‘$A \rightarrow B$’ in crisp logic. The implication-based model of fuzzy relation (i.e., the second approach) is exactly the core concept that underpins the Yager Reasoning Scheme employed by the proposed GenSoYagerFNN.

4. Operations of the GenSoYagerFNN

The proposed GenSoYagerFNN consists of five layers of neurons, the operation of which is elaborated in the following sections.

Fuzzification Layer (#1) – This layer consists of input nodes which function as singleton fuzzifiers that perform fuzzification of the absolute-valued inputs presented to the network. Fuzzification is required for the Yager inference engine to make use of the fuzzified inputs and compute the appropriate fuzzified outputs.

Antecedent Layer (#2) – The fuzzified inputs from layer 1 are compared against the corresponding input labels that form the antecedents of the fuzzy rules in the GenSoYagerFNN. The antecedent derivation in layer 2 computes a dissimilarity measure, which is essentially the negation of membership values of the inputs with respect to the input fuzzy sets.

Rule Layer (#3) – Nodes in layer 3 model the fuzzy rules in the GenSoYagerFNN. Each node computes, and presents as its output, the degree of fulfilment (i.e., the overall similarity) of the current inputs with respect to the antecedents of the fuzzy rule it represents.

Consequent Layer (#4) – Layer 4 consists of the output term nodes that represent the output fuzzy sets of the consequent of the rules in layer 3. Each output term node may be connected to multiple fuzzy rules indicating that they may have the same consequent. As mentioned, the GenSoYagerFNN uses the implication-based model and the conclusions of parallel rules are thus combined conjunctively in this layer.

Defuzzification Layer (#5) – Layer 5 contains the output nodes that are responsible for defuzzifying the derived fuzzy conclusions and present them as crisp outputs. Aggregation using a modified centre of averaging (COA) is applied to produce the final output.

5. Automated Parking System

A 3D driving simulator, illustrated in Figure 2, was developed to gather driving data from human drivers \[2\]. This data is used first to train the car controller and subsequently to gauge its driving capability. Feedback information includes sensory data such as distances from obstacles and the car position with respect to the current track/road profile, and control signals consisting of acceleration, brake, and gear ratio.

Reverse parking is a typical manoeuvre that has rapidly become prominent and challenging along with the rise in urban traffic and the limited parking space. In our approach, the parking process is implemented in three distinct stages. First, the GenSoYagerFNN-controlled car will keep following the track until it finds a free parking slot of suitable size. Next, the car will move forward and make adjustments to maintain a constant distance with respect to the wall. This is to alight itself at a favourable position to perform the reverse parking. Lastly, the car will execute the reverse parking manoeuvre proper. The latter makes uses of all three independently trained networks for steering, brake, and acceleration. Adjustments may be required that will move the car backward and forward repeatedly until the correct parking position is achieved. A typical example of successful manoeuvre is illustrated in Figure 3.
A study of the rule firing strength was conducted to analyze the network ability to maintain the consistency of the rule base. Specifically, the proportion of rules actually used was compared to the total number of rules in the network. Minimizing the number of rules in the network is essential to ensure good performance, and becomes critical when using e.g., an embedded control system with limited storage capacity. A consistent rule base is best displayed by a wide spread of the rules being fired over the total number of rules for all possible parking situations. The experiment results for the rule firing strength throughout the reverse parking process are summarized in Figure 4.

For the steering network, 45 rules were produced during the training process. Two prominent rules (rules 27 and 28) and four auxiliary rules with smaller firing strengths are identified. For the acceleration (TPS) network, 112 rules were crafted during training. There are four most significant rules (rules 36, 65, 66 and 112) followed by three auxiliary rules (rules 57, 71 and 107). A large variation in the distribution of rule firing strength in steering and TPS systems was observed, emphasizing the high complexity of both systems. These results are expected since, in the expert reverse parking, the car had to control its velocity and steering frequently within multiple turns so as to avoid colliding with the surrounding barriers.

The brake system presents a smaller variation as compared to the TPS or steering system. Only two essential rules are identified (i.e. rules 4 and 46). This is mainly due to the characteristics of brake in the parking system. Usually, the brake is applied for a relatively short duration and remains inactive for the rest of the time. Similar deduction applies for the parking slot detection system, where there are only three rules (rules 4, 5 and 6) out of 13 rules being frequently fired. In addition, three rules (rules 1, 2 and 3) are pruned away indicating insignificant/weak rules that may contribute to errors during execution.

The structure of the final rule base generated by the training process is summarized in Figure 5. The third row shows the number of input labels (i.e., clusters) per input dimension (feature) resulting from the training.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Steering system</th>
<th>TPS system</th>
<th>Brake system</th>
<th>Detection system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input labels</td>
<td>35</td>
<td>83</td>
<td>25</td>
<td>13</td>
</tr>
<tr>
<td>Output labels</td>
<td>9</td>
<td>23</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Input labels per input dimension</td>
<td>[2, 7, 6, 11, 9]</td>
<td>[33, 30, 20]</td>
<td>[8, 10, 7]</td>
<td>[5, 5, 3]</td>
</tr>
<tr>
<td>Rules</td>
<td>45</td>
<td>112</td>
<td>46</td>
<td>13</td>
</tr>
</tbody>
</table>

