

A neuro-fuzzy controller for collaborative applications in robotics using LabVIEW

Hiram E. Ponce, Instituto Tecnológico y de Estudios Superiores de Monterrey, Mexico

Dejanira Araiza, Instituto Tecnológico y de Estudios Superiores de Monterrey, Mexico

Pedro Ponce, Instituto Tecnológico y de Estudios Superiores de Monterrey, Mexico

Article originally published in "Applied Computational Intelligence and Soft Computing", V.I(2009), Copyright © 2009 Hiram E. Ponce et al. Distributed under the Creative Commons Attribution License. <http://www.hindawi.com/journals/acisc/2009/657095.html>

ABSTRACT. A neuro-fuzzy controller was designed and implemented using LabVIEW over a mobile robotic platform. The controller is based on fuzzy clusters, neural networks, and search techniques. Also, wireless communication with Bluetooth protocol was used to communicate the robot with the controller running in LabVIEW, allowing a simple collaborative task that consisted in pick and place objects, through knowing the position of the robot and measuring the distance to the objects. The neuro-fuzzy controller was split in two parts: the position controller and the evasion controller against collisions.

KEYWORDS: Artificial neural network, Fuzzy cluster means, LabVIEW, Mobile robotic platform, Neuro-fuzzy controller

Many problems that affect society can be solved using software and hardware that includes powerful tools from artificial intelligence. For example, determining the amount of money necessary in a specific bank according to the demand, having a database to know the most common sicknesses in Mexican children to have enough medicine and institutions to provide health care assistance, making decisions in small businesses, all this means the possibility to use technological resources to reach development and improvement of life's quality among people in different countries (García Fernández, 2004).

For all the mentioned above, the generation of generic and global tools to develop algorithms and programs is a challenging area that deserves the attention of the academic community, as more trained professionals are needed in order to develop and use the tools to improve situations and to solve problems. This project is focused in using modern tools like fuzzy control, neural networks, computational search, wireless communication, and others, to

generate material and ideas for implementing solutions. The results of it can be extrapolated beyond the task that was selected to develop the collaborative work, but can be done in any other field as examples described in Ranjan et al. (2006) and Bhogle et al. (2007).

The main objective of this project includes implementing a collaborative task in a group of microrobots, implicating autonomous navigation in a determined area, through the use and development of algorithms. In order to achieve this, the work was divided in three goals: designing and implementing a neuro-fuzzy controller based on previous works (Cruz et al., 2006; Méndez et al., 2007) to get the autonomous navigation inside a determined area and implementing the collaborative work.

Additionally, some specific objectives were stated: using LabVIEW (NI, 2007) to design and implement the algorithms to evaluate their use, designing a collaborative task for the micro robots used in the project, and generating a new structural design, cheap and flexible, for future projects.

The first steps in the project implicated research about the theory that sustains this work, mainly about fuzzy controllers, neural networks, fuzzy clusters, and search techniques. Subsequently, the efforts were focused in designing the neuro-fuzzy controller (Rattan, Sandhu, 1999), including a model of the system to validate the controller, and making the proper adjustments to the physical structure of the robots. Finally, implementation of intercommunication was achieved using Bluetooth technology, and the collaborative task was designed, followed by several tests and measurements to validate the work.

Prototype description

Robots used in this project contemplate a 15x15 cm design, with a plastic base, two differential modified servomotors, a simple wheel for support, and different boards including a microprocessor Basic Stamp 2, three Ping ultrasonic range sensors for navigation place them one in front and the other two at left and right sides, and an Embedded Bluetooth Transceiver for wireless communication, on each robot. This system also includes a magnet for the collaborative task.

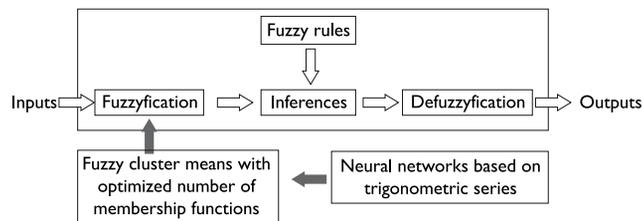
In order to control the avoidance of obstacles and the movement of the robot, a neuro-fuzzy controller was implemented to run

over the LabVIEW platform (NI, 2007), in communication with the robots through Bluetooth protocol, using a Belkin USB-Bluetooth adapter.

Neuro-fuzzy controller

A neuro-fuzzy controller is used in robots in order to obtain the desired movements on them, that is, reaching a final position getting from an initial position. Figure 1 is a block diagram of the neuro-fuzzy controller proposed. It is based on trigonometric series (Cruz et al. 2006) and an interface to communicate the robot with the processing system. In fact, a collaborative work has to be in a real-time-based processing. Then, trigonometric artificial neural networks (Cruz et al. 2006) are implemented because their training phase only requires one epoch. In addition, this controller allows obstacles avoidance and adaptation of parameters according to interactions between the robot and its environment.

