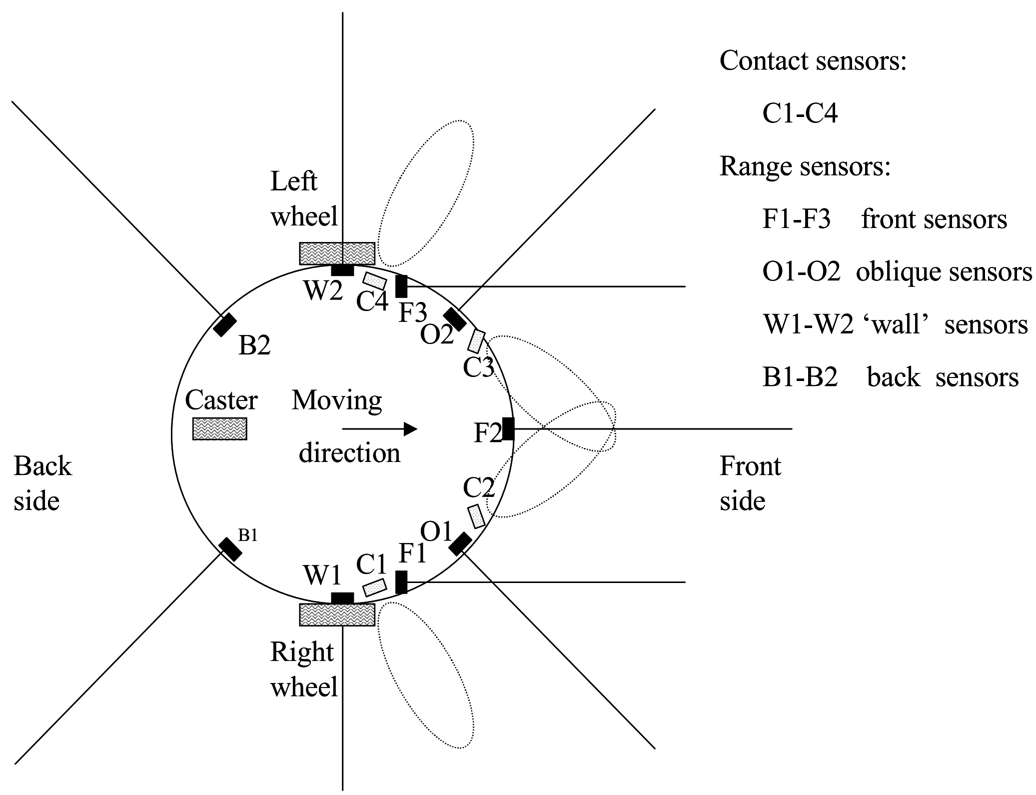
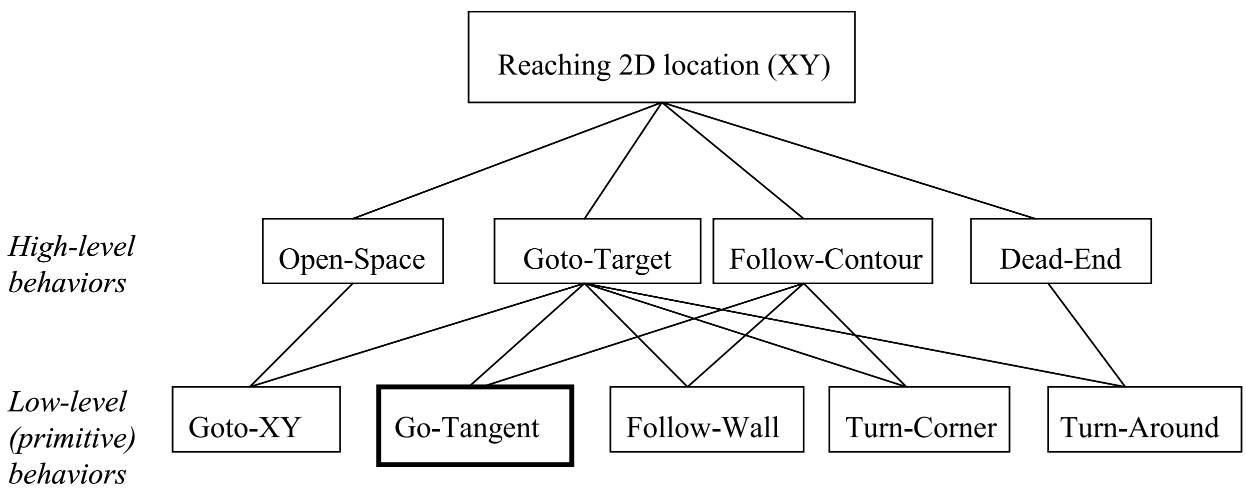
Behavior-Based Neuro-Fuzzy Controller for Mobile Robot Navigation

