

127-4

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS CIUDAD DE MÉXICO

EDUCATIVE PLATFORM APPLYING INTELLIGENCE CONTROL SYSTEMS THROUGH COLLABORATIVE ROBOTICS TASKS



TECNOLÓGICO
DE MONTERREY



TECNOLÓGICO
DE MONTERREY

Biblioteca
Campus Ciudad de México

Antonio González Espinosa

A00994257

Asesor:
Pedro Ponce Cruz

Sinodales:
Dr. Alfredo Víctor Mantilla Caeiros
Dr. Víctor Manuel de la Cueva Hernández

México Distrito Federal
Enero 2010

Tesis
LB1028-3
G66
2010

1201

12997833

DE WIKI

Special thanks to

Xchel Martínez Galicia, my mother and father, Ev., my professors, Athos Campillo and all that contribute to this project or made it possible.

Index

Introduction	3
Objectives	4
Justification.....	5
Chapter I - Robotics.....	6
1.1 ROBOT	6
1.2 SENSOR.....	7
1.3 PRACTICAL EDUCATION.....	7
Chapter II - Artificial intelligence.....	9
2.1 Meta rules	11
2.2 Rule firing	11
2.3 Forward chaining.....	11
2.4 Backward chaining.....	12
2.5 Machine learning and optimization	13
2.6 Perceptron.....	15
2.7 Multilayer neural networks.....	16
2.8 Optimization Algorithms	17
Chapter III - Fuzzy Logic.....	18
3.1 Fuzzy sets.....	19
3.2 Fuzzy Inference.....	22
3.3 Sugeno-style	27
3.4 Inverse model.....	27
Chapter IV - Fuzzy Logic applied on Lego	29
4.1 Velocity control	31
4.2 Inverse model.....	35
4.3 Instrumentation	37
Chapter V - Artificial Neural Networks.....	39
5.1 Perceptron.....	41
5.2 Multilayer neural networks.....	42

Chapter VI	- Artificial Neural Networks applied on Lego	46
Chapter VII	- Particle Swarm Optimization	48
Chapter VIII	- Particle Swarm Optimization applied on Lego	53
Chapter IX	- Outcomes communicating Lego.....	57
	9.1 Bluetooth communication stages	61
Chapter X	- Educative Platform – PBL	62
	10.1 PBL. Problem Based Learning methodology	62
	10.2 Development.....	65
Conclusions	76
References	78
Activities manual	79
	Artificial intelligence applied on Lego® MINDSTOMRS® NXT Kit with PBL methodology	79
	Activity 0 Introduction to the Lego® MINDSTOMRS® NXT Kit.....	79
	Activity No.1 Sensors approach.....	81
	Activity No.2 Actuators approach	86
	Activity No.3 A robot is born	91
	Activity No.4 LabVIEW and the Lego® Toolkit.....	97
	Activity No.5 On-off control. A Simple approach to control techniques	103
	Activity No.6 Conventional Control. The P action	107
	Activity No.7 Artificial Intelligence. Part 1 Intelligent Control	112
	Activity No.8 Artificial Intelligence. Part 2 Fuzzy Logic control	116
	Activity No.9 Communicating. Making knowledge consistency.....	121
	Activity No.10 Skills integration	126

This work focuses on developing an educational platform through collaborative robotic tasks which can teach different disciplines under one scheme that promotes innovative, fun and challenging learning. A key resource for development the project was the use of artificial intelligence for the different practices, exploiting the potential and abilities of students and robots that will be the creation of each individual and their own responsibility to expand their capabilities.

The learning platform is developed based on the methodology of Problem Based Learning PBL because it fosters self-directed learning and problem solving for different situations, to generate a more practical and experiential understanding. It's underpinned by the theory gained through the investigation and search for solutions during the entire process of analysis performed. It is important to recall that during learning process the role played by the teacher is not focused on traditional teaching, in this case he is a guide and the student is the one who builds his own educational landscape according to his capabilities. All this leads the student to learn in a simple but responsible, imposing him new challenges to try to overcome with new proposals.

For the series of activities carried out were taken into account all important aspects arising from the PBL. They are focused on a subject not very known and sometimes feared, the intelligent control. At this early stage concepts and activities of artificial intelligence were included. Some of them are fuzzy logic, artificial neural networks, optimization algorithms and communication protocols. Importantly, this latter issue is one of the key objectives of the work as the need for solving tasks through collaborative robotics, which simplifies and improves the performance of each individual robot to form an overall result from small achievements of each member of the robotics community.

As proposed in the PBL, activities break the traditional scheme of teacher-classroom-student where theory takes a sequential order with no place to contributions due to the concerns of students. The difference lies in the use of different tools. First, it established a blanket or workplace where the student can develop their practices with panels that recreate various scenarios and also makes use of technological resources such as laptop computers, software of G programming and educational robotics kit Lego[®] NXT[®] MINDSTOMRS for building the robots.

The robot described in this educational learning platform (referred to throughout the following text simply as "robot") is intended to help the child to promote self-learning by proposing new problems to solve for knowledge, giving priority to concrete examples and real applications.

It also seeks to leverage multiple robot designs that can be used to transmit knowledge generated when solving a problem. This situation allows solving complex global tasks through various subtasks, divided among the robotics community where each member contributes a local

process. The result is a system of local education and a comprehensive education system that provides continuous training and dynamic when you need to improve a process. A key aspect that the model searches is its application with children due to the methodology that facilitates the understanding of concepts and abstraction.

The first chapters cover the theoretical basis of knowledge of applied of artificial intelligence in order to structure the proposed methodology for the learning platform. Then there is the chapter that raises the platform and educational model and provides an introduction to the PBL system. The penultimate chapter generally describes the process of implementation and achievements obtained during the work. The last part refers to the proposed activities for the learning.

Objectives

The fundamental objective of the project is to develop an educational platform based on the methodology of PBL to achieve a conducive teaching and learning problem solving in a creative and structured equips students interpersonal skills and teamwork ability, and Finally, physical and motor skills that complement their emotional development, intelligence and capacity for abstraction in an instructive, creative and challenging. Also an efficient and multidisciplinary knowledge is generated through the individual's interaction with peers and robots, promoting the study of robotics and in turn the development of technologies to solve problems in everyday life and industry.

Justification

Educating young students will inspire them to explore and challenge their own potential as they develop fundamental skills for problem solving in a collaborative and innovative ways, leading to personal satisfaction and intellectual achievement. Using Lego children creativity is pursued and satisfies their need to learn. Through actions and situations based on real problems, students acquire knowledge in a didactic and not boring or tedious, barrier-free structures of a classroom or forced to limit their learning and innate curiosity.

With these technological and educational tools the usage of technology promotes interest in science that constitutes to the future development of industry in Mexico. An adequate investment of knowledge, time and effort will help educational robotics to ease the daily life, the country's economic growth and promote the culture of research and technological leadership. It is also a methodology that contributes to inspire creativity. It focuses on providing the means necessary to enable the transfer of knowledge and thanks to the collaborative approach it is further accentuated.

The educational robotics can integrate different areas of knowledge naturally and transdisciplinary, creating a learning environment where objectives should not be considered a priori because they should be constructed during the observation and experience focused on the problem at hand. It is important to recall that under this scheme, the error is part of the learning process and motivates the student helping his own personal growth and satisfaction.

1.1 ROBOT

Even though the word robot seems to be quite recent, it was first used in 1921 by Karel Capek in one of his plays which described a robot that substitutes human beings on their manual labors. "Isaac Asimov in his science fiction stories about robots in the 1940s coined the term robotics as the science or study of robots."¹ It is actually a science that includes a wide variety of areas according to Fuller, such as: mechanical, electrical and electronic systems; computer hardware; and computer software (Fuller, 1999, p. 3).

The fascination for robots comes precisely from its initial purpose. The fact that a machine is capable of substituting a human's work on a more accurate and efficient manner; brings infinite possibilities for many industries around the globe. However the development of new technologies generates fear on human being's about the possible unemployment this situation previously mentioned would cause. Therefore, Isaac Asimov set laws defined as the "rules of robotics". Gerald Norman added a fourth law to this set mentioned above.

1. "A robot must not harm a human being, nor through inaction allow one to come to harm.
2. A robot must always obey human beings, unless that is in conflict with the first law.
3. A robot must protect itself from harm, unless that is in conflict with the first two laws.
4. A robot may take a human being's job. But it may not leave that person jobless!" (Fuller, 1999, p. 4)

It is important to mention the various types of robots. According to Fuller there are seven types of robots: "industrial robots, military robots, show or promotional robots, educational robots, medical robots, domestic or personal robots, and hobbyist robots."²

According to the previous classification, a general description about each type of robot will be presented so they can be easily differentiated. The industrial robots are supposed to aid the laborer. Generally they are manipulated and have no mobility skills in terms of displacement. These robots were built so humans won't have to expose themselves into dangerous, difficult or hazardous scenario. The next type of robots are the military, these robots are clearly built for one single purpose: participate in the army activities. Such as detonating bombs over enemy lines, radio controlled airplanes among others.

¹ Fuller, J. (1999). Robotics. *What is a robot?* (2nd. Ed) Prentice Hall p. 2-3

² Fuller, J. (1999). Robotics. *The Study of Robots* (2nd. Ed) Prentice Hall p. 4

Referring to the show or promotional robots which can be described as entertainer robots because of the friendly image and “attitudes” they appeal to have. Another type of robots is the educational robots which “...can be used to teach the principles of robotics”³. The medical robots are easily identified because of the activity they perform or maybe substitute. They can even carry out functions from a disabled person. The medical robot is one of the most congratulated types of robot because the development of new technologies for making them better brings benefits for human beings and assures a higher life quality level.

The domestic robots are basically used for home activities. These robots are particularly interesting because of the variety of activities they can perform such as baby-sitting, cleaning and even home protection.

1.2 SENSOR

The importance of a sensor shows out in defining the primary difference between a dumb machine and an intelligent one. The function of a sensor in a machine is the same as in a human being. If a sensor detects any change on an environment a human may adapt to the new atmosphere. A sensor adds value to any device ensuring to expand the abilities of recognition and detection of external circumstances.

According to the educational field, sensors are basic for educational robots to serve a better purpose and obtain better results on its application. Not all sensors perform the same function and that is how we can divide them into three classes:

- Internal Sensors
- External Sensors
- Interlocks

The possibilities to expand the implementations of robots once sensors are being used are infinite. Another advantage of the usage of sensors is the compilation of data it can be done from a device that uses one. An industrial robot for example may be able to determine the frequency in which a failure is committed in a line of production.

1.3 PRACTICAL EDUCATION

³ Fuller, J. (1999). Robotics. *Educational Robots* (2nd. Ed) Prentice Hall p. 16

Now days we find more and more necessary to introduce researchers and students into practical problems so their possible solutions can be as realistic as the very problem itself. Education is suffering a significant change on its basis; this is of course for the benefit of researchers and the society by all means. More specific, in the field of robotics the practical education is not only innovative but it is necessary. "For the study of advanced robotics, integration of existing technologies is crucial, including mechanical and electrical engineering, computer science, and human science and engineering."⁴ To choose a practical education system results in a clear understanding of technologies.

The stimulation a student can obtain from a kit that contains very simple objects so they can build robots is very significant. "Products as LEGO Dacta and LEGO CyberMaster include cables and radio equipments that allow connecting the robot with your personal computer. This allows the user to control their invention."⁵ According to Seymour Papert, in this type of education, the key learning point is the actions made from the student about constructing and manipulating the robots. The need to explore new methods of acquiring knowledge is as important as participating in the practical field so the theories can be proofed. Receiving the experience will mature the students and will make them differentiate between the classroom and real situations.

"Experiments with Evolutive Robotics show how the simulation of procedures in evolution, make possible the development automation of the intelligent control systems for real robots. Simulating the natural process predicted by Darwin's theory of evolution and playing with various components, the students achieve a better theoretical comprehension than other methods."⁶ The complete integration of experience and knowledge will obtain better results in the perspective of the students.

Learning while doing Practical Robotics Education will give the students a wider point of view about the accuracy they may achieve in order to result in better devices that fulfill the needs. The point is trying to manage as much variations as possible without knowing what is going to happen in the near future (like real life experience). The students will improve mechanisms and add artifacts that their robot "may use"; the creativity will be endless, exploring new possibilities will be necessary.

According to the Institute of Psychology, National Research Council and Faculty of Psychology from the University of Naples in Italy and the Department of Computer Science from the LEGO Lab in Denmark the easiest way to start experimenting with robots is to construct LEGO machines with sensors, ACCIONADORES and control capacities. The main reason of this affirmation is the easy familiarization the student can develop with an accessible tool kit such as LEGO Dacta or LEGO Cybermaster.

⁴ Learning while doing Practical Robotics Education (2001)
[http://0ieeexplore.ieee.org/millennium.itesm.mx/stamp/stamp.jsp?tp=&arnumber=932756&isnumber=20189\(electronic\)](http://0ieeexplore.ieee.org/millennium.itesm.mx/stamp/stamp.jsp?tp=&arnumber=932756&isnumber=20189(electronic)) Obtained on November 20th.

⁵ *La robótica como herramienta para la educación.*
<http://www.donosgune.net/2000/dokumen/EduRobSp.pdf>

⁶ <http://www.donosgune.net/2000/dokumen/EduRobSp.pdf> p. 15. Obtained on November 19th.



During the last century, the technology development has made possible unimaginable discoveries and helped the human race through the path of evolution. In this conquest of knowledge there has been present an issue of ethical and philosophical discussion, to build a machine that can mimic or exceed human capabilities, in other words, create intelligent systems. Obviously these non-human machines are meant to have minds and think. In fact, some machines have already been built that we can call 'intelligent'. Let say what 'intelligence' and 'think' mean by some dictionary definitions:

- Someone's intelligence is their ability to understand and learn things.⁷
- Intelligence is the ability to think and understand instead of doing things by instinct or automatically.⁸
- Thinking is the activity of using your brain to consider a problem or to create an idea.⁹

According to the definitions intelligence may be or not only a human quality process. This gives the possibility to accept that machines can also be intelligent, but regarding that the process of intelligence implies the action of thinking leads to the question 'Can machines think?'¹⁰

Research in artificial intelligence (AI) would help understanding intelligence and human thinking process itself and provide the knowledge necessary to make machines that act as humans but smarter and better to interact in the day life side-by-side with humans to their benefits. The most ambitious project is to equal 'human mental capabilities including reasoning, understanding, recognition, creativity and emotions.'¹¹

Nowadays, intelligent systems are also known as *Expert systems* and the key of them are the knowledge-based systems (KBS). The difference of a conventional system and a KBS is the way the knowledge is used to generate it, in the first one, the knowledge is intimately intertwined with the program or software by which the application is controlled; meanwhile in the KBS there is a specific area for the knowledge referred as a module called the *knowledge base* explicitly separated from the main control. This separation makes it easier to add new knowledge leaving the door open to update the knowledge base of the system at any time during the program's lifetime, as the human brain learning behaviour, accumulating knowledge and experience.

⁷ Essential English Dictionary, Collins, London, 1990

⁸ Essential English Dictionary, Collins, London, 1990

⁹ Essential English Dictionary, Collins, London, 1990

¹⁰ Artificial Intelligence, Netnevitsky, England, 2005

¹¹ Intelligent Systems for Engineers and Scientists, Hoggood, USA, 2001

The explanation of why they are called *Expert systems* is the fact that an 'expert' has to put the knowledge on it in order to make the system become intelligent in some way for a specific domain. If this role lacks of expertise the system would fail no matter how complex it is. That is why it is important to recognize an expert to fulfil the needs for the system.

The module that controls the system is called the *inference engine* and the one who carries out the reasoning using the *knowledge base*.

When we try to express something we usually never give an algorithm as an answer, we use *rules* and that is what these systems take advantage of. Instead of trying to make an algorithm that could represent mental process, they use rules where knowledge can be formulated as simple statements. The syntax of a rule is:

IF <antecedent> **THEN** <consequent>

In the same rule we could have multiple antecedents each represented by a linguistic object and its value, 'joined by the keywords AND (conjunction), OR (disjunction), or a combination of both'.¹² Also there could be multiple consequents. The relation between them can represent relations, recommendations, directives, strategies and heuristics (Durkin, 1994).