Figure 1. Proposed neuro-fuzzy controller (Cruz et al., 2006)



The robot is a MIMO system in which we have to control the rotation of the two wheels via two digital pulses coming from the microcontroller. So, it can be divided in two neuro-fuzzy controllers to simplify the designing of the fuzzy controller part: one that avoids collision against obstacles, and one to control the position through the motion in the servos, considering a position as the set point.

Neural networks

Based on the definition of a fuzzy controller and its parts, and referring to Sugeno approach in the inference mechanism (Nguyen et al., 2002), neural networks can be applied to the clusters obtained through the Fuzzy Cluster Means (FCM) method. Fuzzy inferences of the form *if_then*, and a neural network in the output, can be expressed like

$$R_i : \text{if } x_1 \text{ is } B_{i1}, \dots, x_n \text{ is } B_{in} \text{ then } y_i = g_i(*), \quad (1)$$

Where $y_i=g_i$ can be a group of trigonometric functions (Cruz et al. 2006), considering that Fourier series can be used to model the input in a neural network, and this method is useful to obtain the coefficients of the weights (Cruz, Suárez, 2005).

Considering a Fourier series defined with (2) and similarities which describes neural networks as (3) it makes evident the possibility of its description, where, coefficients of (2) are the Fourier Coefficients:

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} (a_n \cos(nwx) + b_n \sin(nwx)), w = \frac{\pi}{l}, \quad (2)$$

$$Y = f\left(\sum_{i=0}^n X_i w_i\right) \quad (3)$$

The topology includes two layers of neurons, the first composed by neurons with a trigonometric activation function and a weight that depends directly on the frequency. The second layer adds the trigonometric functions, multiplied by their respective weights plus a constant, as Figure 2 shows (Cruz, Suárez, 2005). The coefficients or weights in the neural network can be calculated considering if the signal is even or odd. A minimum squares method is used to find the coefficients analytically, through (4) converted in a matrix (Cruz, Suárez, 2005):

$$\sum_{i=1}^m w_i g(b_0, b_1, \dots, b_p; x_i) \sin(jwx) = \sum_{i=1}^m w_i y_i \sin(jwx) \quad \text{for } j \neq 0. \quad (4)$$

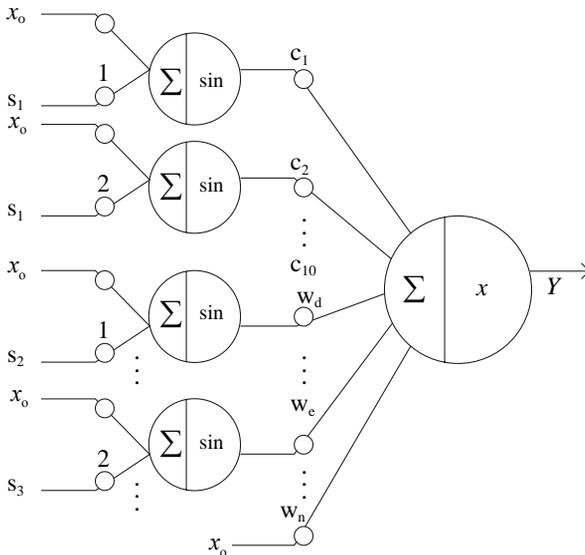


Figure 2. Topology of a trigonometric artificial neural network (Cruz, Suarez, 2005)

Fuzzy cluster means

In the inputs, it has been proved that better results are obtained using the method called Fuzzy Cluster Means to adapt and tune membership functions according to the environment surrounding a robot. This method allows the minimization of the distance between the elements of each cluster and the maximization of the distances that separate the centroids in the clusters. This can be defined by the objective function in (5).

Neural networks based on trigonometric series can be applied after obtaining the clusters, to soften the shapes of the membership functions, and to approach them to trigonometric series (Cruz et al., 2006):

$$J = \frac{1}{2} \sum_{x=1}^N \sum_{i=1}^C \mu_{x,i}^m d^2(z_x, v_i) \quad (5)$$

where $\mu_{x,i}$ is the membership value of the element $x=\{1, 2, \dots, N\}$ in the fuzzy cluster $i=\{1, 2, \dots, c\}$; v_i is the centroid in each cluster; z_x , with $x=\{1, 2, \dots, N\}$, is the group of data; m is the fuzzyfication value; $d^2(z_x, v_i)$ is the Euclidian distance between z_x and v_i ; N is the number of samples in data; c is the number of clusters (Bezdek, Pal, 1992).

Tabu search method

The Tabu search method is a heuristic way to find a best solution combining several strategies: descendent method, long and short terms memory, and diversification strategies. As the search is happening, some of all the possible solutions are named as “taboo” and included in a list through the memories, during a defined number of iterations. These solutions will not be visited in some points of the search; meanwhile better solutions are found to replace an initial proposed solution (Glover, Laguna, 1997).

An algorithm to find a better set of fuzzy rules than the proposed ones was developed and programmed, and it is included in Figure 3.