Petru Rusu [1], Emil M. Petriu [1],
Thom E. Whalen [2], Aurel Cornell [3],
and Hans J.W. Spoelder [4]

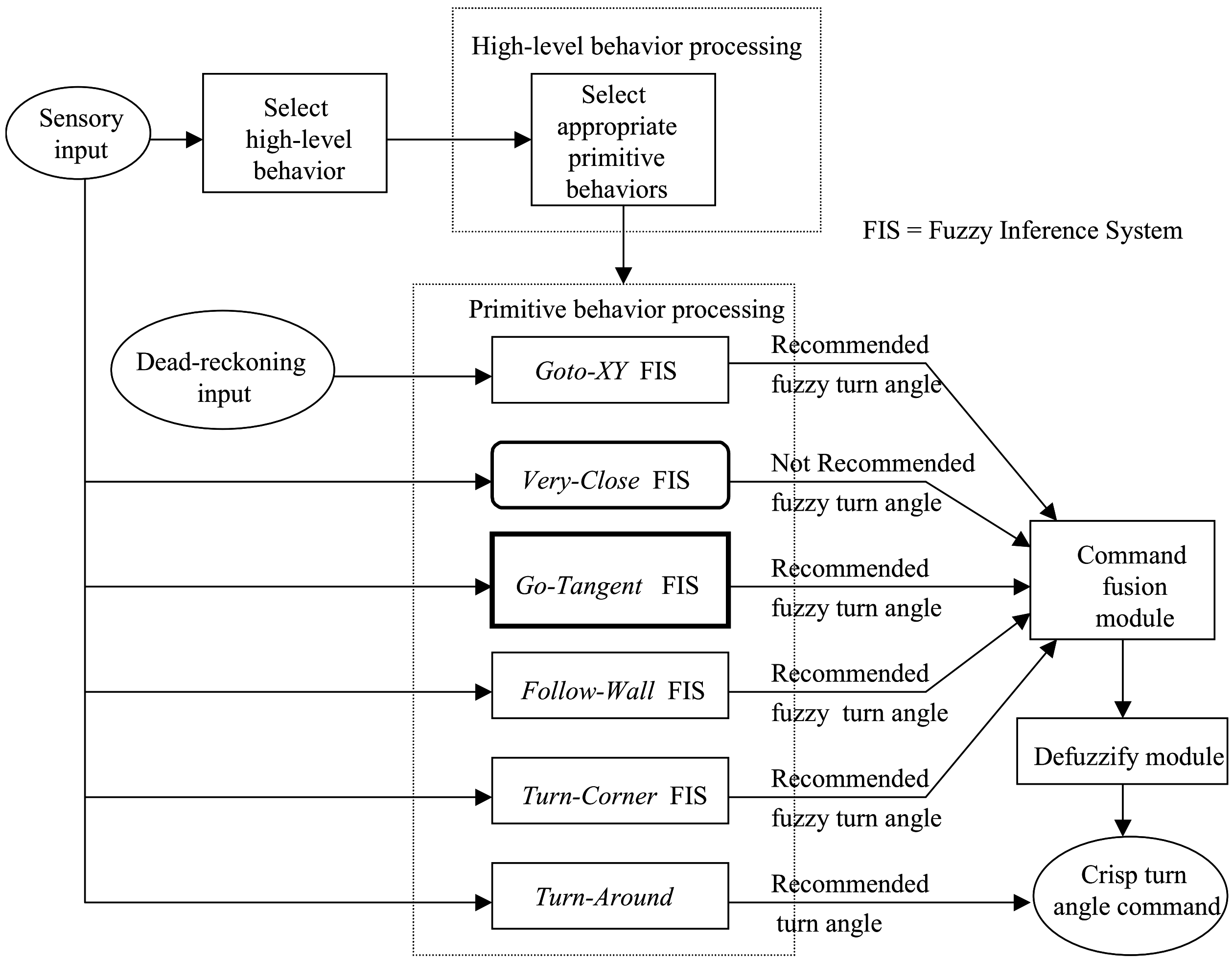
[1] University of Ottawa, Canada
[2] Communications Research Centre Canada, Ottawa, ON, Canada
[3] Brigham Young University, Provo, UT, USA
[4] Vrije Universiteit, Amsterdam, The Netherlands