IF	<antecedent 1>	IF	<antecedent>
AND	<antecedent 2>	THEN	<consequent 1>
AND	<antecedent m>		<consequent i>
THEN	<consequent>		

IF <antecedent 1>
OR <antecedent 2>
OR <antecedent k>
THEN <consequent>

To develop an expert system one could use software called *Expert System Shell*, which is an expert system without knowledge. This gives the advantage that no programming is necessary instead you have to add the knowledge for the specific problem.

An expert system shell has these main components:

¹² Artificial Intelligence, Netnevitsky, England, 2005

- The knowledge base: as we mentioned, this is the set of rules in the IF-THEN structure.
- The data base: facts to match the conditions with the rules.
- The inference engine: the reasoning module and the link between the knowledge and the facts.
- The explanation facilities: the responsible of giving explanations to the user of how a why a rule and facts are reached.
- The user interface: interface between user and system.
- The external interface: allows to work with external data or a conventional program.
- The developer interface: knowledge base editors, inputs and outputs and debugging.

It is important to recall that alike human expert can make mistakes, an expert system is not exempt of making them, even though they are trust worthy and it is only a matter of being aware.

2.1 Meta rules

Meta rules are the ones who dictate how to apply the knowledge. These rules are not concerned specifically with the knowledge of the application but how the rules should be applied, 'rules about rules'.

2.2 Rule firing

A rule is fired when a fact matches the condition of the IF part of the rule, making the consequent of the THEN part to be executed as well. This process is made by the inference engine and leads to inference chains while matching the IF antecedent to the facts. There are two ways for a rule to be executed, forward chaining and backward chaining.

2.3 Forward chaining

Also known as data-driven strategy, this inference consists in applying the data in the fact base, where only the topmost rule is executed. Each time a rule is fired, new data is added to the fact base, as only one rule is fired on a given cycle, all rule can be comprised if the conditions to be triggered exist even if the solution does not correspond directly to the goal.

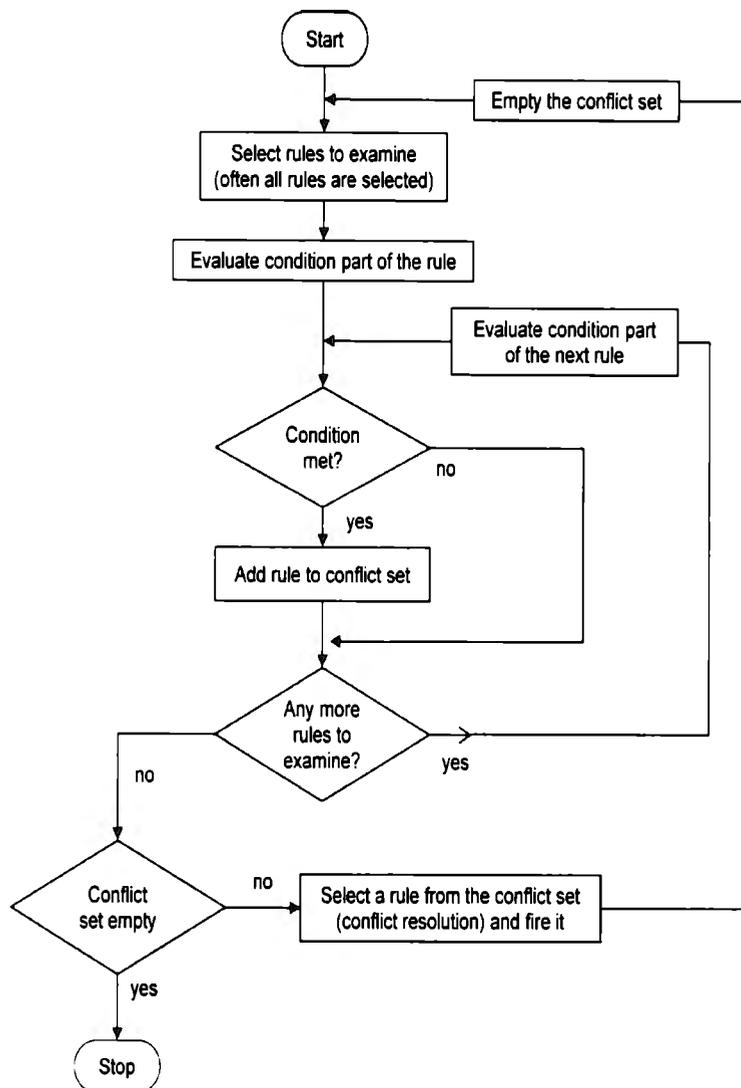


Figure 2.0 Forward chaining flow diagram¹³

2.4 Backward chaining

Also known as goal-driven strategy, this method supposes that the goal exist as a hypothesis and tries to prove it. When an action on the THEN part satisfies the match, the rule is chosen and the fact can be proved. However, this usually never happen because when a fact matches the IF condition may be not the antecedent for the goal and a new sub-goal must be set up. The new sub-goal is now tried to be proved by the inference engine and searches in the

¹³ Intelligent Systems for Engineers and Scientists, Hoggood, USA, 2001

knowledge base to repeat the process putting aside the past rule and focusing on the new sub-goal. Process continues until no rule satisfies the sub-goal and fails or the main goal can be proved and the whole line is completed.

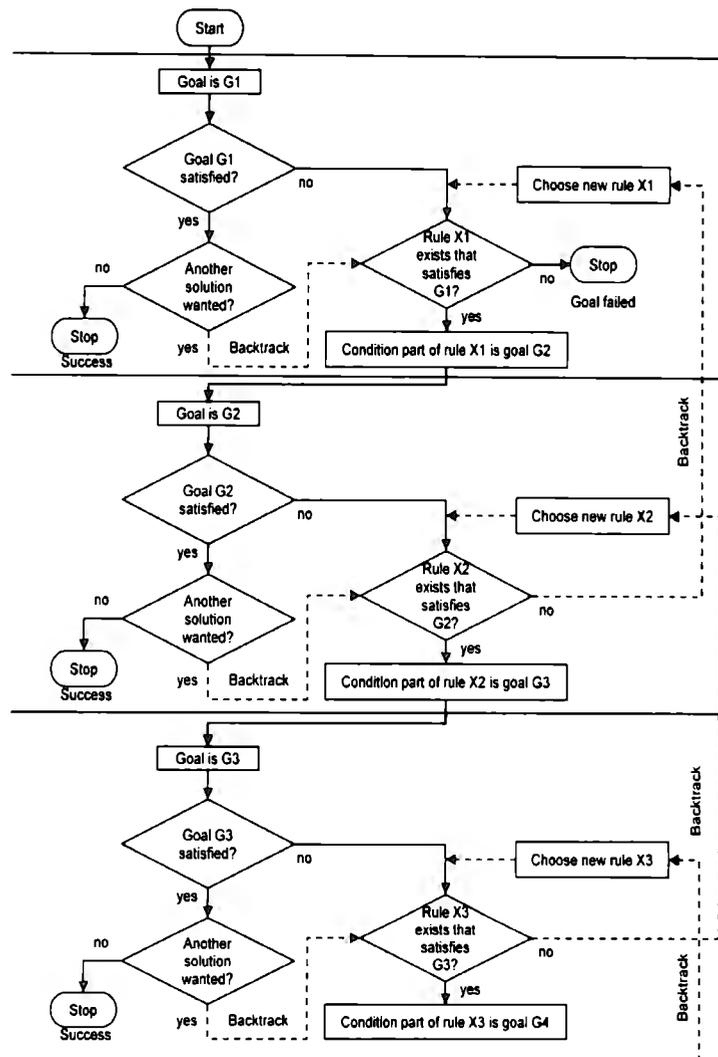


Figure 2.1 Backward chaining flow diagram¹⁴

2.5 Machine learning and optimization

One difficult aspect when developing an intelligent system is to determine exactly the most suitable values for the membership functions, even though an expert is required, the parameters could not be good enough to obtain the best behaviour from the system. This is why

¹⁴ Intelligent Systems for Engineers and Scientists, Hopgood, USA, 2001

learning techniques and optimization algorithms are extensively used to obtain the best performance.

A possible answer could be found in the *Artificial Neural Networks (ANNs)*. 'A neural network can be defined as a model of reasoning based on the human brain.'¹⁵ Let say how the human brain works first. The basic nerve cell is the neuron, and has multiple connections to other neurons in order to create a densely interconnected structure or network. The process in the net is made by an electrochemical reaction which conducts the impulses to propagate the information and make a specific task. The whole process is made in parallel and is highly complex and nonlinear. Each neuron computes information independently of the other neurons but its inputs could be outputs form another neuron.

Artificial neural networks are good for nonlinear problems and those which we may know the inputs and outputs rather than the function.

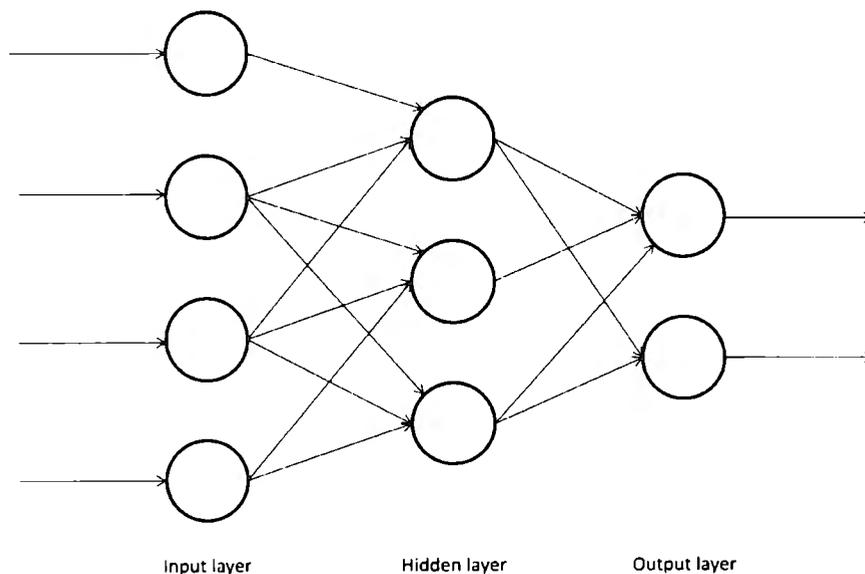


Figure 2.2 Artificial Neural Network scheme

An artificial neural network is analogous to the biological but less complex. It has processing units called neurons and their connections are weighted to recreate the strength of the links; these are the basic means of long-term memory. Each neuron has a function to compute the information called the activation function. The neurons are hierarchy layered and each layer has one or several neurons. One of the characteristics that make powerful the artificial neural

¹⁵ Artificial Intelligence, Netnevitsky, England, 2005

networks is the plasticity, changes in the strength of their connections that makes the right answer to be strengthened and wrong answers to be weakened.

There are several activation functions for different problems, but most common are step function, sign function, sigmoid function and linear function.

A single neuron could have several inputs but one single output, and computes the output from the sum of the weighted inputs evaluated with the activation function.

$$f = f\left(\sum_{i=1}^n x_i \omega_i\right)$$

The learning process of a neural network could be *supervised* or *unsupervised*. The training process is the process to find the correct weights which links the interconnections.

2.6 Perceptron

Is the simplest structure of a neural network, in fact it is just one neuron. By definition it has two inputs, and is capable to classify elements that are linearly separable. It could have a bias in order to adjust the line, in case it does not cross the zero.

The process to train the perceptron consists of four steps and the training data should be applied:

1. Initialization: weights and bias are set randomly within a certain range.
2. Activation: with the training inputs and weights compute the output.

$$f = f\left(\sum_{i=1}^n x_i \omega_i\right)$$

3. Weight training: update the weights

$$\omega(k+1) = \omega(k) + \alpha x(k) * e(k)$$

Where α is the learning rate, and $e(k)$ is the error given by:

$$e(k) = [V_d(k) - f(k)]$$

Where $V_d(k)$ is the desired value.

4. Iteration: go back to step 2 until convergence.

2.7 Multilayer neural networks

Multilayer network can perform more complex task rather than just classifying. It consist of one input layer, which is normally the input itself not a neuron, at least one hidden layer, the one that propagates the signals; and the output layer which leads to the output of the entire network. The simplest network with one hidden layer can perform continuous functions, with two hidden layers even discontinuous are possible to represent.

The learning process in a multilayer network is basically the same way in a perceptron, but in this case every neuron contributes with its own weight and for more than one output. All the weights are adjusted to reduce the error when the training pattern is applied as in a perceptron. There are many algorithms to train a multilayer network, but the most popular method is back-propagation.

In this method the error signal is propagated backward to previous layers, starting from the output. In this case an error gradient is produced to adjust weights from all layer behind the output which is the only that has directly the proportion of error produced due to certain data training. The steps for the method are similar to the perceptron:

1. Initialisation: set randomly weights and bias.
2. Activation: calculate outputs.
3. Weight training: this step is different due to complexity of the new error gradient and equations, but in practical terms is the same and it will be discussed in chapter VI.
4. Iteration: go to step 2 and stop until convergence.

2.8 Optimization Algorithms

The need of algorithms to refined parameters progressively with experience is important to find the correct rules, relationships, membership functions for the best performance of our designs. The best way to understand how this process works is the relationship that relates optimization and some mathematical procedures, these techniques tries to determine maximum or minimum values of a function for one or more variables, when minimized is often called as a *cost function* and when maximized is often called *fitness function*. 'In fact the two approaches are equivalent.'¹⁶ When both functions are used it is called objective function.

The possible solutions for a certain variable should be considered as the *search space* or *parameter space* and it would be n-dimensional with respect to n-variables or parameters. For example for a single variable the search space is one-dimensional.

For a better approach to the maximum and minimum the values are associated with each point in the search space. Graphically this is a plot that shows how functions of fitness and cost 'involved fining peaks in the fitness landscape or valley in the cost landscape.'¹⁷ There could be found two types of values, *Local optima* or *Global optima*. In order to achieve good optimization global optima must be found, because this value will lead to the best answer, meanwhile a local optima will provide a better approach to the solution, because individual points will not necessarily provide the best solution with respect to its neighbours.

For this particular work we will explain in detail in chapter VII one of the most popular algorithms called *Particle Swarm Optimization*.

The particle swarm optimization method consist, as all other techniques, in finding maximum and minimum for variables given to control a system. The process involves a population, where every single individual has certain attributes and contribute to the refining process. Analyzing these particles we can obtain the global optimum in order to achieve the goal.

The behaviour of the particles is based on the behaviour of biological communities like ants or bees, where any individual can became a global optimum at any moment to find the best way to the food and vice versa. This behaviour has been extensively studied and lot of research has been done to recreate this pattern and develop new algorithms.

¹⁶ Intelligent Systems for Engineers and Scientists, Hopgood, USA, 2001

¹⁷ Intelligent Systems for Engineers and Scientists, Hopgood, USA, 2001

Fuzzy logic is based on classical logic and the propositional calculus. The very first impression to this theory was a distrust posture as the fuzzy word is usually used in not a good sense. The theory then says that 'Fuzzy logic is not a logic that is fuzzy, but logic that is used to describe fuzziness.'¹⁸ Classical logic uses crisp values, like two-valued Boolean logic, in order to be characterized either true or false. This main difference is that fuzzy logic takes advantage of, using multi-valued system. Experts usually represent things relying on common sense because things are not always either completely true or completely false. Let say that sometimes problems are partly true and partly false, so we might say is a vague or ambiguous term which describes it the best. That is why fuzzy logic works with degrees of membership and truth values to categorize each element to be evaluated, and deal with imprecise inference with linguistic terms.

In order to understand the propositional calculus on fuzzy logic, we have to define two concepts: possibility and necessity. The first is a measure to which a proposition might be true. The second is the credibility that a proposition must be true. This may be confused with the truth value, a measure that can be within the interval [0, 1] and says how truth is a proposition or the proportion of its validity. Alike Boolean logic, when we make comparisons in fuzzy logic, classic logic is not valid as fuzzy elements could have truth values anywhere between 0 and 1. So we must use other formulas to compute AND, OR and NOT, like:

X AND Y

Zadeh operator:

$$X \text{ AND } Y = \min(X, Y)$$

Probabilistic operator, assuming Independence:

$$X \text{ AND } Y = X * Y$$

Bounded difference operator:

$$X \text{ AND } Y = \max(0, X + Y - 1)$$

X OR Y

Zadeh operator:

$$X \text{ OR } Y = \max(X, Y)$$

¹⁸ Artificial Intelligence, Netnevitsky, England, 2005

Probabilistic operator, assuming independence:

$$X \text{ OR } Y = X + Y - X * Y$$

Bounded sum operator:

$$X \text{ OR } Y = \min (1, X + Y)$$

NOT X:

$$\text{NOT } X = 1 - X$$

We will be using some of these comparisons to explain fuzzy logic, but there are many ways to do it. The NOT will be always the same. Specifically we will use min-max fuzzy logic as Lotfi Zadeh originally proposed to obtain truth values, min for AND, max for OR.