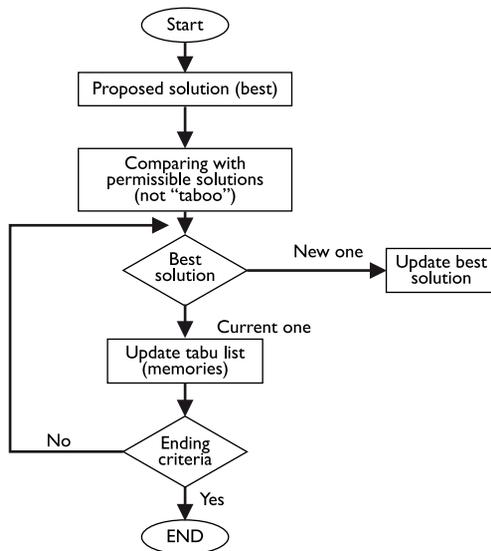


Figure 3. Tabu search algorithm

Design of each controller

The procedure of designing is described in Figures 4 and 5, for both the position controller and the evasion one.

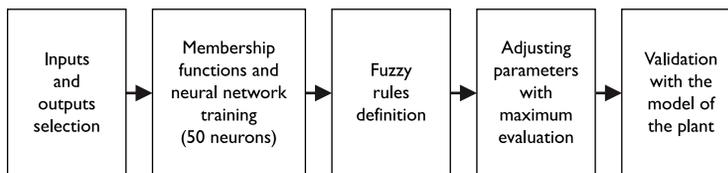


Figure 4. Design of the position controller

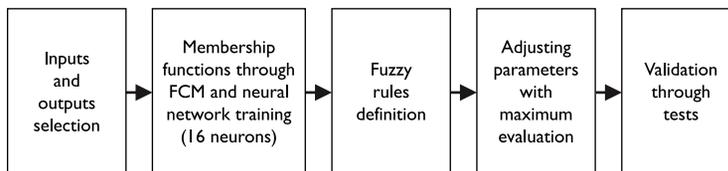


Figure 5. Design of the evasion controller

The final neuro-fuzzy controller can be explained as following.

- (I) Two neuro-fuzzy controllers run parallel: one for avoiding obstacles and one for correcting the position of robots.
- (II) Each neuro-fuzzy controller has the structure of Figure 1, in which inputs are fuzzified by memberships obtained from a fuzzy clustering determined by the environment

and trigonometric neural networks. Then, a set of optimal rules obtained by the Tabu search method are evaluated in order to obtain a desired fuzzy output. Finally, outputs are computed by trigonometric neural networks. As seen, membership functions and rules are changing depending on the environment, given to robots the characteristic of adaptation.

- (III) Avoidance obstacles neuro-fuzzy controller uses the left, right, and center measured distances between the robot and the nearest obstacle. Outputs are the pulses for correcting the position of the robot generated by some obstacle.
- (IV) Position neuro-fuzzy controller uses the final position as inputs and outputs are the pulses for correcting the position of the robot.
- (V) Avoidance obstacle controller has the highest priority.

Collaborative task

The general idea of the collaborative task selected implies the detection and collection of small objects through the use of basic sensors-ultrasonic and infrared sensors. After localizing and collecting the objects, the robots move to deposit them in a certain place, using intercommunication between the computer and the robots to control the movements.

Certain physical parameters have been stated for the collaborative task: a plane surface with 2x2 m per side, 15 cm tall walls around the surface, five spherical objects with metallic incrustations, and half illuminated space. Other specific parameters are 10 cm minimum of separation between each object, with a predetermined distribution for the objects and also for the robots. This is illustrated in Figure 6.

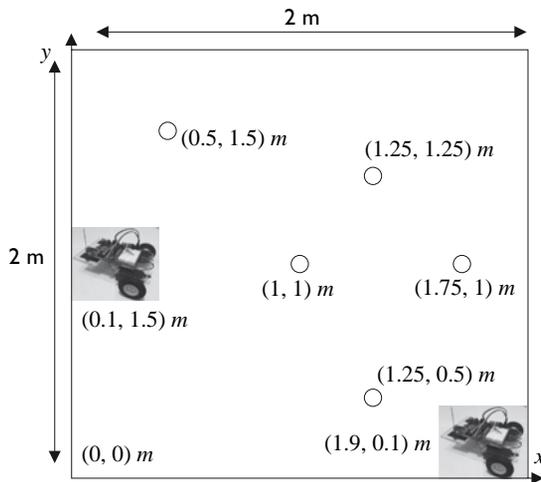


Figure 6. Fixed parameters for the collaborative task

The proposed algorithm for the scenario is shown in a schematic way in Figure 7. The initial position of each robot is called *home*, and it is the place where the objects will be deposited after the recollection. In particular, the algorithm was proved with two robots. Actually, scaling the number of robots is also possible with the implementation of this neuro-fuzzy controller.