The sensors of the mobile robot

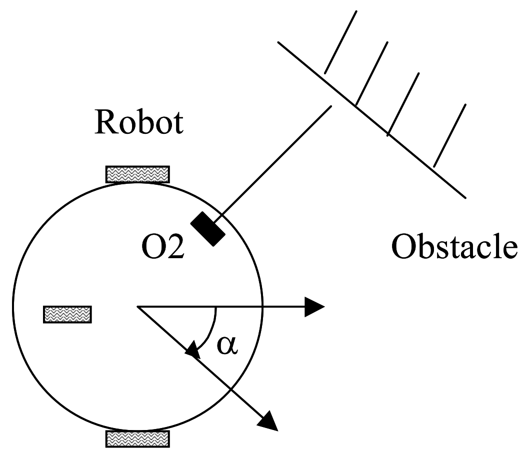


The hierarchy of the sensor-based robot behaviors

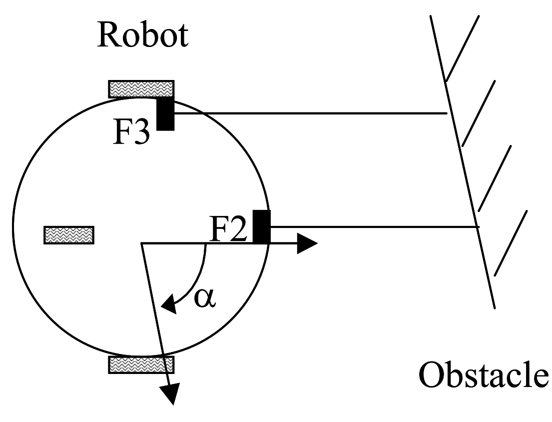


The architecture of the behavior-based controller

NEURO-FUZZY IMPLEMENTATION OF *GO-TANGENT* BEHAVIOR



Sensors used in *Go-Tangent-Oblique-Sensor* case.



Sensors used in *Go-Tangent-Front-Sensor* case.

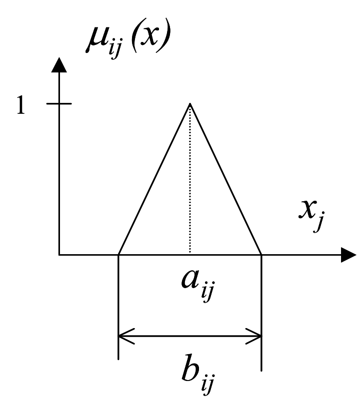
The fuzzy rule for a 2-input and 1-output first-order Sugeno FLC:

IF x is A_i *AND* y is B_j *THEN* $F_k = p_k x + q_k y + r_k$
for i = 1, ..,L; j = 1, …,M; k = 1, …,N; N = L x M