Also we have to state that for our analysis on fuzzy logic we making deductive inference with the tautology of *modus ponens*:

$$(a \wedge (a \Rightarrow b)) \Rightarrow b$$
¹⁹

This can be understood as: "IF a is true and a implies b then b is true".

3.1 Fuzzy sets

To describe fuzzy logic in a mathematical way, a fuzzy set is defined by its grade of membership, written as the characteristic function denoted by $\mu(x)$. This means that $\mu(x)$ represent the membership function of a fuzzy member for a universe expressed in real number, because we assumed that fuzzy sets describe words in numbers. Even though there are two systems in fuzzy logic, discrete fuzzy sets, and fuzzy numbers, the approach we are giving is that fuzzy logic helps to describe things in a common sense like an expert does and we use numeric discrete fuzzy set as linguistic variables, where 'the names of its members, known as linguistic values, and for each linguistic value, a membership function like those for fuzzy numbers.'²⁰

An example of a linguistic variable might be *Distance*, whose members are *Short*, *Medium* and *Long*, and representing these terms with its own membership function could be done by several mathematical expressions and representations. The universe of discourse is the range of possible values of a linguistic variable, and is different for a specific linguistic variable. Also is common to use modifiers such as *very*, *quite*, *slightly* to quality the linguistic variables as it could be fuzzy. These modifiers are called hedges and have their mathematical representation, but we

¹⁹ Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems, Chen and Pham, USA, 2001

²⁰ Fuzzy Expert Systems and Fussy Reasoning, Siler and Buckley, USA, 2005

are not discussing them. For explain purposes we are showing only those functions used in the entire work which are *triangular*, *left-shoulder* and *right-shoulder* functions.

Evaluating inputs to obtain the degree of membership depends on the function; this is the same as finding the truth value. For simplicity we use the same universe of discourse.

Triangular function

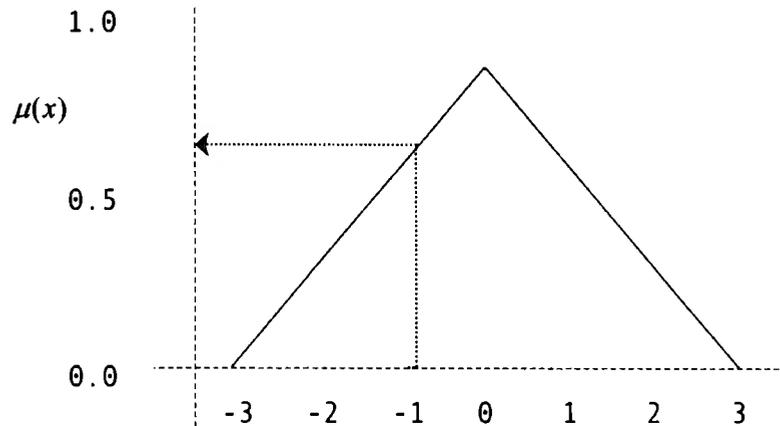


Figure 3.0 representation of a triangular function with universe of discourse from -3 to 3

For this particular function the corresponding values follows this simple equations:

$$\mu(x) = 0 \text{ for } x < \alpha$$

$$\mu(x) = \frac{1}{\beta - \alpha} (x - \alpha) \text{ for } \alpha < x < \beta$$

$$\mu(x) = \frac{-1}{\gamma - \beta} (x - \gamma) \text{ for } \beta < x < \gamma$$

$$\mu(x) = 0 \text{ for } x > \gamma$$

Where α is the beginning point of the edge with positive slope of the triangle, β is the top the triangle and γ is the end point of the edge with negative slope. For this case $\alpha = -3$, $\beta = 0$ and $\gamma = 3$. For the triangular function we have selected the value -1 which corresponds to 0.66.

Left-shoulder function

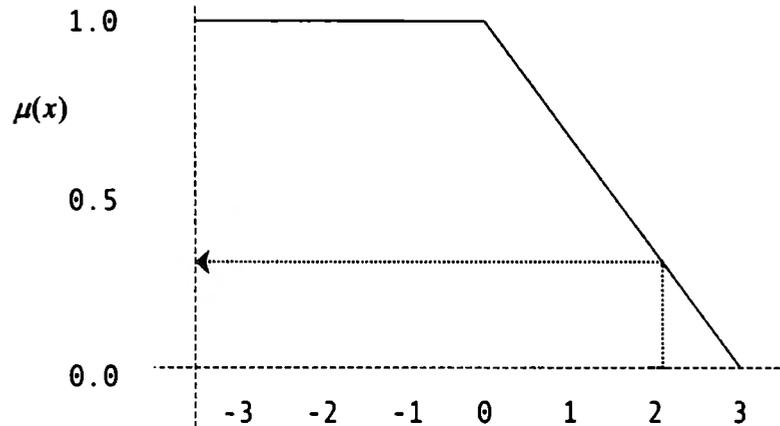


Figure 3.1 representation of a left-shoulder function with a universe of discourse from -3 to 3

This function is commonly the last one at the left side of all functions because everything on its left side has a degree of membership of 1 relative to that linguistic value. Equations are:

$$\mu(x) = 1 \text{ for } x < \beta$$

$$\mu(x) = \frac{-1}{\gamma - \beta} (x - \gamma) \text{ for } \beta < x < \gamma$$

$$\mu(x) = 0 \text{ for } x > \gamma$$

In the example we have the values for $\beta = 3$ and $\gamma = 0$, and the degree of membership taking the value of 2 is 0.33.

Right-shoulder function

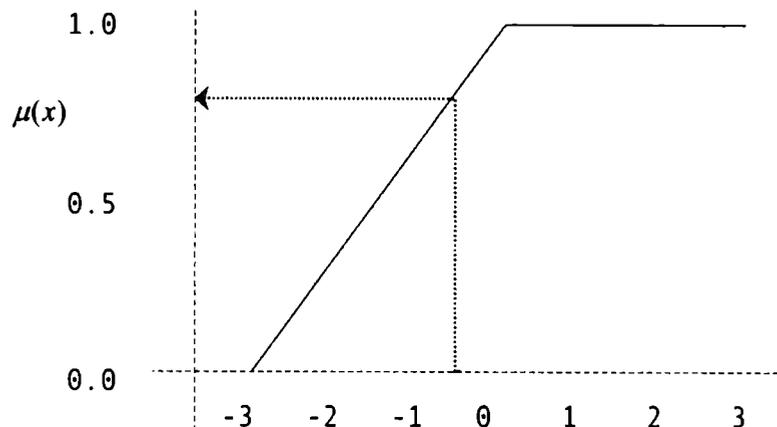


Figure 3.2 representation of a right-shoulder function with universe of discourse from -3 to 3

As the left-shoulder function, this function is most commonly used at the end of the right side of all functions, and also everything on its right side has a degree of membership of 1. The equations for the function are:

$$\mu(x) = 0 \text{ for } x < \alpha$$

$$\mu(x) = \frac{1}{\beta - \alpha} (x - \alpha) \text{ for } \alpha < x < \beta$$

$$\mu(x) = 1 \text{ for } x > \beta$$

In the example we have the values for $\alpha = -3$ and $\beta = 0$. The result with value of -0.5 will be 0.83.

3.2 Fuzzy Inference

'Fuzzy inference can be defined as the process of mapping from a given input to an output, using the theory of fuzzy sets.'²¹ There are many inference techniques with different reasoning that can be applied to a system, but as we chose the most commonly used inference technique, the so-called Mamdani method, we are only describing this method to understand the inference on Fuzzy logic, and we are mentioning the Sugeno-Style to make a comparison between Mamdani method and Sugeno method.

²¹ Artificial Intelligence, Netnevitsky, England, 2005

Mamdani method was implemented for the first time by Professor Ebrahim Mamdani of London University on a steam engine and boiler to control the system. This method consists of four steps that make the whole process to get an output from the given input.

1. Fuzzification
2. Rule evaluation
3. Aggregation of the rule outputs
4. Defuzzification

1. Fuzzification

The first step is to obtain fuzzy values to apply the rules proposed. This means that we first obtain the degree of membership to all inputs; all inputs must be fuzzified over all the membership functions. The inputs in this case are given as crisp values, always a numerical value limited to the universe of discourse selected.

The range of the universe of discourse is determined by the expert judgement. This is a key element to obtain the best behaviour for the fuzzy system. Different fuzzy systems use different universe of discourse as they have different crisp inputs, but sometimes are easier to determine the range. When we have elements that could be measured directly (height, distance, speed, etc.) we have to limit the range according to the values for the inputs, otherwise the range should be estimated based on the expert judgement.

2. Rule evaluation

The second step is made by applying the fuzzified inputs to the antecedents of the fuzzy rules. In this process we have to take in consideration multiple antecedents and use the convenient operator (AND or OR) to obtain a single number. As in the previous chapter we use the operator AND to evaluate the conjunction, with the min method:

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$$

And to evaluate the disjunction we use the OR operator with max method:

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$$

Then the evaluation is applied to the consequent function, it can be either clipped or scaled, each method has different uses, but both correlate the truth value of the antecedent to the rule of the consequent.

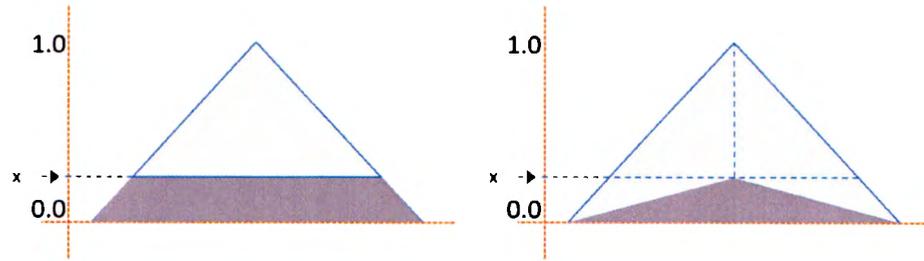


Figure 3.2 consequent function: a) clipped, b) scaled²²

3. Aggregation of the rule outputs

In this process we have all membership functions of the antecedents clipped or scaled, so we unify the rules in order to make one single fuzzy set for each output variable.

4. Defuzzification

This is the last step of the inference process, and is specially needed because at the end we must communicate the output with the system and has to be not in a linguistic variable but a scalar that the system can compute. Several methods are used to defuzzify the output; among them we list some: the *average maximum* method, the *weighted average maxima* method and the most commonly used the *centroid* method. This last method is the one we are explaining as is the most popular one and the one we use in practice for our particular case.

The *centroid method* is mathematically the **centre of gravity** (COG) for the aggregation of the output membership functions, where one could draw a line which divides exactly in two equal masses. It is calculated by the equation:

$$COG = \frac{\int_a^b \mu_A(x)x dx}{\int_a^b \mu_A(x) dx}$$

The representative point is found on the interval ab . In theory we calculate the COG over a continuum of points but in practice is preferred a simplified method using samples to obtain a sum with the equation:

$$COG = \frac{\sum_{x=a}^b \mu_A(x)x}{\sum_{x=a}^b \mu_A(x)}$$

This approximation is obtained only if we 'assign singleton membership functions to the output linguistic variable.'²³ Recall that a singleton function is the one who has a degree of membership equal to one only for one value of its argument, everywhere else is zero.

²² Artificial Intelligence, Netnevitsky, England, 2005

²³ Fuzzy Expert Systems and Fussy Reasoning, Siler and Buckley, USA, 2005

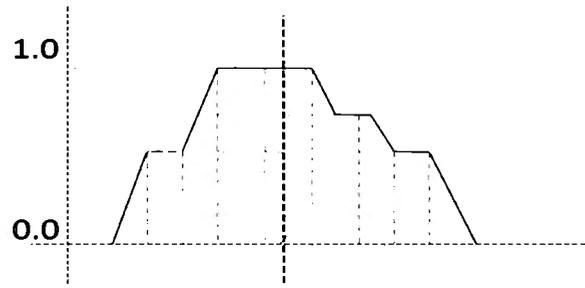


Figure 3.4 centre of gravity. Aggregation of the output membership function

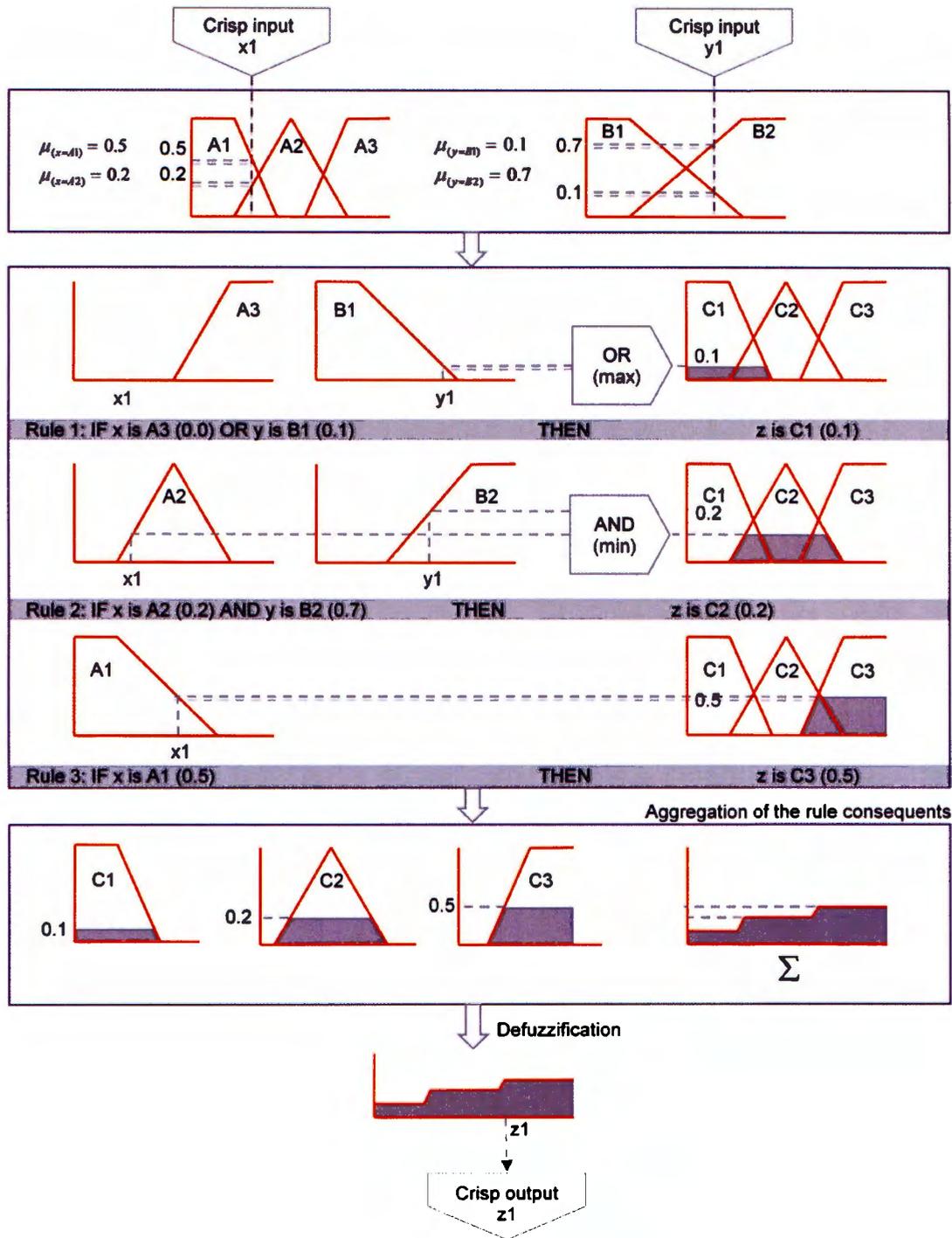


Figure 3.5 Mamdani-style fuzzy inference method²⁴

²⁴ Artificial Intelligence, Netnevitsky, England, 2005

3.3 Sugeno-style

Sugeno-style was first introduced by Michio Sugeno. It is very similar to Mamdani-style, but differs on the rule consequent, the action of the THEN part. This is the main difference that makes Sugeno-style useful and gets an advantage for particular purposes as 'optimization and adaptive techniques, particularly for dynamic non linear systems.'²⁵ It implements a polynomial function of the input variable to express the consequent instead of using another linguistic member. This fact gives the possibility to represent in mathematical language, the description of the action that the THEN part implies, in other words it could model the problem in such a sophisticated manner than words alone could not do.