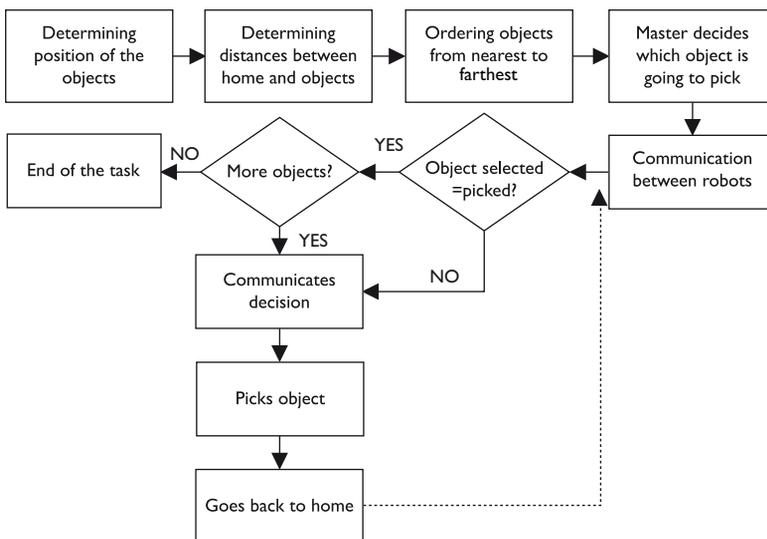


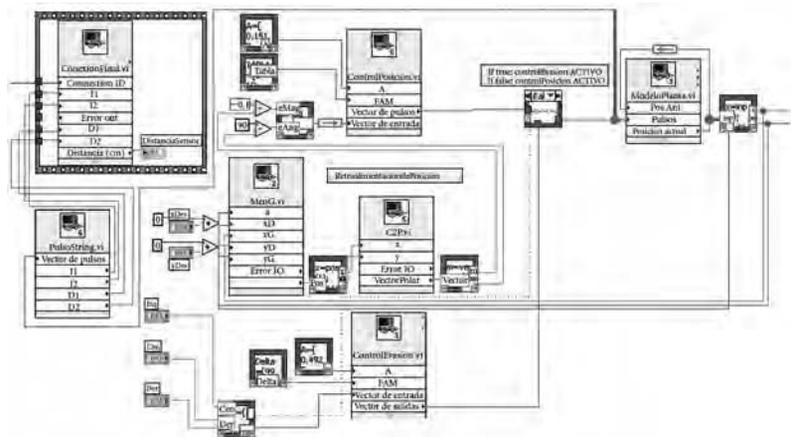
Figure 7. Proposed algorithm for the simple collaborative task

Results

Neuro-fuzzy controller

The main result is the generic neuro-fuzzy controller. It was designed based on a Sugeno approach for generic fuzzy controllers, and it was programmed on LabVIEW (NI, 2007) as the frame to create a generic method that allows to create any kind of neuro-fuzzy controller only changing basic parameters like the number of membership functions. This generic controller allows the scalability on the number of robots in the collaborative task. The total controller implemented on LabVIEW is shown in Figure 8.

Figure 8. Controller developed in LabVIEW (Block diagram)



Position controller

With all programs ready in LabVIEW (NI, 2007), several tests were made to verify the position controller error in different trajectories and also to prove the right behavior of the evasion controller using the most common obstacle cases in the proposed scenario. For the position controller, a linear desired position results in the trajectory shown in Figure 9, with an error of 5.46%. On the other hand, a diagonal desired position has the trajectory of Figure 10 and an error of 6.2%.

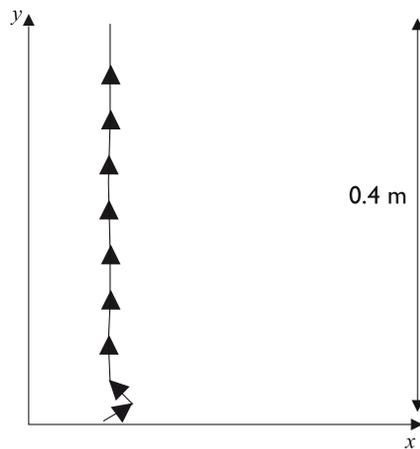


Figure 9. Trajectory for a linear desired position

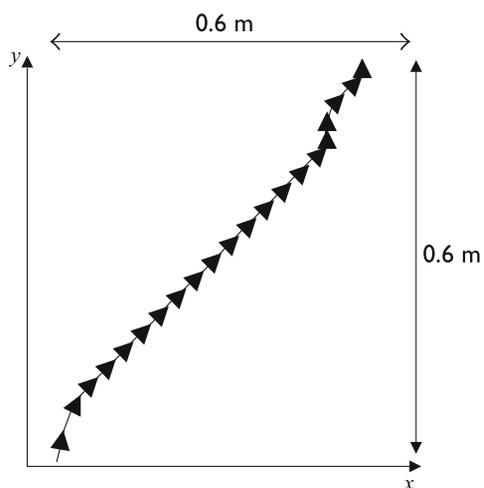


Figure 10. Trajectory for a diagonal desired position

The set of fuzzy rules proposed was verified using Tabu search method, with this set of rules and other selected randomly, with the results shown in Tables 1 and 2 for both cases, respectively.