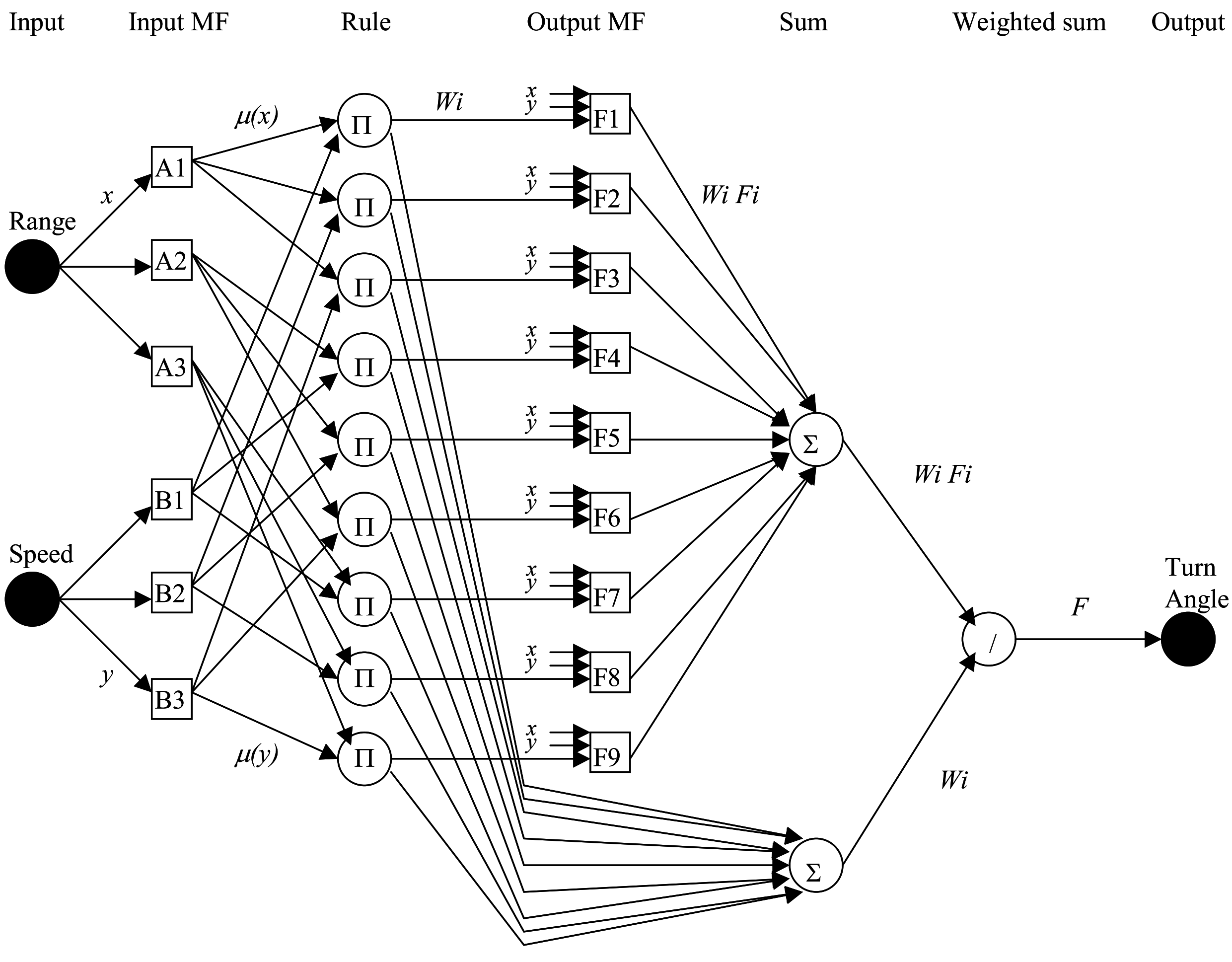
where ***x*** and ***y*** are the linguistic variables, F_k is the output for the *k*-th rule, *L* is the size of the fuzzy set *A*, *M* is the size of the fuzzy set *B*, and *N* is the size of the rule base.

For *Go-Tangent-Oblique-Sensor* ANFIS, ***x*** is the distance to the obstacle with the fuzzy set *A* = {*short, medium, long*}, **y** is the speed towards the obstacle with the fuzzy set B = {*slow, medium, fast*}. Output F_k is the turn angle for the *k*-th rule, and N = 9.
For *Go-Tangent-Front-Sensor* ANFIS both ***x*** and ***y*** are range values. The fuzzy sets for *x is A* = {*short, medium, long*}, and for *y is B* = {*short, medium, long*}. Ouput F_k is the turn angle for the *k*-th rule, and N = 9.