3.4 Inverse model

The inverse model is used in circumstances where an open-loop control is required and when there is no need of a mathematical model for the system, in other words, less complexity is required. This means that it would be useful when a fast control is needed to obtain an output rapidly. Because there is no feedback loop, the system takes less time computing the answer.

The first step is to implement a control, in this case using fuzzy logic, to model the system and obtain the outputs due to the inputs. Every possible value must be computed to determine all outputs of the system. Obtaining this data gives the possibility to know all the actions to control the system; this is why no feedback is needed.

Once we have the data for the system, we have to rewrite the rules backwards. Let say we have this rule given in the fuzzy control:

IF A and B THEN C

This rule should be applied as:

C IF A and B

²⁵ Artificial Intelligence, Netnevitsky, England, 2005

Rewriting the rule ends up with an output that will be the input of another controller to obtain the final output for the system. This shouldn't be considered as a waste of time because of the work done to obtain every single response of the system because once you do the whole process the system is modelled and the control will be faster.

The only thing that must be remembered is that this model is only for the system that was modelled and if the parameters change, the system must be changed as well as the data for the control, so the process should be done again

This chapter is about the work done with the Lego MINDSTORMS kit using fuzzy Logic to control some of our robots from the community. Background is omitted as in the previous chapter was widely explained or at least the theory used for this work. We have to remember that Lego MINDSTORMS has already specific software to develop programs included in the kit. As we know this is a very basic but powerful tool to create robots that can do lots of things, even though it is still an educational software less powerful than the system that is based on, LabView from National Instruments, one of the most popular graphic environment to develop application for education and industry, and the engine of all our applications on intelligent control.

For the Lego system with fuzzy logic, we developed several controls to control the robots for a specific task to achieve the collaborative global goal. The first and one of the most important parts for this work was to program the basic functions used in fuzzy logic, the triangular function, the left-shoulder function and the right-shoulder function.

All functions were done in a single 'vi' so we could include them in other programs any time and number we need.

Left-shoulder function

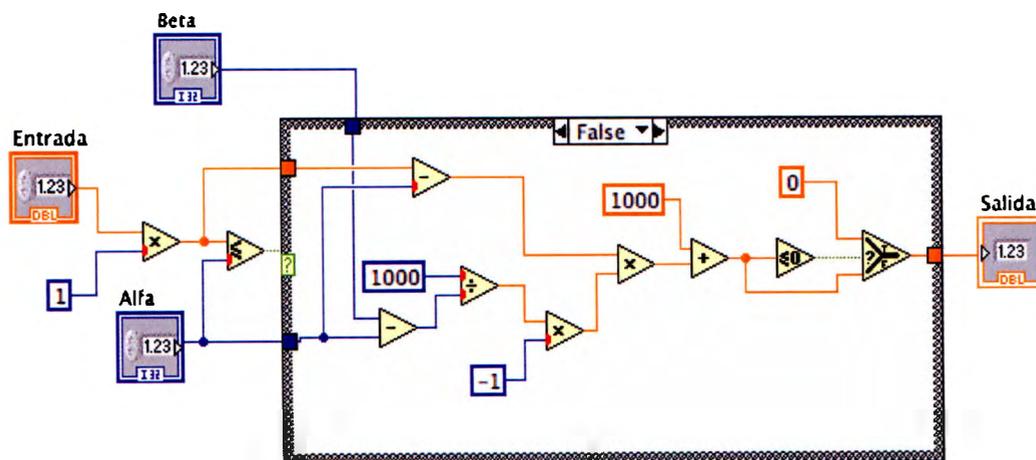


Figure 4.0 Labview code for left-shoulder function implementation

Triangular function

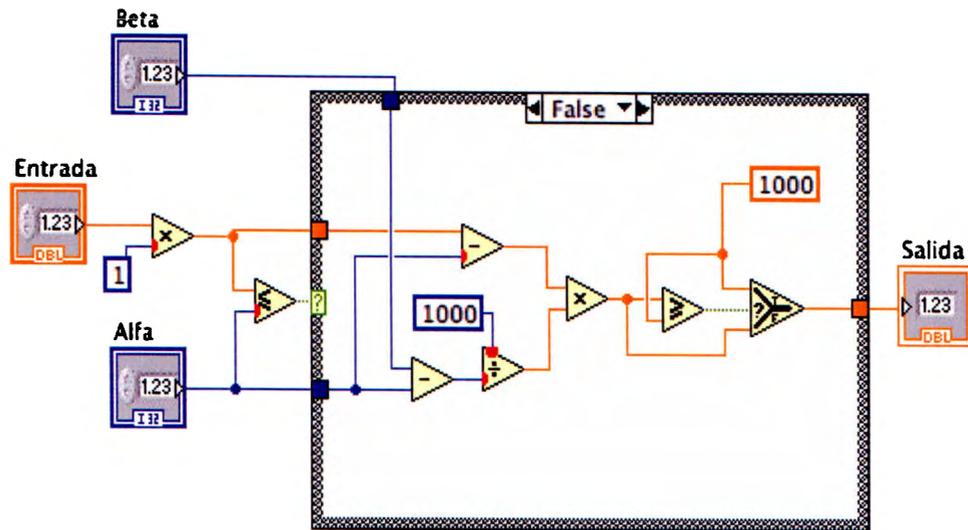


Figure 4.1 Labview code for triangular function implementation

Right-shoulder function

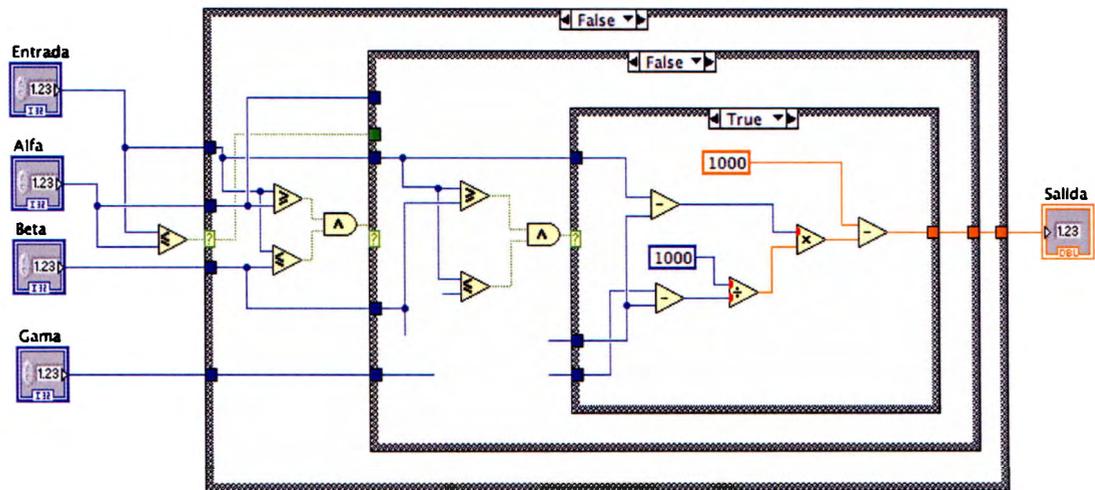


Figure 4.2 Labview code for right-shoulder function implementation

4.1 Velocity control

One of our best controls was achieved with this fuzzy logic velocity control. As we mentioned we used the Mamdani-style inference technique with tautology of modus ponens. In order to avoid static and dynamic obstacles the velocity of each motor was controlled applying three actions: reduce, boost or break the movement. The action done by the Lego was specifically the control of the PWM (Pulse Width Modulation).

Inputs for this application were two ultrasonic sensors mounted on the front of the robots, placed with an angle of about of X degree with respect to each other. This configuration magnifies the angle of vision, and improves the behaviour of our robots.

Below we can observe part of the code for the first step, the fuzzification of the fuzzy inference, where the first data on the left are the values of alpha, beta and gamma variables for the membership functions, followed by the functions for each motor left and right. Next to the functions we have the rule evaluations, the aggregation of the rule outputs and the last step, the defuzzification using the centroid method.

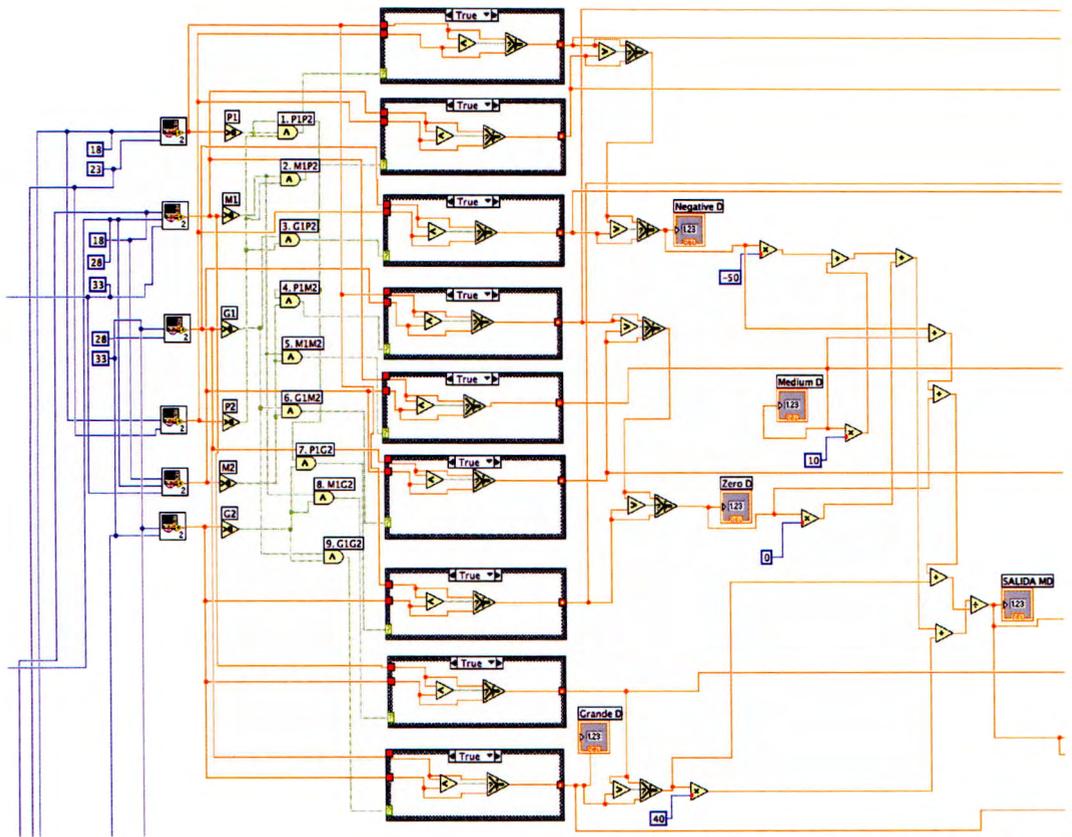


Figure 4.3 Labview code, part of the velocity control: Fuzzification, Rule evaluation, Aggregation of the rule outputs