Table I. Tabu method results for the proposed FAM

Initial		Tabu result	
Left pulse	Right pulse	Left pulse	Right pulse
0	10	0	10
0	5	0	5
10	10	10	10
5	0	5	0
10	0	10	0
0	10	0	10
0	5	0	5
5	5	5	5
5	0	5	0
10	0	10	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	10	0	10
0	5	0	5
-5	-5	-5	-5
5	0	0	10
10	0	10	0
0	10	0	10
0	5	0	5
-10	-10	-10	-10
5	0	5	0
10	0	10	0

Initial		Tabu result	
Left pulse	Right pulse	Left pulse	Right pulse
0	0	0	0
0	0	0	0
0	0	0	0
0	5	0	5
0	5	0	5
10	0	10	0
10	10	10	10
5	0	5	0
5	0	5	0
10	10	10	10
0	10	5	0
0	5	0	5
5	0	5	0
10	0	0	0
10	0	0	0
5	5	5	5
5	0	5	0
0	5	0	5
10	0	10	0
10	0	10	0
10	10	10	10
0	10	0	10
0	10	0	10
5	0	5	0
5	0	5	0

Table 2. Tabu method results for the random FAM

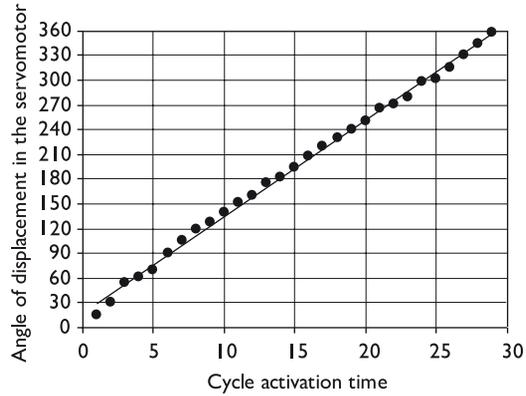
Mathematical model of the plant

In order to finish the design and adjustment of the position control, a mathematical model from the navigation behavior was needed. For this, the servomotors were characterized first, and then a model that predicts the position according to the real motion of the servos was developed.

This model describes the relation between the pulses applied to the servos and the displacement considering the direction of the turn. Characterization of servomotors was done. The result of the measures is shown in Figure 11, and this can be expressed as (6)

$$\theta = 11.927p + 9.220 \quad (6)$$

Figure 11. Measurement of angles in the servomotor



Define the input variable $u=[u_1 \ u_2]^T$ as the vector of machine times that are needed to move the wheels, and the output variable $y=[y_1 \ y_2 \ y_3]^T$ as the position vector in the 2D navigation plane, with direction angle extension respecting from a coordinated plane M in \mathbb{R}^2 defined in the initial states.

In general, the action of the plan can be divided in two main blocks: the angular displacement model and the placement in the M plane, as Figure 12 shows.

Figure 12. Block diagram of the model

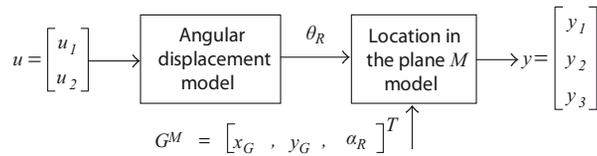
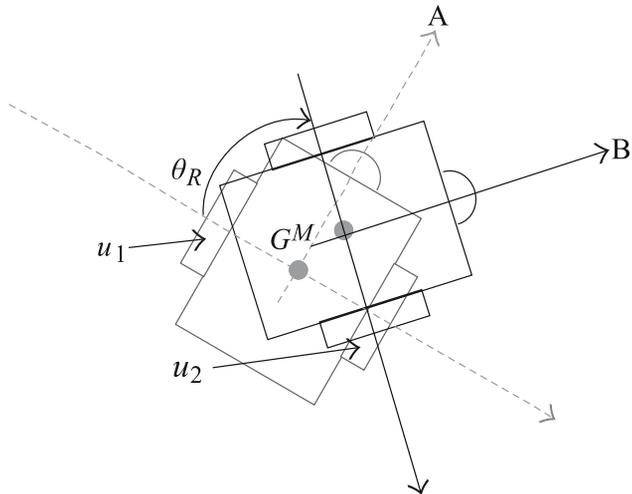


Figure 13. Angular displacement model after applying a group u of pulses to the wheels. Point G^M represents the point of interest in the robot, and plane R is defined from it. (a) Plane R before the pulse input, (b) plane R after the movement produced by the pulses



The angular displacement model considers an input variable $u=[u_1 \ u_2]^T$, which contains the information about the activation time in the rotation of each wheel, u_1 is the activation time in the left wheel, and u_2 represents the activation time in the right wheel. The output variable $y_d=\theta_R$ in this block is the relative angular displacement of plane R in \mathbb{R}^2 , with its center in the origin G^M , mapped referring to the robot itself. This block determines the relative angular displacement according to the previous location and the new position after the movement (Figure 11).