=> We have experimented with several types of membership functions for the fuzzy sets A and B and with various sizes for *L* and *M*. The *triangular membership functions* and a size 3 for each of the two fuzzy sets, L= M= 3, were found to be the simplest and best suited for this case.



$$\mu_{ij}(x_j) = \begin{cases} 1 - \dfrac{2|x_j - a_{ij}|}{b_{ij}}, & \textit{for} \quad a_{ij} - \dfrac{b_{ij}}{2} < x_j \le a_{ij} + \dfrac{b_{ij}}{2} \\ 0 \quad , & \textit{otherwise} \end{cases}$$



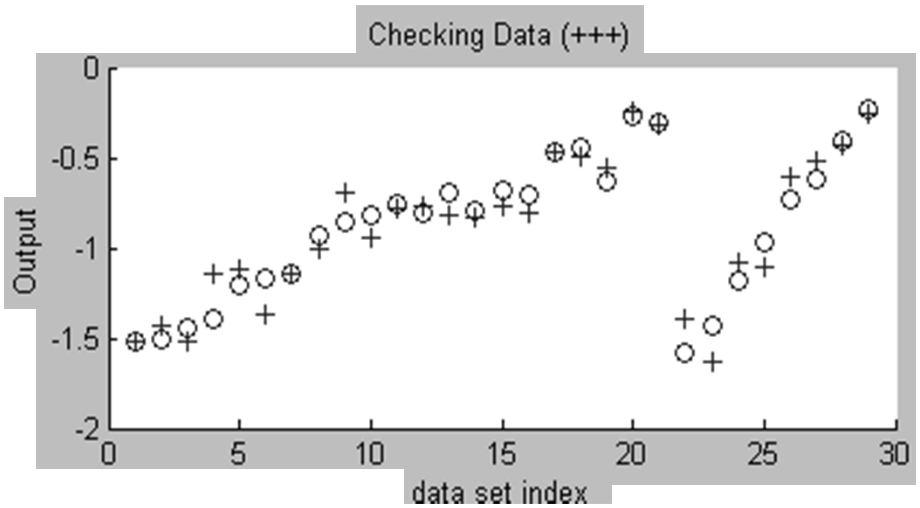
Neural Network identification of the Sugeno ANFIS parameters $\{a_{ij}, b_{ij}\}$ and $\{ p_k , q_k, r_k \}$

The *square* elements represent the *adaptive nodes* depending on the parameter set of the adaptive network. The *circles* represent *fixed nodes*, which are independent of the parameter set.

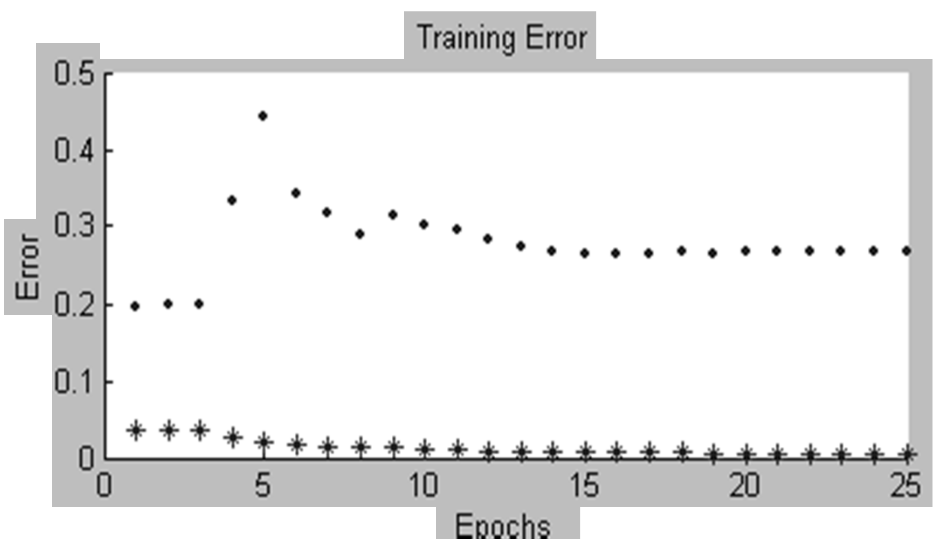
The *first layer* is composed of adaptive nodes representing the membership functions associated with each linguistic value.
The *second layer* implements the fuzzy rules. It includes only fixed nodes implementing a product Π between the membership degrees of the two inputs, μ (x) and μ (y), corresponding to the two propositions in the antecedent of each fuzzy rule.
The *third layer* consists of adaptive nodes, which include the output membership function.
The *other two layers* consist of fixed nodes that implement the weighted average for the output *F* representing the turn angle.