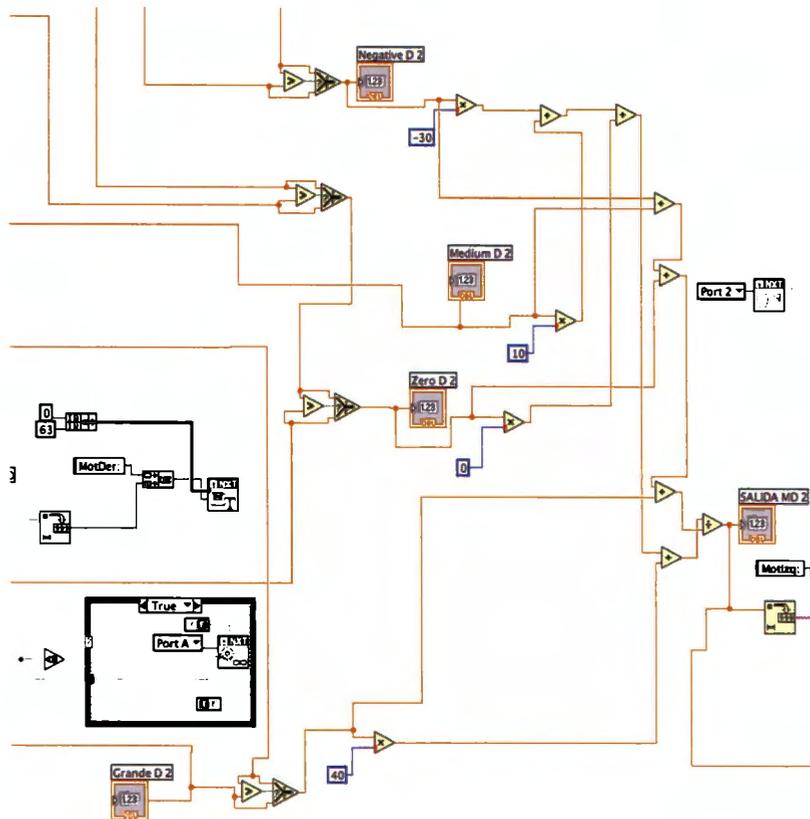


Figure 4.4 Labview code for velocity control: defuzzification

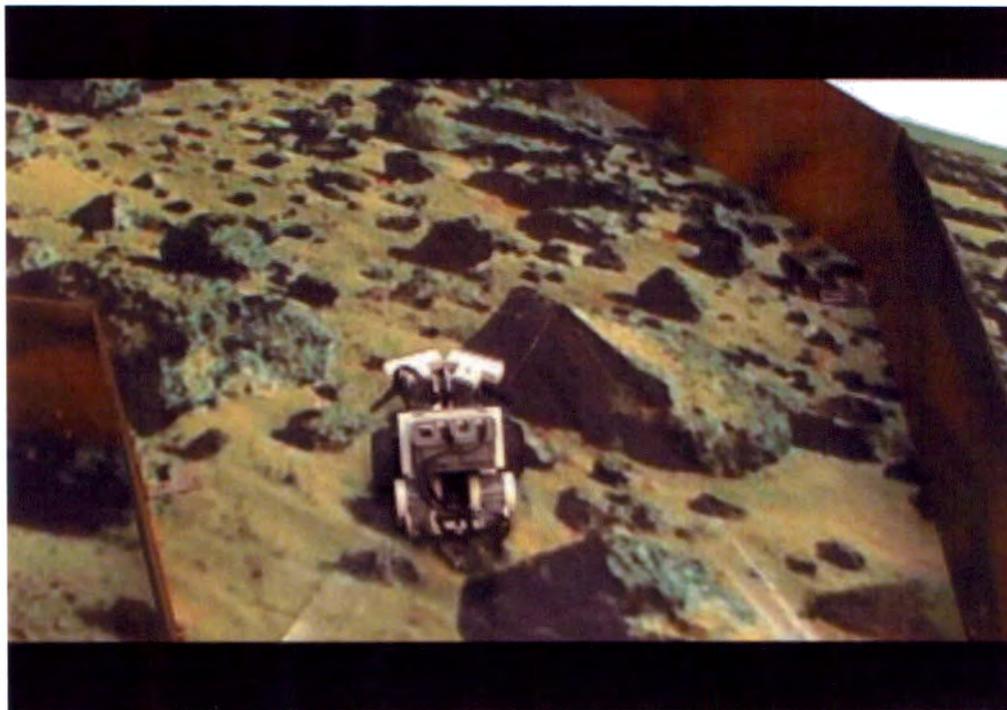


Figure 4.5 robot in action with the velocity control

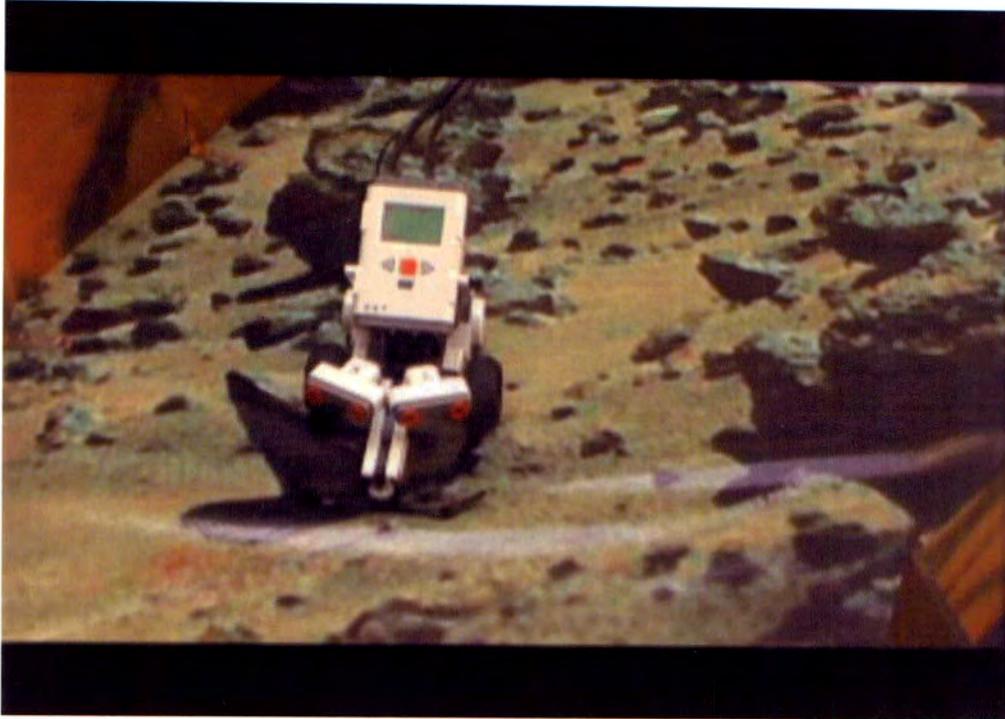


Figure 4.6 robot with the velocity control



Figure 4.7 robot reaching the goal

4.2 Inverse model

For labyrinth solving task, we also implement a controller with inverse model²⁶ fuzzy logic in order to model the behaviour of the robot during the trajectory. This control is specifically applied to motors. It models the degrees of angle in both directions, clockwise and counter clockwise, needed to avoid the obstacle in a range of 0° to 360°. After the fuzzy model was obtained, we constructed the inverse model.

Below, part of the code is shown, beginning with the inputs for gamma, beta and alpha for the membership functions, the membership functions. This is the fuzzification, then the rules evaluations followed by the aggregation of the rule outputs and defuzzification for the first process which determines the degrees needed to control the turn of each motor due to a certain input distance read.

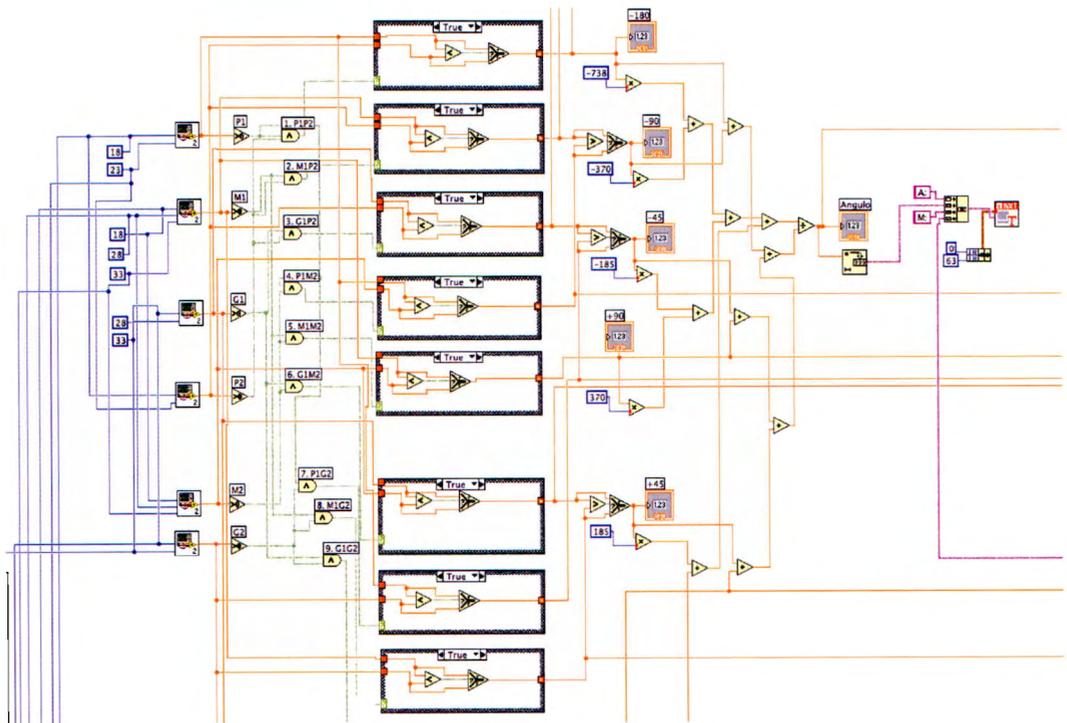


Figure 4.8 first controller of the inverse model

26

In order to determine the output of each motor with respect to a given angle we have to specify the degrees provided by the first controller. Assuming that this is the input for the second controller it is very important to compute every single movement action due to the distance with respect to measuring obstacle, otherwise a lack of possibilities would lead to a malfunction of the control when an unknown action is requested.

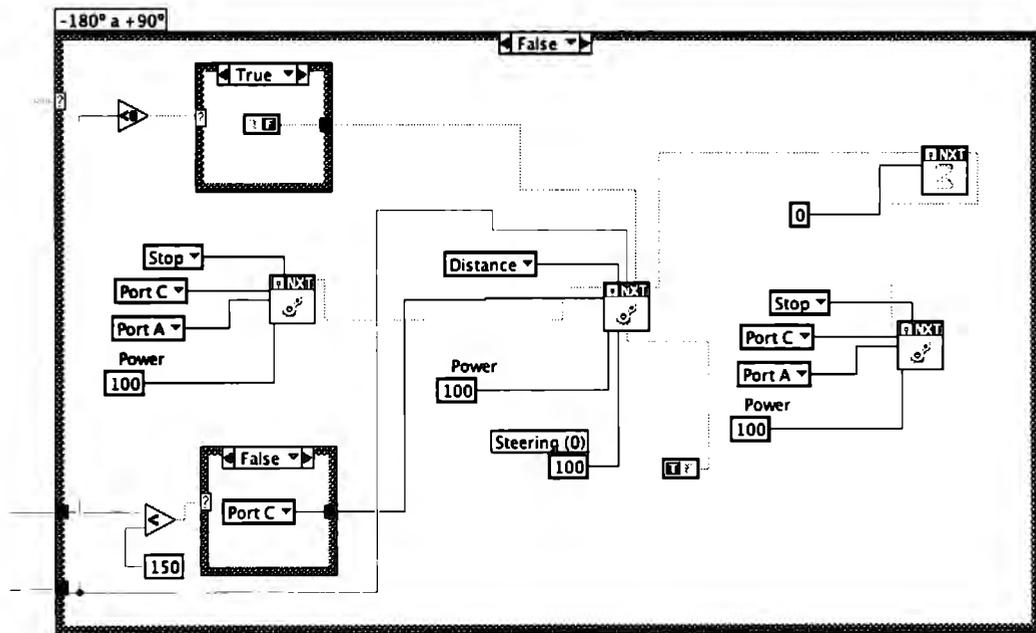


Figure 4.9 second controller to compute the final output taking the first controller as the input



Figure 4.10 robot in action with the inverse model, where there is a similar behaviour of the velocity control, both with good performance

4.3 Instrumentation

As one of the most important aspects for the project was to evaluate and prove that in some way we were creating some kind of intelligence, the need of something to quantify the learning process was vital. We decided to implement a clock that count the time that a robot takes to get from one end to the other end of the labyrinth. The next figure is the code of the clock done in Labview and the implementation, in other words the code required to display text on the brick's screen the time.

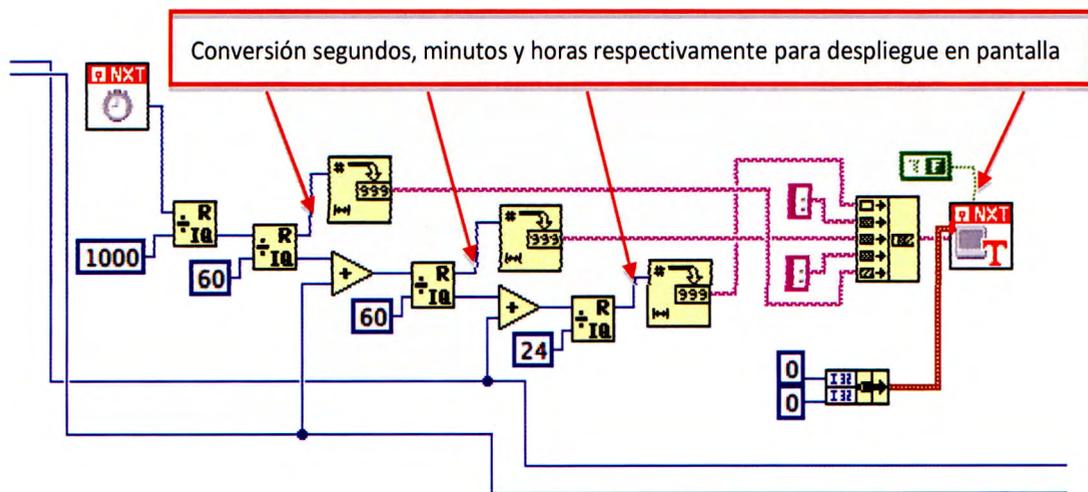


Figure 4.11 Labview clock implementation's code



'A neural network can be defined as a model of reasoning based on the human brain.'²⁷

Let say how the human brain works first. The basic nerve cell is the neuron, and has multiple connections to other neurons in order to create a densely interconnected structure or network. The process in the net is made by an electrochemical reaction which conducts the impulses to propagate the information and make a specific task. The whole process is made in parallel and is highly complex and nonlinear. Each neuron computes information independently of the other neurons but its inputs could be outputs form another neuron. The structure of a neuron is composed by the soma which is the cell body, the dendrites, a number of fibres and the axon, a long fibre. The pad of the electrical pulse, the action potential, is sent down through the axon, and it reaches synapses.

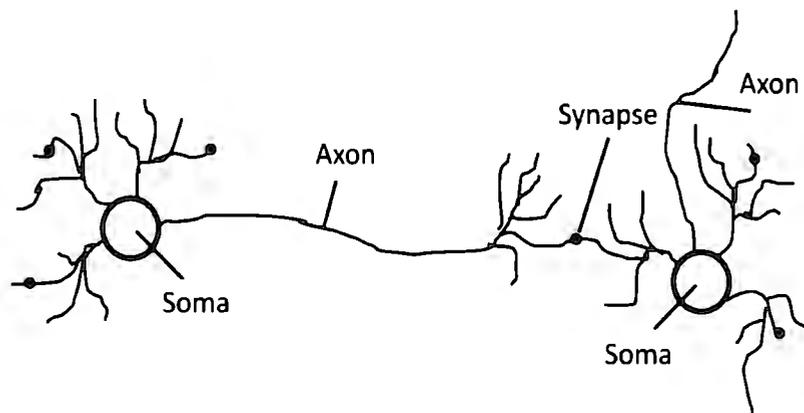


Figure 5.0 biological neuron diagram

The analogy between the biological neural network and the artificial neural network is:

Biological neural network	Artificial neural network
Soma	Neuron
Dendrite	Input
Axon	Output
Synapse	Weight ²⁸

Table 5.0 Analogy between biological neural network and artificial neural network

Artificial neural networks are good for nonlinear problems and those which we may know the inputs and outputs rather than the function.

²⁷ Artificial Intelligence, Netnevitsky, England, 2005

²⁸ Artificial Intelligence, Netnevitsky, England, 2005

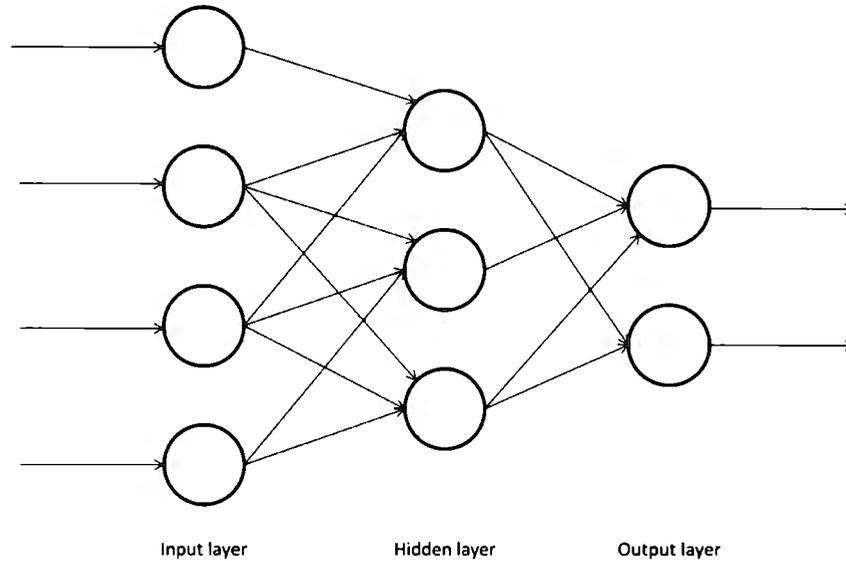


Figure 5.1 artificial neural network scheme²⁹

An artificial neural network is analogous to the biological but less complex. It has processing units called neurons and their connections are weighted to recreate the strength of the links; these are the basic means of long-term memory. Each neuron has a function to compute the information called the activation function. The neurons are hierarchy layered and each layer has one or several neurons. One of the characteristics that make powerful the artificial neural networks is the plasticity, changes in the strength of their connections that makes the right answer to be strengthened and wrong answers to be weakened.

There are several activation functions for different problems, but most common are step function, sign function, sigmoid function and linear function.

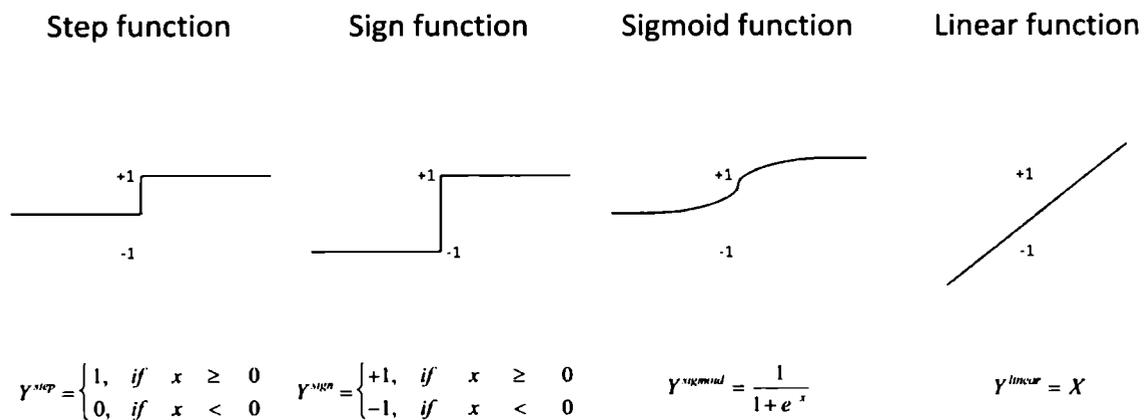


Figure 5.2 activation functions used in AI³⁰

²⁹ Artificial Intelligence, Netnevitsky, England, 2005

A single neuron could have several inputs but one single output, which spreads out to many neurons. It computes the output from the sum of the weighted inputs evaluated with the activation function.

$$f = f\left(\sum_{i=1}^n x_i \omega_i\right)$$

The learning process of a neural network could be *supervised* or *unsupervised*. The training process is the process to find the correct weights which links the interconnections.