A series of criteria was determined to delimitate the angular displacement model. (a) The robot can move forward or backward and the vector u has its elements with the same value. (b) The robot can turn to the left, and $u_1=0, u_2 \neq 0$. (c) The robot can turn to the right, and $u_1 \neq 0, u_2=0$. (d) The robot can remain static and the vector u has 0 values. (e) A movement forward takes place if the vector is positive. (f) A movement backward takes place if the vector is negative.

The absolute displacement depends on the actual position in plane M ; so a point G^M is defined as the origin of Euclidian plane R (Figure 14). Point G^M has the direction value measured in plane M . Mathematically, G^M is a position vector (7):

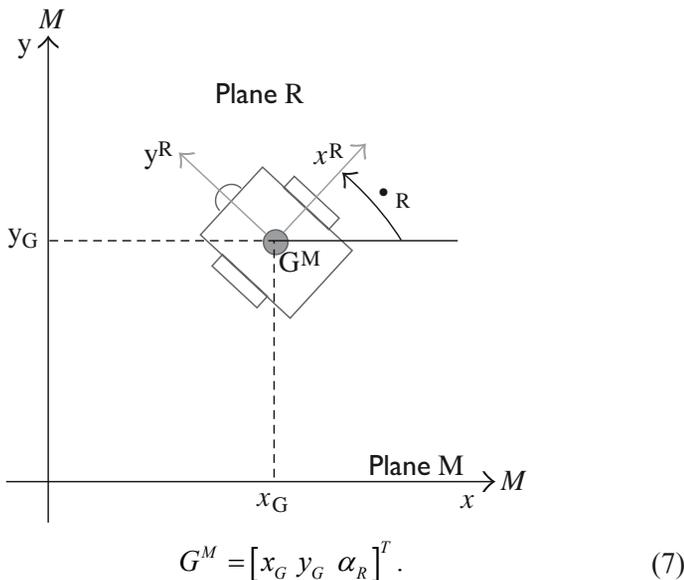


Figure 14. Relation between plane M and plane R , both in \mathbb{R}^2

The transformation between plane R and plane M is done using

homogeneous matrices (Efimov, 1970; DATC, 2009), where the mapping $\varphi: R \rightarrow M$ starts with (8):

$$\varphi' = T\beta', \quad (8)$$

where

$$T = \begin{bmatrix} \cos \alpha_R & -\sin \alpha_R & x_G \\ \sin \alpha_R & \cos \alpha_R & y_G \\ 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

$$\beta' = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The coordinates in φ correspond to the new position in plane M . The new angle of direction is (10):

$$\alpha_R^{k+1} = \begin{cases} \alpha_R^k, & u_1 = u_2, \\ \alpha_R^k + \theta_R, & u_1 = 0, \quad u_2 \neq 0, \\ \alpha_R^k - \theta_R, & u_1 \neq 0, \quad u_2 = 0. \end{cases} \quad (10)$$

The output vector is defined as $y = [y_1 \ y_2 \ y_3]^T$, where the values of its element imply (11):

$$y = \begin{bmatrix} \varphi \\ \alpha_R^{k+1} \end{bmatrix}. \quad (11)$$

Twenty observations were taken to validate the proposed model.

Evasion controller

Four different cases were tested to verify the evasion against obstacles, which are shown in Figure 15 together with the trajectory followed by the robot on each case. The evaluation of this controller is acceptable.

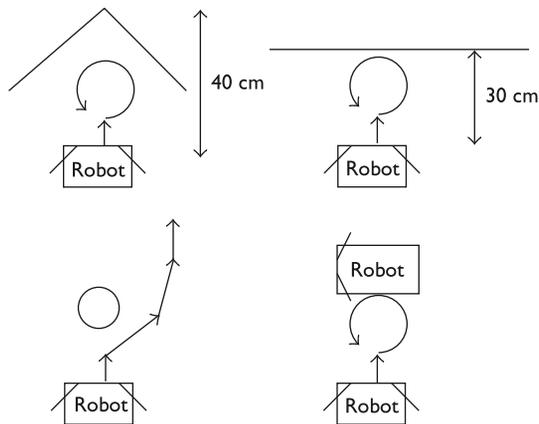


Figure 15. Different cases for the evasion controller evaluation

Wireless communication

In order to send and receive the numerical data from the microcontroller to the computer and back, a basic protocol was designed. This protocol sent characters such as “go” or “hi” as the control signal, and then the respective digits of the correction signal or the feedback.

To establish and implement the communication through Bluetooth, a routine was designed based on predetermined blocks including exploration to find the device, sending, and receiving.