$$F = \sum_{k=1}^{N} (\overline{W_k} \bullet x) \bullet p_k + \sum_{k=1}^{N} (\overline{W_k} \bullet y) \bullet q_k + \sum_{k=1}^{N} \overline{W_k} \bullet r_k$$

As the size of the rule base of the Sugeno FIS is N=9, we will have to identify 27 consequent parameters $\{p_1,\ldots,p_9,\ q_1,\ldots,q_9,\ r_1,\ldots,r_9\}$. This will be done by a backpropagation NN using a training set {x,y,F} of size P.
In the *forward pass*, the input membership functions are fixed and the consequent parameters associated with the output are calculated by applying the least square estimation. Using these parameters, the NN estimates the turn angle. The difference between this estimate and the turn angle's value from the training set is then *backpropagated* in a *second pass* when the premise parameters associated with the input membership functions are calculated.



The 29 points (range, speed, turn angle)data set used for training Go-Tangent-Oblique-Sensor ANFIS. A checking data set was obtained by applying a 10% random noise to the training data set.

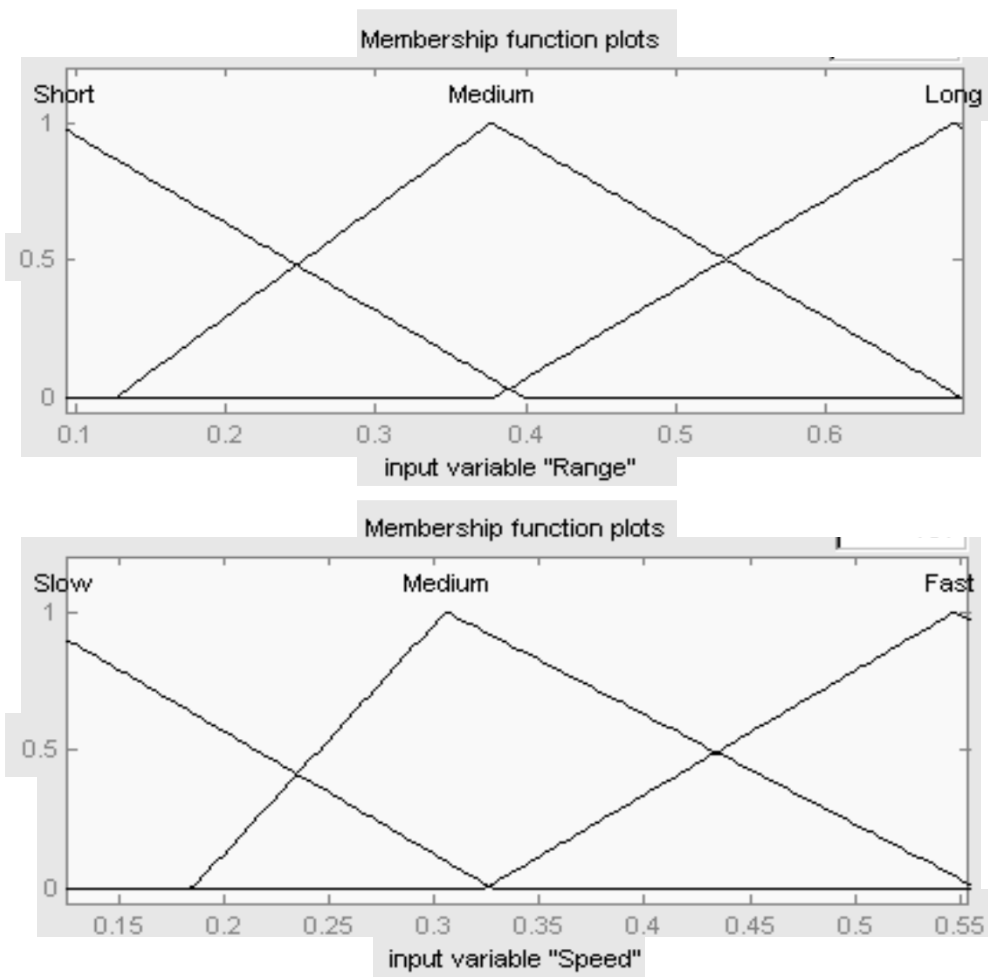


The root mean squared errors of the output over 25 training epochs .