5.1 Perceptron

Is the simplest structure of a neural network, in fact it is just one neuron. By definition it has two inputs, and is capable to classify elements that are linearly separable. It could have a bias in order to adjust the line, in case it does not cross the zero.

The process to train the perceptron consists of four steps and the training data should be applied:

5. Initialization: weights and bias are set randomly within a certain range.
6. Activation: with the training inputs and weights compute the output.

$$f = f\left(\sum_{i=1}^n x_i \omega_i\right)$$

7. Weight training: update the weights

$$\omega(k+1) = \omega(k) + \alpha x(k) * e(k)$$

Where α is the learning rate, and $e(k)$ is the error given by:

$$e(k) = [V_d(k) - f(k)]$$

Where $V_d(k)$ is the desired value.

8. Iteration: go back to step 2 until convergence.

³⁰ Artificial Intelligence, Netnevitsky, England, 2005

One useful tip consist on taking care of the values of the initial weights to avoid falling into the saturation zone, this means to determine the max and min proper values for them. Also the inputs could be normalized.

5.2 Multilayer neural networks

Multilayer network can perform more complex task rather than just classifying. It consist of one input layer, which is normally the input itself not a neuron, at least one hidden layer, the one that propagates the signals; and the output layer which leads to the output of the entire network. The simplest network with one hidden layer can perform continuous functions, with two hidden layers even discontinuous are possible to represent.

The learning process in a multilayer network is basically the same way in a perceptron, but in this case every neuron contributes with its own weight to propagate the information. All the weights are adjusted to reduce the error when the training pattern is applied like in a perceptron is done. There are many algorithms to train a multilayer network, but the most popular method is back-propagation.

In this method the error signal is propagated backward to previous layers, starting from the output. In this case an error gradient is produced to adjust weights from all layer behind the output which is the only that has directly the proportion of error produced due to certain data training. Then a sensibility is generated to propagate the information about the error, the gradient. The steps for the method are similar to the perceptron:

5. Initialisation: set randomly weights and bias.
6. Activation: calculate outputs.
7. Weight training: this step is different due to complexity of the new error gradient and equations, but in practical terms is the same. The whole procedure is shown below.
8. Iteration: go to step 2 and stop until convergence.

The next diagram is a general artificial neural network to understand the back-propagation algorithm.

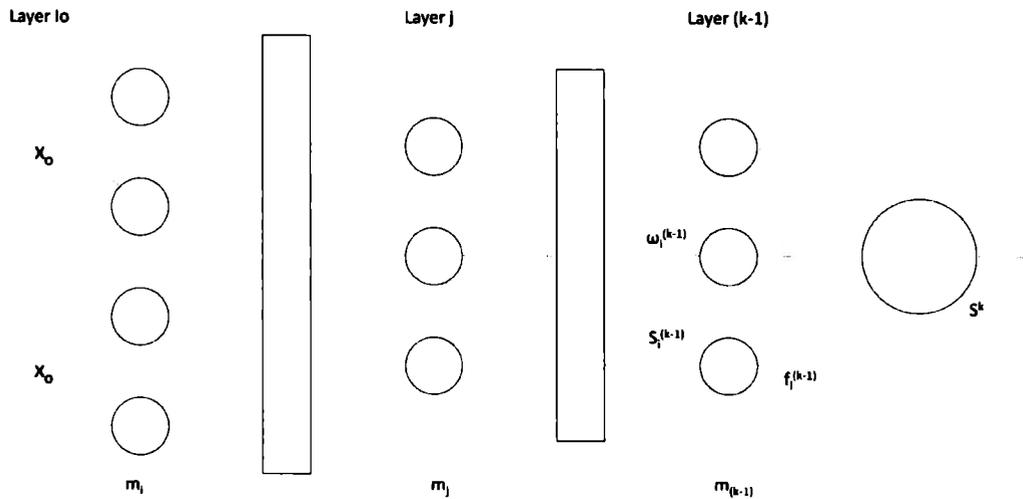


Figure 5.3 diagram of the back-propagation

Where:

$j = \text{layer}$

$i = \text{element}$

$$\frac{\partial \zeta}{\partial w_i^{(j)}} = \left[\frac{\partial \zeta}{\partial w_i^{(j)}} \dots \frac{\partial \zeta}{\partial w_{j-1}^{(j)}} \right]$$

$$\frac{\partial \zeta}{\partial w_i^{(j)}} = \frac{\partial \zeta}{\partial s_i^{(j)}} \frac{\partial s_i^{(j)}}{\partial w_i^{(j)}}$$

$$s_i^{(j)} = x^{(j-1)} w_i^{(j)}$$

$$\frac{\partial s_i^{(j)}}{\partial w_i^{(j)}} = x^{(j-1)}$$

Substituting

$$\frac{\partial \zeta}{\partial w_i^{(j)}} = \frac{\partial \zeta}{\partial s_i^{(j)}} x^{(j-1)}$$

Recalling:

$$\frac{\partial \zeta}{\partial s_i^{(j)}} = \frac{\partial (d-f)^2}{\partial s_i^{(j)}} = -2(d-f) \frac{\partial f}{\partial s_i^{(j)}}$$

$$\frac{\partial \zeta}{\partial w_i^{(j)}} = -2(d-f) \frac{\partial f}{\partial s_i^{(j)}} x^{(j-1)}$$

If δ is the sensibility:

$$\delta = (d - f) \frac{\partial f}{\partial \mathfrak{a}_i^{(j)}}$$

$$\frac{\partial \mathfrak{z}}{\partial \omega_i^{(j)}} = -2\delta_i^{(j)} x^{(j-1)}$$

$$\therefore$$

$$\omega_{i_{new}}^{(j)} = \omega_i^{(j)} + c_i^{(j)} \delta_i^{(j)} x^{(j-1)}$$

Where c is the learning factor

The last layer (k) weight calculation

$$\delta^k = (d - f) \frac{\partial f}{\partial \mathfrak{a}^{(k)}}$$

For sigmoid function

$$\frac{\partial f}{\partial S^k} = f(1-f) \therefore \delta^{(k)} = (d-f)f(1-f)$$

$$\omega_{new}^k = \omega^k + c^k (d-f)f(1-f)x^{(k-1)}$$

For hidden layer:

$$S_i^{(j)} = (d-f) \left[\frac{\partial f}{\partial \mathfrak{a}_i^{(j+1)}} \frac{\mathfrak{a}_i^{(j+1)}}{\mathfrak{a}_i^{(j)}} + \dots + \frac{\partial f}{\partial \mathfrak{a}_{m_{j+1}}^{(j+1)}} \frac{\mathfrak{a}_{m_{j+1}}^{(j+1)}}{\mathfrak{a}_i^{(j)}} \right]$$

$$= \sum_{j=1}^{m_{j+1}} (d-f) \frac{\partial f}{\partial \mathfrak{a}_i^{(j+1)}} \frac{\mathfrak{a}_i^{(j+1)}}{\mathfrak{a}_i^{(j)}} = \sum_{j=1}^{m_{j+1}} \delta_i^{(j+1)} \frac{\mathfrak{a}_i^{(j+1)}}{\mathfrak{a}_i^{(j)}}$$

Weight do not depend on S

$$\frac{\mathfrak{a}_i^{(j+1)}}{\mathfrak{a}_i^{(j)}} = \sum_{i=1}^{m_{j+1}} f_i \omega_i^{(j+1)}$$

Where:

$$S_i^{(j+1)} = x^j \omega_i^{(j+1)}$$

$$\frac{\mathfrak{a}_i^{(j+1)}}{\mathfrak{a}_i^{(j)}} = \frac{\partial \left[\sum_{i=1}^{m_{j+1}} f_i^{(j)} \omega_i^{(j+1)} \right]}{\mathfrak{a}_i^{(j)}}$$

If

$$\frac{\partial f_i^{(j)}}{\partial \delta_i^{(j)}} = f_i^{(j)} (1 - f_i^{(j)})$$

Where:

$$S_i^{(j)} = f_i^{(j)} (1 - f_i^{(j)}) \sum_{i=1}^{m_{j+1}} \delta_i^{(j+1)} \omega_i^{(j+1)}$$

And $\delta_i^{(j+1)}$ transmits the information

Therefore it is independent of the error ending up with the "Generic rule" as:

$$\boxed{\omega_{i_{new}}^{(j)} = \omega_i^{(j)} + c_i^{(j)} \delta_i^{(j)} x^{(j+1)}}$$

For the learning technique previously proposed in past chapters we implemented an artificial neural network in Lego. This was a simple three layer neural network trained with back-propagation algorithm. The neural network was trained before the direct implementation and was programmed with the adjusted weights after iterations end up with the final values till convergence. Even though the network on the robots has fixed weights, the whole algorithm was programmed to make it easier the algorithm to converge and for debugging purposes, so changes could be done or other training patterns could be applied to the neural network.

The objective of the neural network was to classify objects according to their colour. Therefore a light sensor was used to determine whether an object was dark or clear depending on the reflection produced by its surface. The input was used as one or zero value making a comparison with given values. The range for dark object was from x to x and for clear ones was from x to x . This value was multiplied times k in order to obtain a number adequate to compare and use with Boolean logic.

The topology used for the neural network was the next:

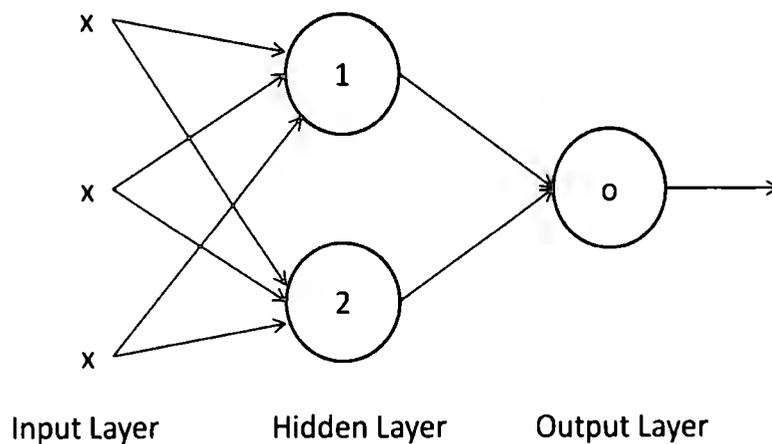


Figure 6.0 Topology of the artificial neural network used on Lego®

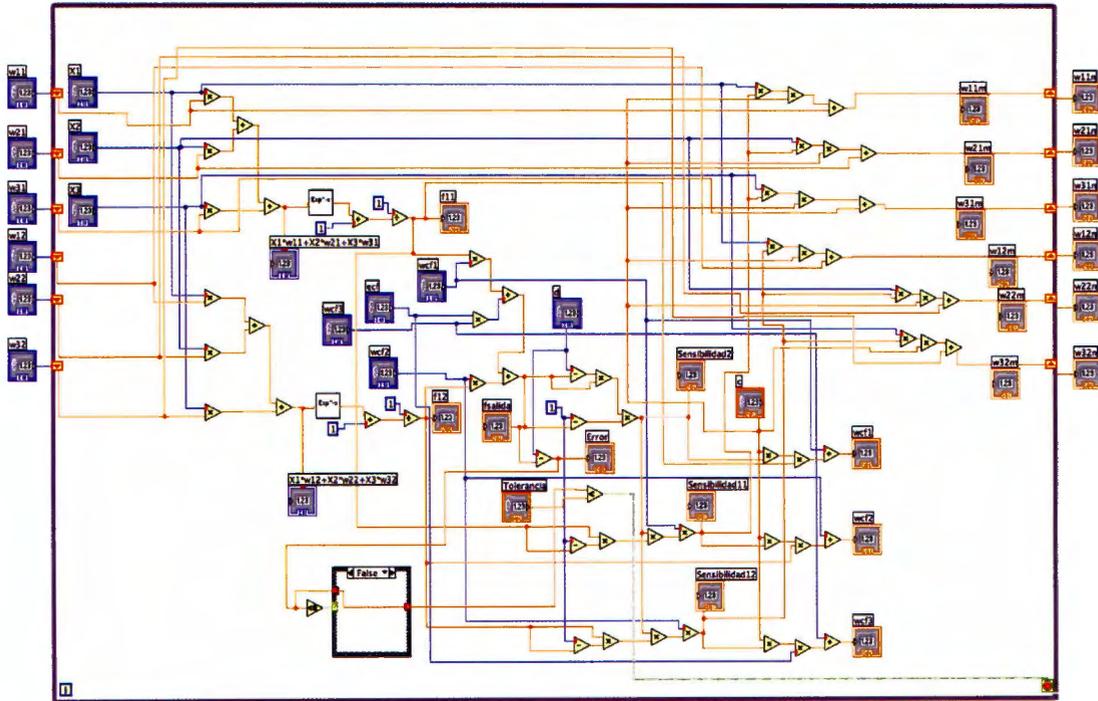


Figure 6.1 Labview code for the neural network topology used with the back-propagation algorithm

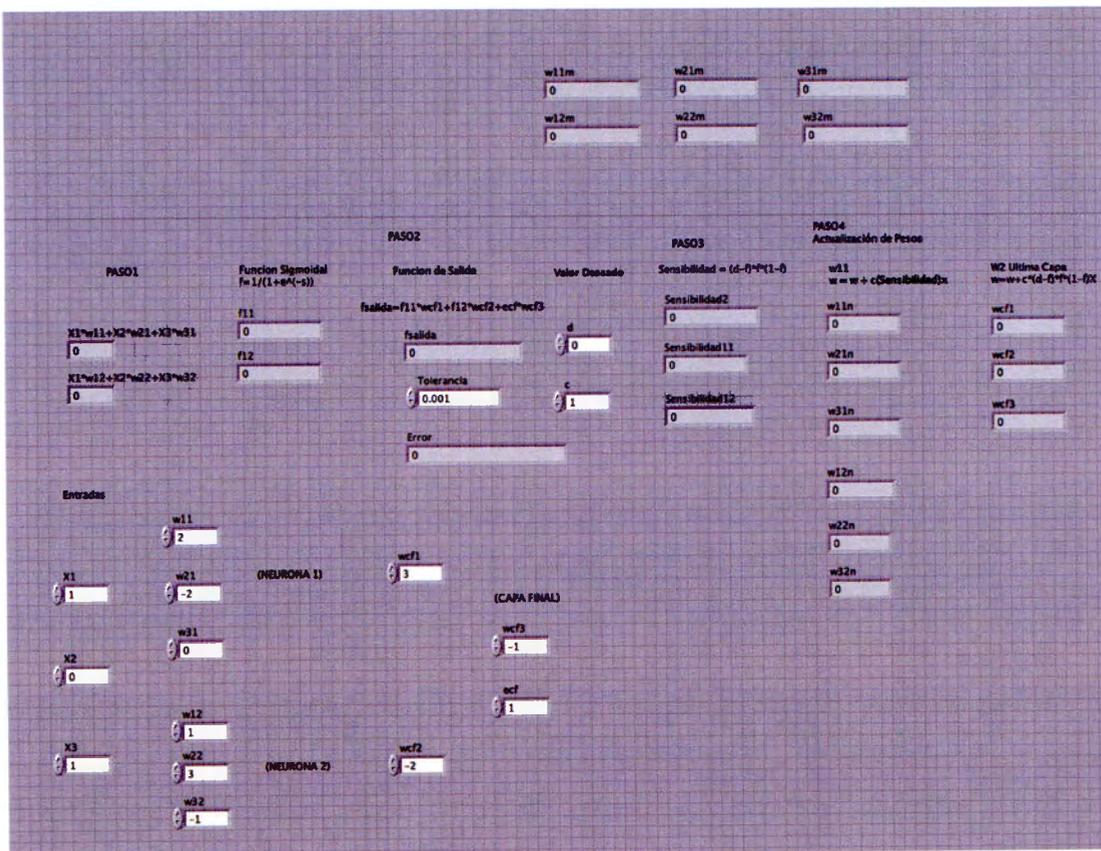


Figure 6.2 Front panel of the ANN code, parameters are evaluated and displayed during debugging



The “swarm” intelligence’s roots come from modelling the knowledge due to outcome gotten from multiple interactions with the environment emphasizing the importance of different options that exist due to changes in parameters within a system. This intelligence is strongly linked with the biological studies of behaviours which are auto organized in a social environment that implies concepts like communication, movement, regrouping, autonomy’s unit and coordinated efforts related to the whole group focused in getting one single target.