Designing a new robot

Searching for a more flexible and less expensive platform, a new structure was designed and implemented as shown in Figure 16, using plastics and polymers to substitute aluminum and other heavier materials. The saving was of 16% in costs and 10% in weight.

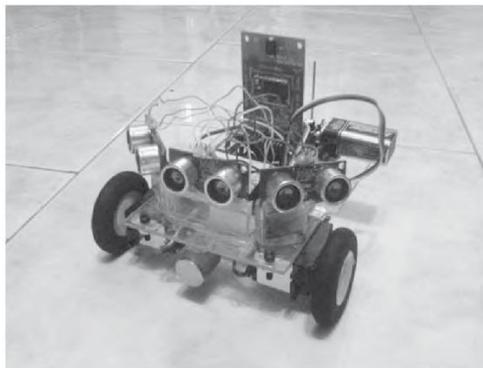


Figure 16. Implementation of the new structural design

Collaborative task

Several experiments were made to observe the performance of the proposed system during the collaborative task, and the results of the trajectories followed by the robots are shown in Figure 17, according to the algorithm proposed previously. The total time was 17 minutes and 20 seconds. According to the results, time might be enhanced by proving other communication protocols and removing the master station. In that way, robots have to have a better microcontroller that allows programming the entire neuro-fuzzy controller in any one. On the other hand, Figure 18 shows how a robot is catching an object with the magneto at the front part in this collaborative task. However, the neuro-fuzzy controller has its main characteristic that can be used in other tasks different from that one, because of its inherent adaptation to the environment.

Figure 17. Trajectories followed by the two robots during the collaborative task

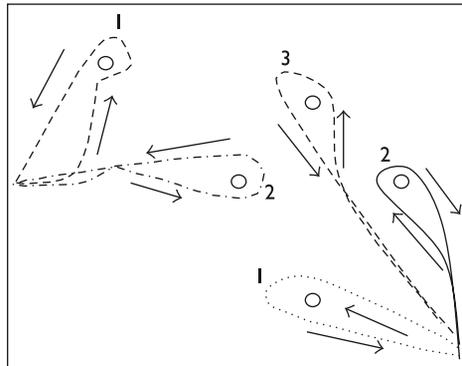
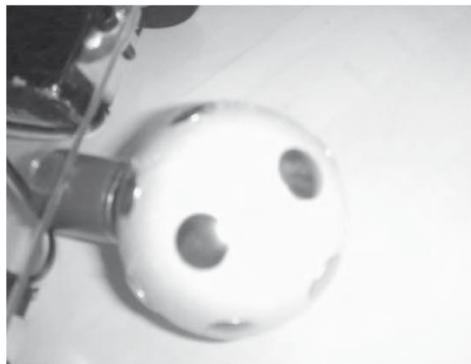


Figure 18. A robot taking an object with the magneto



Conclusions

A neural fuzzy-controller topology was proposed to control the autonomous navigation of a robot, and it was implemented in LabVIEW (NI, 2007) to be tested in a collaborative task. The controller acts as expected, collecting the objects and evading obstacles. The proposed blocks of algorithms allowed adjusting parameters as many times as it was necessary, and the whole methodology gave a possibility to extrapolate the results to other situations and tasks. The final prototype of the generic neuro-fuzzy controller was programming on LabVIEW, and it can be extended to other situations (Ranjan et al., 2006; Bhogle et al., 2007).

References

- Bezdek James, Pal Sankar (1992), *Fuzzy Models for Pattern Recognition*, Institute of Electrical & Electronics Enginee, New York, NY, USA
- BhoglePrashantPurushottam.PatreBalasahebMaharudraappa.WaghmareLaxman Madhavrao, Panchade Vidyadhar Malikarjun (2007), *Neuro fuzzy temperature controller*. Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '07), Harbin, China, 5-9 August 2007, pp. 3.344-3.348
- Cruz Pedro Ponce, Azcue Eduardo, Silva Jose, Silva Juan (2006), *A Novel Neuro-Fuzzy Controller Based on Both Trigonometric Series and Fuzzy Clusters*, Tecnológico de Monterrey, México
- Cruz Pedro Ponce, Suárez R. S. M. (2005), *Neural Networks Based on Fourier Series*, Tecnológico de Monterrey, México
- DATC - Departamento de Arquitectura y Tecnología de Computadoras (2009), *Matrices homogéneas*, Universidad de Sevilla, Spain, [Power Point presentation]
- Efimov N. (1970), *Formas cuadráticas y matrices*, Mir Editions, Moscou, Russia
- García Fernández Luis Alberto (2004), *Usos y aplicaciones de la inteligencia artificial*, "La Ciencia y el Hombre", V. 17, n. 3, p. 1

All URLs checked
June 2010

Glover Fred, Laguna Manuel (1997), *Tabu Search*, Kluwer Academic Publishers, Boston, USA

Méndez David, Ramírez Fernando David (2007), *Sistema de Navegación Neuro-Difuso para Robots Móviles*, Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Ciudad de México