One result of the training is the generation of the set of premise parameters $\{(a_{ij}, b_{ij}) | i=1,2,3; j=1,2,3\}$ of the membership functions.

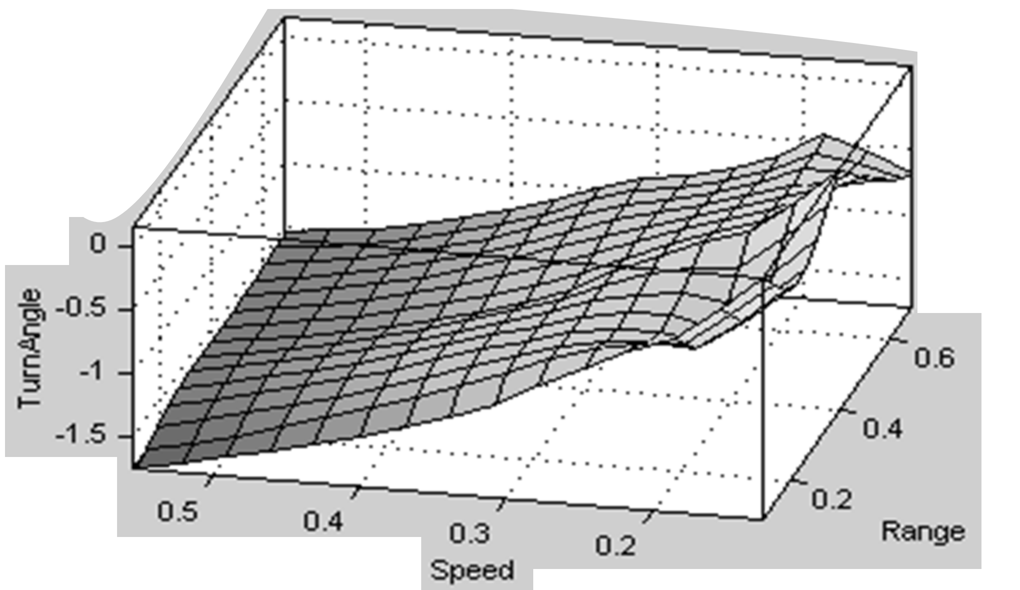
The following *figure* shows the resulting membership functions for the *range* input and respectively the *speed* input.

It may be interesting to note that, although the training started with symmetric membership functions, their shape eventually became asymmetric after training.

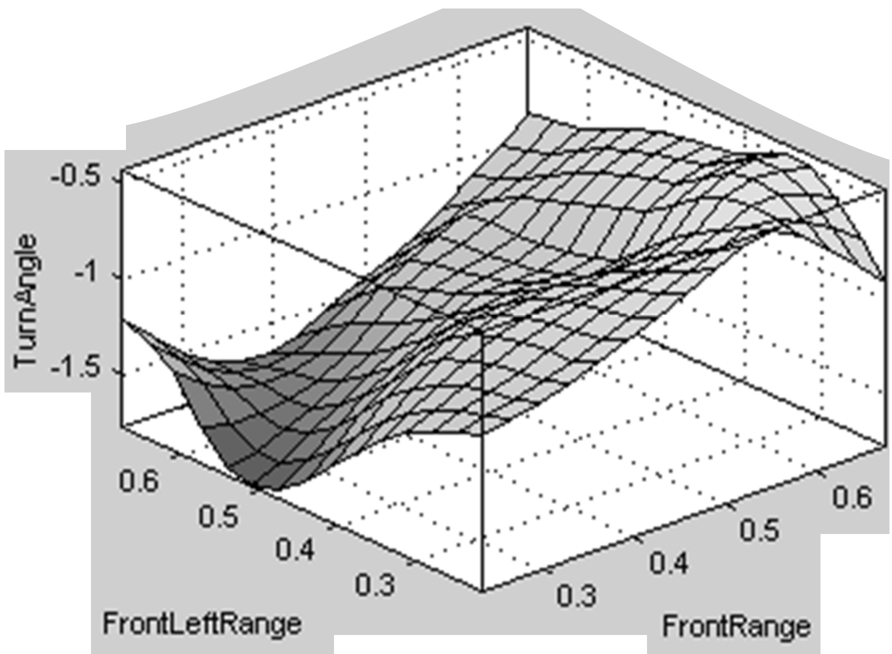


The second result of the training is the generation of the set of consequent parameters $\{(p_k, q_k, r_k) | k=1,2,,9\}$ in the fuzzy rules. With these parameters the Sugeno fuzzy rules are:

If (*Range* is Short and *Speed* is Slow)
then (*TurnAngle* = -3.253 x - 8.858 y + 1.561)
If (*Range* is Short and *Speed* is Medium)
then (*TurnAngle* = 4.215 x - 8.077 y + 1.133)
If (*Range* is Short and *Speed* is Fast)
then (*TurnAngle* = 0.006 x - 6.715 y + 2.008)
If (*Range* is Medium and *Speed* is Slow)
then (*TurnAngle* = 2.967 x - 9.659 y + 3.276)
If (*Range* is Medium and *Speed* is Medium)
then (*TurnAngle* = 2.241 x - 8.763 y + 0.933)
If (*Range* is Medium and *Speed* is Fast)
then (*TurnAngle* = -1.113 x - 7.021 y + 2.762)
If (*Range* is Long and *Speed* is Slow)
then (*TurnAngle* = 3.907 x + 4.525 y - 3.954)
If (Range is Long and S*peed* is Slow)
then (*TurnAngle* = 1.249 x + 3.560 y - 2.943)
If (*Range* is Long and *Speed* is Slow)
then (*TurnAngle* = -0.951 x +4.639 y - 3.477)

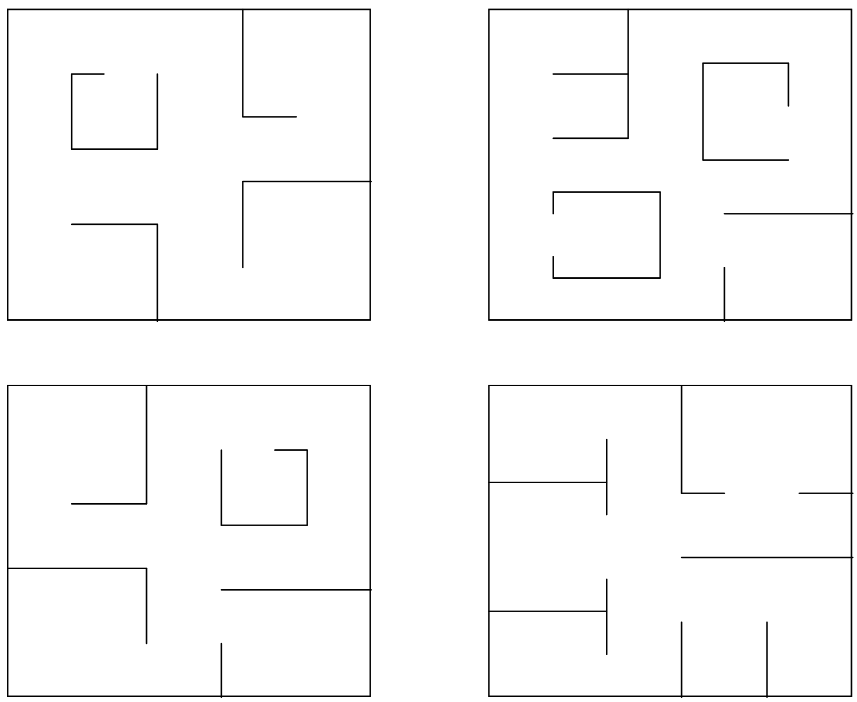


The input-output characteristics of Go-Tangent-Oblique-Sensor ANFIS.



The input-output characteristics of Go-Tangent-Front-Sensor ANFIS.

The behavior-based robot controller has been *tested in four simulated maze-like indoor environments*, for over 2,000 endpoints.



We used the *Rossum's Playhouse simulation environment* , [9]. We developed our own client application for the behavior-based neuro-fuzzy controller. For a given set of endpoints, the simulation environment monitors the time needed for each target search.

The speed of the robot was set at 0.4 m/s. The four test environments varied in size from 10x10 m to 13x13 m. Traveling between two opposite corners of any environment following a path parallel with the walls, has required 60 seconds. None of the target searches required more than 200 seconds.