The first models are based on animal’s and insect’s behaviour living in their own environment to hunt, survive or solving one specific problem. As an example, the ants utilize territorial disputes against other species acting in a coordinate way, splitting and attacking, then regrouping quickly again.

It has been used in the militia³¹ where exist a need for convergence of attacks against enemy forces in a coordinate way (avoiding friendly fire) applying multiple flanks and different attack directions, allowing defeat the enemy even if the equipment and the forces are lower than the rival. In telecommunication has many applications, e.g., for information interchange, net traffic and protocol designs as the BitTorrent³² protocol designed to interchange files among peers of the same condition within the net (that’s why it is called “peer to peer” or “P2P”) where there is no task distribution between them, just a communication between nodes with a client/server relation.

This algorithm is modeled with principles of “swarm”, where the interchange is realized not necessarily starting from the beginning of the file, but it is divided at random into hundreds of small pieces, as well as statistics about the number of transfers, nodes with copy file to be interchanged and nodes that keep only one portion of it.

31

32

The efforts of the whole community are integrated to supply information that support the best solution about the route to go, speed of swarm's transfer, file portion to be interchanged and the best node to be sent.

This technique has demonstrated being effective in many applications of autonomous control of individual agents, because surprisingly the complexity of these behaviors and its collective structures do not reflect the simplicity related to individual behavior of each of its members. The advantage that offers this method is its nature itself where the complexity in cognitive terms of skills of each agent can be higher in absolute sense, but its individual behavior is simple and easy to process.

In many cases an agent by itself could not find an efficient solution face to global problem, but as a group and with the characteristic as "a whole", then the solution to the problem can be found easily.

The Particle Swarm Optimization Algorithm (PSO) was first introduced by James Kennedy and Russel C. Eberhart in 1995³³ and proposes that the influence and the social learning allow persons to keep cognitive consistency about their actions and explain how this consistency generates knowledge and intelligence. Persons solve their problems talking about them with other persons; as they interact with beliefs, attitudes and social behaviors, they generate solutions to their problems that change the way of thinking and act in the future for similar situations.

These changes typically can be represented as individuals moving in a space socio cognitive. The Particle Swarm Optimization Algorithm simulates this type of social optimization.

This type of optimizations has been extended to systems of intelligent control due to its capability of adaptation and learning in variable environments, allowing flexible systems to support changes.

33

As this work is about artificial intelligence, we combined the optimization algorithm with fuzzy logic design. It is well known that one of the main problems of fuzzy controllers design is the way to determine precise and complete set of rules “*if_ then*” and the shape and number of membership functions³⁴. That is why and in order to overcome these problems it is proposed to optimize the membership value functions of the fuzzy controller with the Particle Swarm Optimization Algorithm. The formal analysis of this algorithm is shown with each step and its application in the system of artificial intelligence proposed:

- a) **A problem is set.** Set the Robot’s intelligent behavior. Optimization of the robot’s behavior to solve labyrinths and avoid obstacles (membership functions optimization).
- b) **It is defined how it will be solved, setting one function (fitness function) that models the problem.** Some fitness functions are used underlining the following: metric distances between obstacles, estimation by value error and time ran in completing the task.
- c) **The social communication structure and the net are defined.** Neighbors are assigned that can interact each other as individual agents.
- d) **The community is initialized.** Input Readings from different sensors are known now as “particles”. They are defined as random elements for problem solving because these individual agents are candidates to be a solution of the problem.
- e) **An iterative process is set to improve candidates.** The particles acting in an iterative loop assess the “fitness” to find possible solutions and remember where they got the best success. The best individual solution is called “the best local”.
- f) **Each particle shares this information to its neighbors.** They also determine where their other neighbors have had the best success.
- g) Finally there are numerous movements through the search space within the possible “best locals” guided by the different “global success” of the community which normally converges at the end of this process.

It is important to recall that the “particles” have three capabilities of essential reasoning:

- 1) It remembers its best current local position (individual and its own itself).
- 2) The knowledge of the best globals from individual perspective of its neighbors and the best global up to the moment from the whole community perspective.
- 3) Within the community, members communicate and advertise the best positions of each other and adjust their own speed and position based on these criteria.

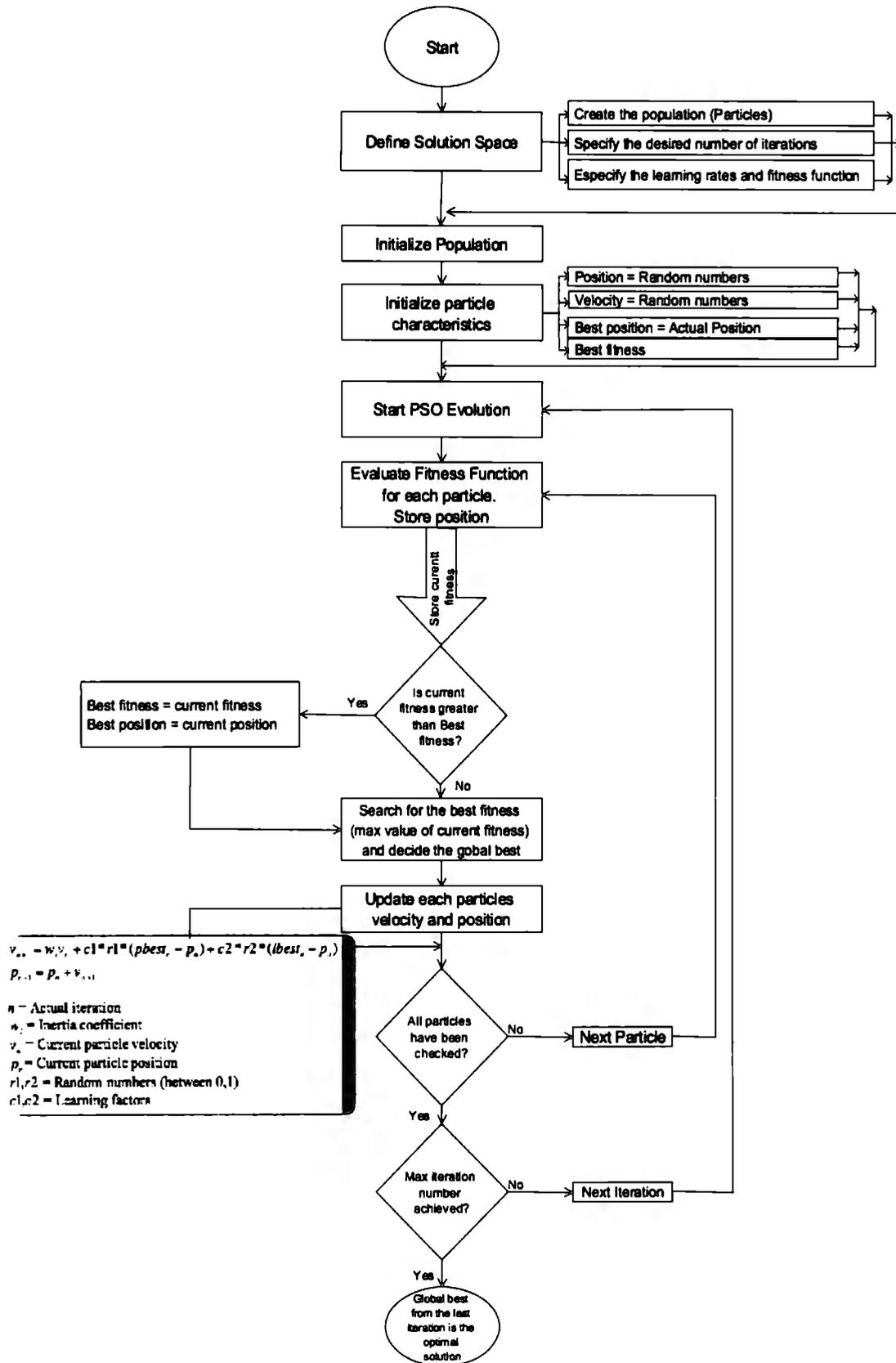


Figure 7.0 Particle Swarm Optimization algorithm flow diagram and equation



As we mentioned in previous chapters, artificial intelligence is achieved with different techniques in order to create “intelligence” in a machine. For doing so, we proposed the controllers optimization to be done with the particle swarm algorithm so a learning capability could be obtained during the learning pattern communication.

Despite of the expert knowledge applied on the system, there always the need to improve the behaviour of our design, and determining the perfect set and number of “*if_then*” rules is one of the main problems when fuzzy logic control is used, as well as the shape for the membership functions. The PSO is directly implemented to optimise the parameters for the membership functions.

The code below shows the algorithm implementation on lego with Labview. First we can observe the initialization of the community with a certain value. We specify this number with respect on how many individuals will provide a possible solution in the search space. We also define the fitness function that would model the problem and the social structure.

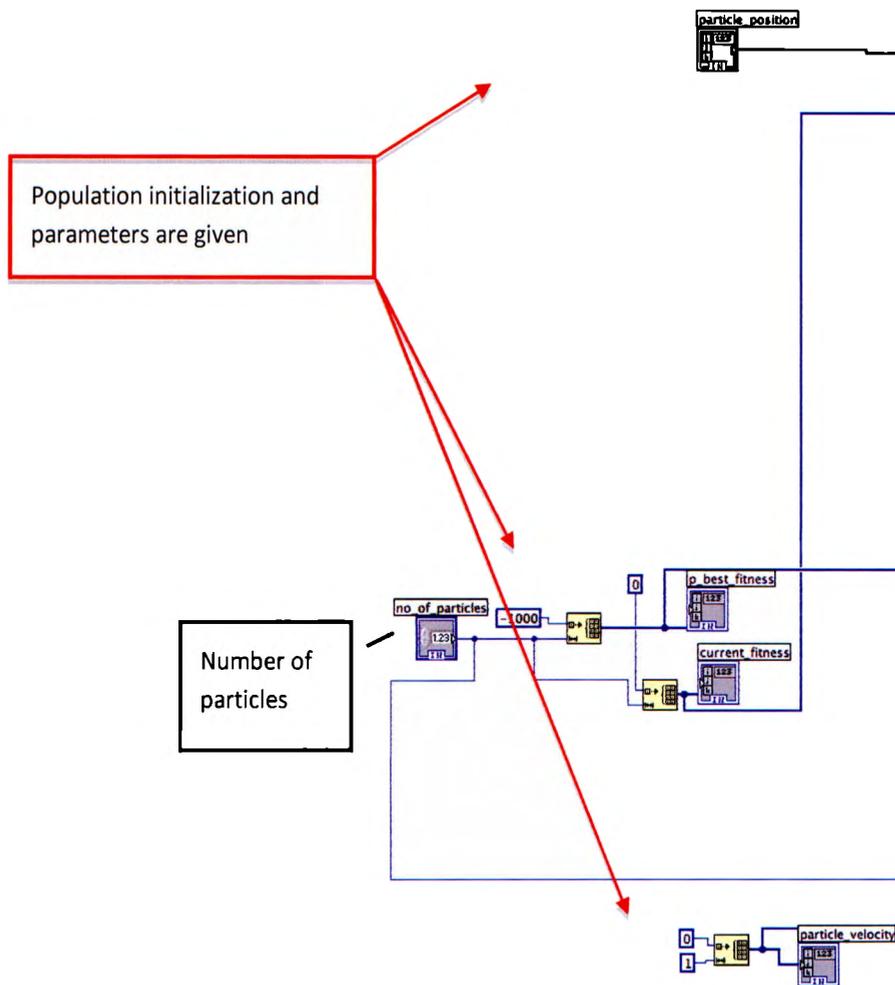


Figure 8.0 Labview code. Initialization of the algorithm, population and parameters.

The next code shown is the main process of the algorithm which was described in the previous chapter:

- a) **A problem is set.**
- b) **It is defined how it will be solved, setting one function (fitness function) that models the problem.**
- c) **The social communication structure and the net are defined.**
- d) **The community is initialized.**
- e) **An iterative process is set to improve candidates.**

- f) Each particle shares this information to its neighbors.
- g) Finally there are numerous movements through the search space within the possible “best locals” guided by the different “global success” of the community which normally converges at the end of this process.

It is important to recall the three capabilities of essential reasoning of each particle.

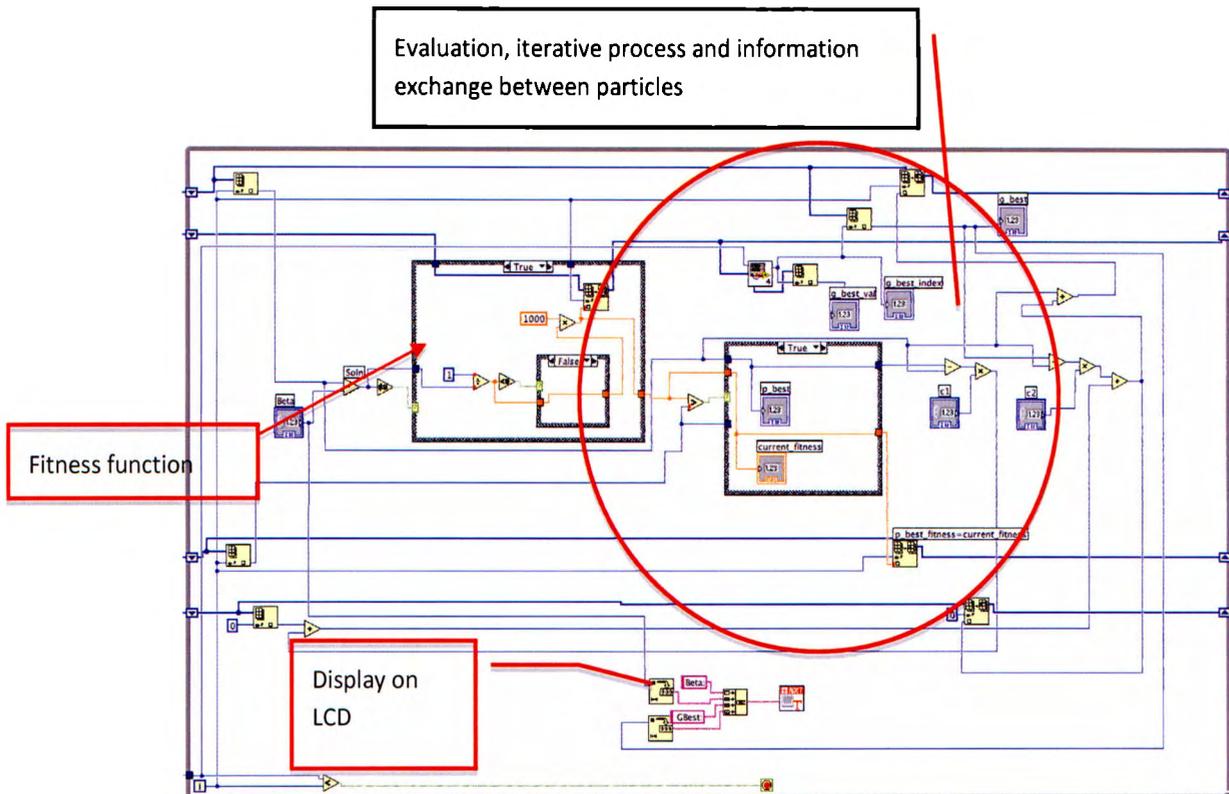


Figure 8.1 Labview code. PSO algorithm main structure

Last figure shows the final product of the algorithm iterative process when the convergence was achieved for the best possible solution. This is basically the end of the first run for the algorithm to obtain the optimization for the membership function values.