Nguyen Hung, Prasad Nadipuram, Walker Carol, Walker Ebert (2002), *A First Course in Fuzzy and Neural Control*. 1st edition, Chapman & Hall, Boca Raton, USA

NI - National Instruments (2007), *LabVIEW 8.5*, Programming software, United States of America

Ranjan Rahul, Awasthi Anuj, Aggarawal Neelesh, Gulati Jagdeep (2006), *Applications of fuzzy and neuro-fuzzy in biomedical health sciences*, in "IEEE International Conference on Electro Information Technology", East Lansing, Mich, USA, pp. 60-65

Rattan Kuldip S., Sandhu Gurpreet S. (1999), *Design of a proportional plus derivative neuro fuzzy controller*. Proceedings of 18th International Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS '99), New York, USA, 10-12 June 1999, pp. 874-878

Sintesi

Dopo due millenni di dominio incontrastato da parte della logica aristotelica, nella metà dello XX secolo il matematico azero Lofti Zadeh portò a compimento il cammino iniziato anni addietro da Descartes e portato avanti da vari studiosi tra i quali Russell, Heisenberg e Black, formalizzando la logica fuzzy. A differenza di quella costruita dal filosofo di Stagira, la logica fuzzy non ammette solo due valori di verità per le proposizioni (o bianco, o nero), bensì infiniti valori (i toni di grigio, in aggiunta al bianco e al nero). Nella risoluzione di problemi che hanno origine nell'osservazione del quotidiano, la nuova logica si dimostrò più adatta. Ad esempio, dovendo scegliere se prendere con sé un ombrello o meno, non ci si può limitare a sapere se piove o non piove (aristotelica), ma è necessario conoscere quanto piove (fuzzy); è infatti la "misura della pioggia" a suggerire la risposta.

Nel giro di soli trent'anni, la logica è entrata in modo dirompente nel nostro quotidiano, nei grandi e piccoli elettrodomestici, nell'analisi dei mercati azionari, nella siderurgia, per fare solo alcuni esempi. Particolarmente interessante è stato l'andamento virtuoso dello sviluppo della teoria e delle applicazioni, in cui ognuna diveniva trampolino per l'altra.

È in questo particolare sviluppo che si inserisce la ricerca dei Ponce e di Araiza. Ricerca che da una parte si prefigge la realizzazione di sistemi automatizzati in tempo reale, dall'altra il test e il progresso di tool che sono d'ausilio alla realizzazione di tali sistemi. Pur nella sua (apparente) semplicità, il caso di studio approntato approfondisce i pregi di questa tecnologia e indica quali ne possano essere delle possibili applicazioni.

In un'arena di dimensioni note sono posti due robot (A e B) e una serie di oggetti: alcuni fissi (ostacoli), altri mobili (bersagli). I due automi comunicano tra loro e con un'unità master, posta fuori dall'arena, in tempo reale, costituendo così una rete neurale. L'obiettivo è la rimozione, da parte dei robot, di tutti gli bersagli, che debbono essere riposti in un'opportuna zona; il tutto ottimizzando alcune variabili (quali, ad esempio, tempo e distanza percorsa). Non meno importante nella presente ricerca, sia il software che guida automi e master, sia l'hardware che li costituisce, sono sviluppati con tool appositi in cui sono integrate componenti di logica fuzzy (in questo caso LabVIEW).

Le dinamiche osservate mettono in risalto come logica fuzzy, comunicazione diretta tra le parti (rete neurale) e i tool di sviluppo, riescano a fornire una risposta buona e, talvolta, inaspettata al problema. Questo in parte è dovuto alle decisioni che devono essere prese da tutte le componenti in base all'elaborazione di informazioni che vanno via via aggiungendosi al quadro iniziale (basti pensare alla manifestazione di un nuovo oggetto - ostacolo o bersaglio che sia - precedentemente nascosto dietro un bersaglio rimosso).

L'importanza della ricerca risulta maggiormente comprensibile quando da una parte la si amplia dal punto di vista del numero di attori partecipanti e dall'altra se ne modifica lo scenario in cui è inserita. Un esempio su tutti è la gestione del traffico urbano dove il ruolo dei robot è assunto dai semafori i quali forniscono ad un'unità centrale informazioni sulla situazione di congestione. L'unità centrale, facendo tesoro di queste, rimanda ai semafori una strategia da applicare in cui vengono ottimizzate attese e code. Non solo, i semafori possono sviluppare una strategia locale che risolve situazioni particolari e che si sostituisca a quella individuata dall'unità centrale.

Visti i passi da gigante compiuti negli ultimi decenni in tutti e tre i campi toccati nella presente ricerca (logica fuzzy, reti neurali e tool), c'è da aspettarsi che la messa in pratica su larga scala di quanto evidenziato non sia così lontano da venire. E così come è stato in passato, potremmo non accorgerci subito dei vantaggi che apporterà.