Figure 3: Reverse Parking Using GenSoYagerFNN

Figure 4: Rule Firing Strength of the Car Controllers

Figure 5: Structure of the GenSoYagerFNN for the Sample Reverse Parking Manoeuvre
To illustrate the intuitiveness and ease of the interpretation of the fuzzy rules crafted by the GenSoYagerFNN, an analysis of the rule base of the parking detection system is presented here. There are 13 rules crafted in the detection system, which are linked to 13 input labels and 3 output labels. In this system, the inputs to the GenSoYagerFNN consist of three features: the front left distance (Side Front Left Sensor), the middle left distance (Side Middle Left Sensor) and lastly the back left distance (Side Back Left Sensor). The front left distance has 5 fuzzy sets, while the middle left distance and the back left distance have 5 and 3 fuzzy sets respectively. The fuzzy term sets extracted from the parking detection system are as follows:

- **Input “Front left distance”** = {very short, short, medium, long, very long}
- **Input “Middle left distance”** = {very short, short, medium, long, very long}
- **Input “Back left distance”** = {short, medium, long}
- **Output “Detection”** = {off, half-active, on}

As can be seen in Figure 4, the most frequently fired fuzzy rules of for the parking detection system are rules 4, 5, and 6. Considering rule 4, the corresponding fuzzy rule extracted from the rule base of the detection system can thus be formulated as:

\[
\text{if } \text{Front left distance is medium and Middle left distance is medium and Back left distance is medium then Detection is on}
\]

The above conclusion implies that a suitable parking slot is detected and therefore the car is stopped and parking adjustment is performed. The extracted fuzzy rules are intuitive for the human cognitive process and, as demonstrated by the results, the most frequently fired rule 4 fully agrees with human knowledge of parking slot detection in the reverse parking process.

The performance of the GenSoYagerFNN on the reverse parking manoeuvre was also investigated from the point of view of parking quality, as determined by the final car position in the parking slot. The parking slot area is designed such that a parked vehicle is ideally centred about both the width and length of the area. The eight distances considered (f1, f2, l1, l2, r1, r2, b1 and b2) are illustrated in Figure 6, where L is the length and W is the width of the parking area. Perfect quality is achieved when the car is at the centre, hence, when the parameter values are such that \( l_1 = l_2 = r_1 = r_2 \) and similarly \( f_1 = f_2 = b_1 = b_2 \). The optimal values for a car of length \( L_c \) and width \( W_c \) are given in Equation 4.

\[
l_{\text{desired}} = \frac{W - W_c}{2} \quad \text{and} \quad b_{\text{desired}} = \frac{L - L_c}{2}
\]

Once the car is within the parking area, at least three distance values are required to correctly identify its position and orientation e.g., two distance values from the same side to identify its orientation plus the offset from that side. With another distance value from another side neighbouring with previous side, the exact positioning information can be obtained. In this experiment, only three sensors: l2, b1 and b2 are used and the quality of the parking can be expressed in the form of standard deviation as per Equation 5.

\[
\text{Error} = \sqrt{\frac{(l_2 - l_{\text{desired}})^2 + (b_1 - b_{\text{desired}})^2 + (b_2 - b_{\text{desired}})^2}{3}}
\]

\[
\text{Quality} = 1 - \text{Error}
\]

In the experiment, a parking area of length 40 and width 25 was placed beside a road of width 37 as shown in Figure 7. The car is of size 3x1.5 meters.

![Figure 7: Dimensions of the Car and Parking Slot Area](image)

<table>
<thead>
<tr>
<th>Schema</th>
<th>Initial Pos. [x, y, z] and Orientation</th>
<th>Error</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>l1</td>
<td>[2, 0, 84], 0</td>
<td>0.168</td>
<td>0.832</td>
</tr>
<tr>
<td>l2</td>
<td>[-3, 0, 84], -10</td>
<td>0.357</td>
<td>0.643</td>
</tr>
<tr>
<td>l1</td>
<td>[0, 0, 56], 281</td>
<td>0.277</td>
<td>0.732</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>0.267</td>
<td>0.733</td>
</tr>
</tbody>
</table>

Figure 8: Quality Measurement in GenSoYagerFNN
(stage 1) or straight adjustment (stage 2) involved. The experiment concentrates only on the actual reverse parking manoeuvre (stage 3). Two networks were used to control the car, namely the GenSoFNN-CRI(S) and the GenSoFNN-Yager(S) networks. The results of the experiment are summarized in Figure 8.

![Figure 9: Experimental Setup with the Microprocessor Controlled Model Car and its Parking Slot](image)

6. Conclusion

This paper presented the general structure, working principles, and training cycle of the GenSoYagerFNN. Training consists of the self-organisation (clustering), rule formulation and parameter learning (using back-propagation) phases and takes place in a single pass of the training data. This allows the GenSoYager network to perform on-line learning and is suitable for even (some) real-time applications.

The Yager inference scheme employed by the GenSoYagerFNN adopts the implication-based model of fuzzy relation. It computes the final output based on the level of disagreement instead of agreement. Intuitively, this is closer to human reasoning since when the input matches exactly with the antecedent the resultant output matches the consequent exactly as well.

Integration of the GenSoYagerFNN with a car-driving simulator has been performed to demonstrate its capability as a reliable, self-trained control system, resulting in the realization of an intelligent automated driving and parking system. Experiments on the left-in reverse parking manoeuvre have shown that the GenSoYagerFNN performed efficiently in controlling the vehicle to its objective without the aid of human driver. Current experiments demonstrate the applicability of the approach to a microprocessor-controlled model car, shown in Figure 9.

7. References