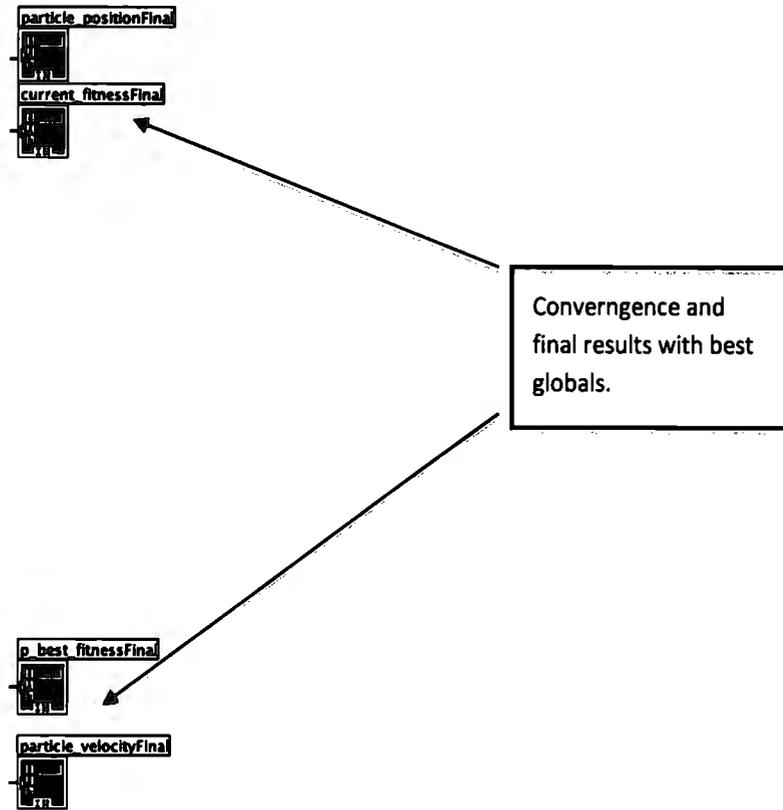


Figure 8.2 data driven after the convergence of the iteration process



As we mention on previous chapters one of the most important things for the community is to maintain communication between members to allow social learning to keep cognitive consistency about their actions and generate knowledge and intelligence. With this premise the community will work together and none won't be isolated but took as part of the population development.

To achieve this vital aspect, all members include a communications protocol to establish bidirectional exchange of information and learning patterns. The protocol used is the Bluetooth protocol because is the one which is implemented in the Lego brick.

Previous to the programming for each member, there is one step that must be done in order to communicate one member with the rest and vice versa, pairing the devices. This step is important so that every member knows the rest of the community and this is the way they will be introduced:

A connection must be established between members, accomplished from one device to another. The first will be considered as master and the second as slave. It is important to recall that this hierarchy is meaningless when transferring information is processed, as bidirectional communication is made. This could be interpreted as the one who knows other pals but they do not necessarily know each other as between humans one must be introduced before knowing about other's existence. We must remember that every device will now have a contact list but every connection must be reestablish

The only constraint found in the system is that just four members will be related to one certain *subgroup* of the community, this means that only four devices could be paired. But this limitation is overcome with a simple strategy presented below:

One master could have its own three slaves, but if we consider that one slave could be the master for other three slaves, we just rest the communication between the two of them and took one channel for each, we obtain a six-group instead of the four-group mentioned.

The transferring process takes place when the button transfer is pressed in one robot, then we could say that the learning patterns are send. This stage was done taking the consideration that we are training the community to play the collaborative task, so we promote the communication between robots.

Something that has to be remembered is that anything that would be sent through the Lego system should be converted into a string type data. So if we want to use this data we also might convert the data again into a number type so other use than displaying text could be done.

To send and receive and messages, each connection has ten mailboxes, and the text string could be up to fifty eight characters long. To distinguish which mailbox will used which variable, one has to specify the number in the receiver and in the sender.

The code below represents the communication programming for the learning pattern transfer. We use this process to send parameters of the membership functions, the values for alpha, beta and gamma points of the universe of discourse. These particular variables are used to apply the expert knowledge and set the best values for the functions. We might remember that we first run the particle swarm optimization algorithm in one robot to find the best global value so the artificial intelligence could be achieved during the practice done by the robot. After that the knowledge could be transfer so every individual could posses it.

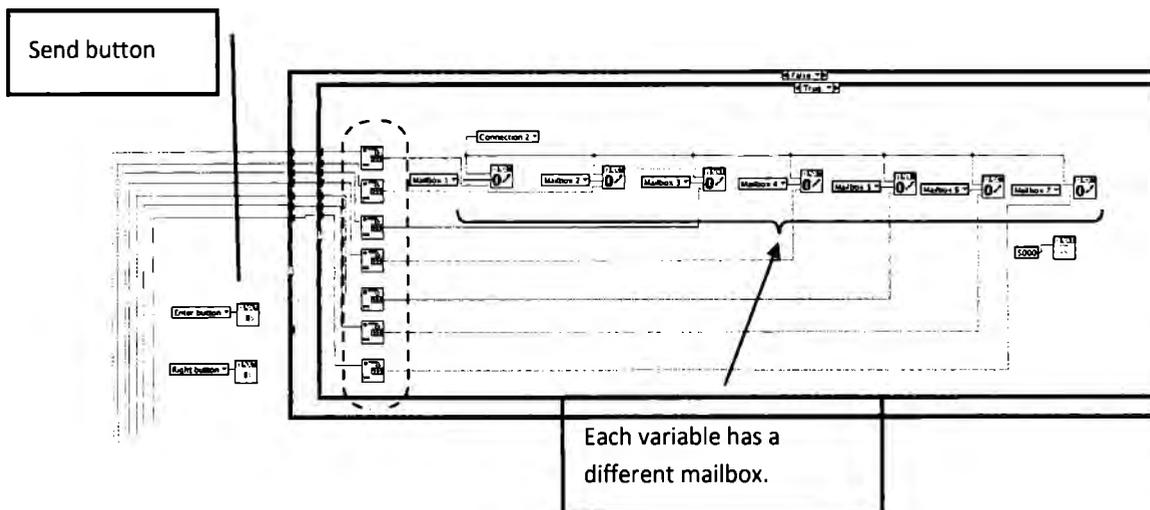


Figure 9.0 Labview code for the Bluetooth communication process

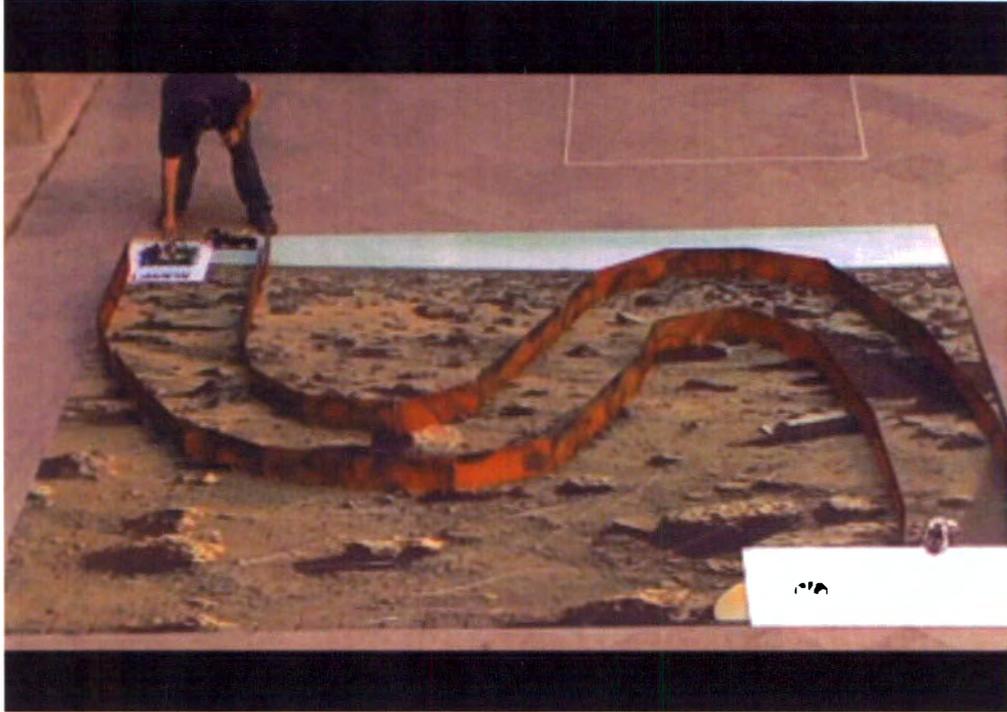


Figure 9.1 Bluetooth communication test. The first run is to optimize the route

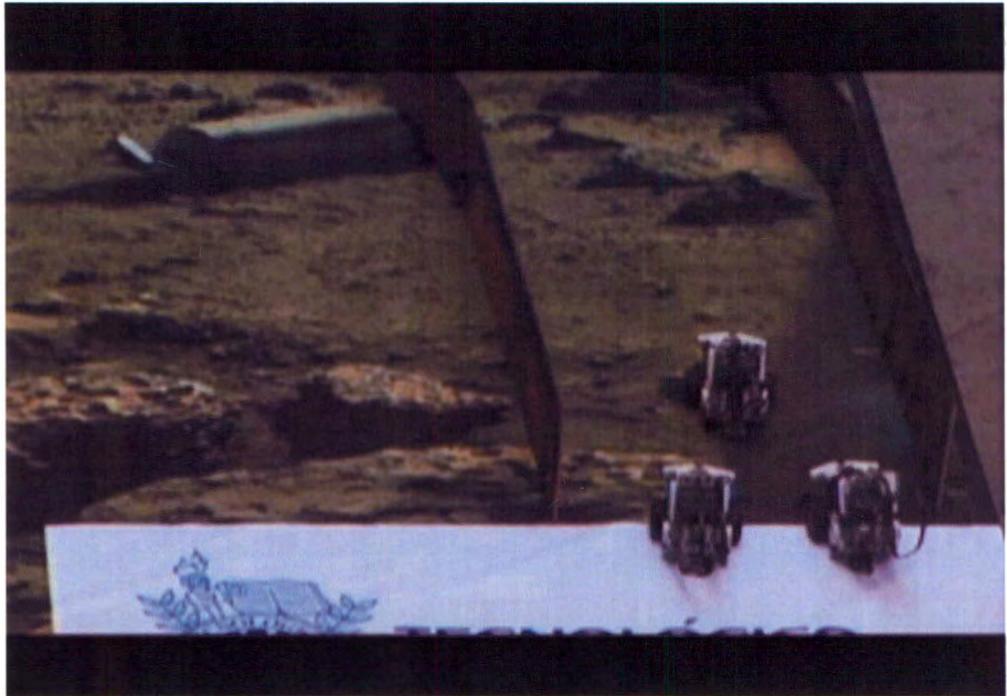


Figure 9.2 the three robots are ready to go



Figure 9.3 the first robot has finished the route and transfer the learning patterns to the other two



Figure 9.4 the second robot reaches the goal with the patterns learned during the communication

9.1 Bluetooth communication stages

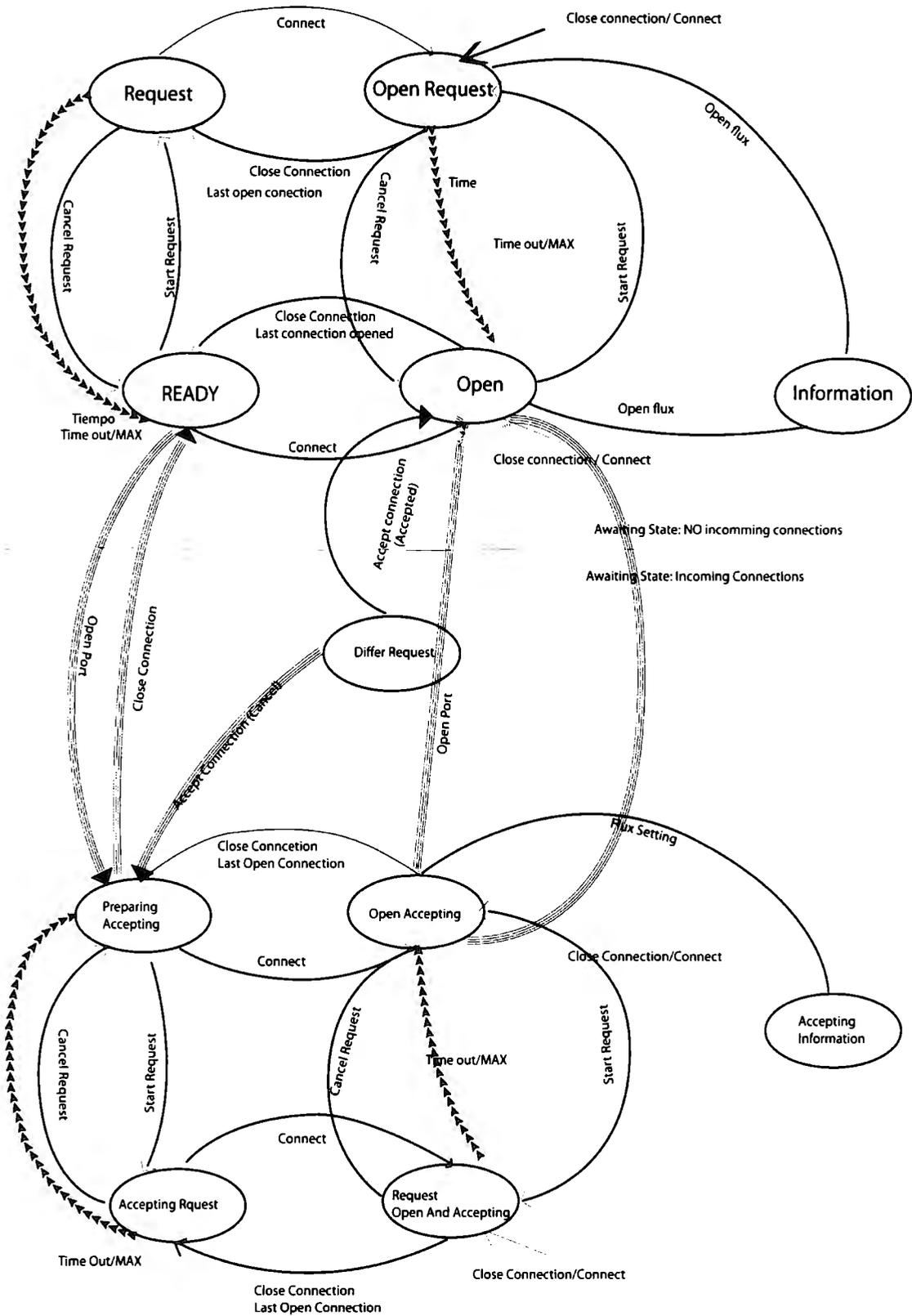


Figure 9.5 Bluetooth communication stages diagram

10.1 PBL. Problem Based Learning methodology

Problem based learning (which has its roots in the School Medicine of the University of McMaster, Canada) has been adopted successfully by different prestige universities throughout the world; it presents an innovative educational proposal where the knowledge is significant and centred in the student, who develops abilities and reaches greater cognitive levels through little work groups that learn in a collaborative way in the search of a resolution to an initial complex and challenging problem. The objective is to trigger self-directed-deductive learning focusing the attention neither in the content nor in the memorization of the information or facts but prioritizes concrete examples and the applications about abstract concepts⁹.

PBL proposes the need that the student develops new skills like the solution of problems, the search for information and critical thinking, meaning that the student has to learn to learn. The knowledge has to be relevant, in such a way that it can be applied to daily life, making emphasis in developing the capacity to work in team. It is important that it should be based on philosophical, psychological, pedagogical and psychometric contexts.¹

Psychological for the knowledge to be active, developing persons inside, favouring cooperation. Philosophical to generate a multicultural knowledge, liberating and critical so it breaks the traditional scheme. Pedagogical to instruct in a diverse applied form to smaller groups. Psychometric, where the grade has an evolution based in parameters to appreciate the quality of the course, like a continuous tool that supports knowledge.

PBL seeks to apply innovative strategies with valid elements of a traditional model to improve and incorporate techniques that enrich the student's learning, directing to a constructive model. Emphasizing on problem based learning, motivating the research of possible answers, new information, looking for more global goals and encouraging abilities of investigation, emphasizing on team work and not individual work.

For this, it is intended to design a board of strategies that fundament the search and propose a solution, based in the methodology that brings certainty to the resolution of the problem.

It is necessary to construct a mental map of the course that is a graphic document which contents are the main concepts and its ideas, ordered in a hierarchical way, showing the interdependence and the associations they present among themselves.

It is also imperative to create a tutorial guide, which presents the general objective of the activity, the student's previous knowledge, problem definition, terms classification, the learning objectives, the topic or topics related to the activity, difficulties, necessary logistic and estimated time for each activity.²

Problem based learning presents the following fundamental characteristics³:

- Student centred learning.
- Knowledge is generated in small groups of students.
- The problems form the focus of the organization and the stimulus for learning.
- The problems are a vehicle for the development of resolution skills.
- New information is acquired through self-directed learning.

The methodology for PBL has the following series of steps shown in figure 10.0

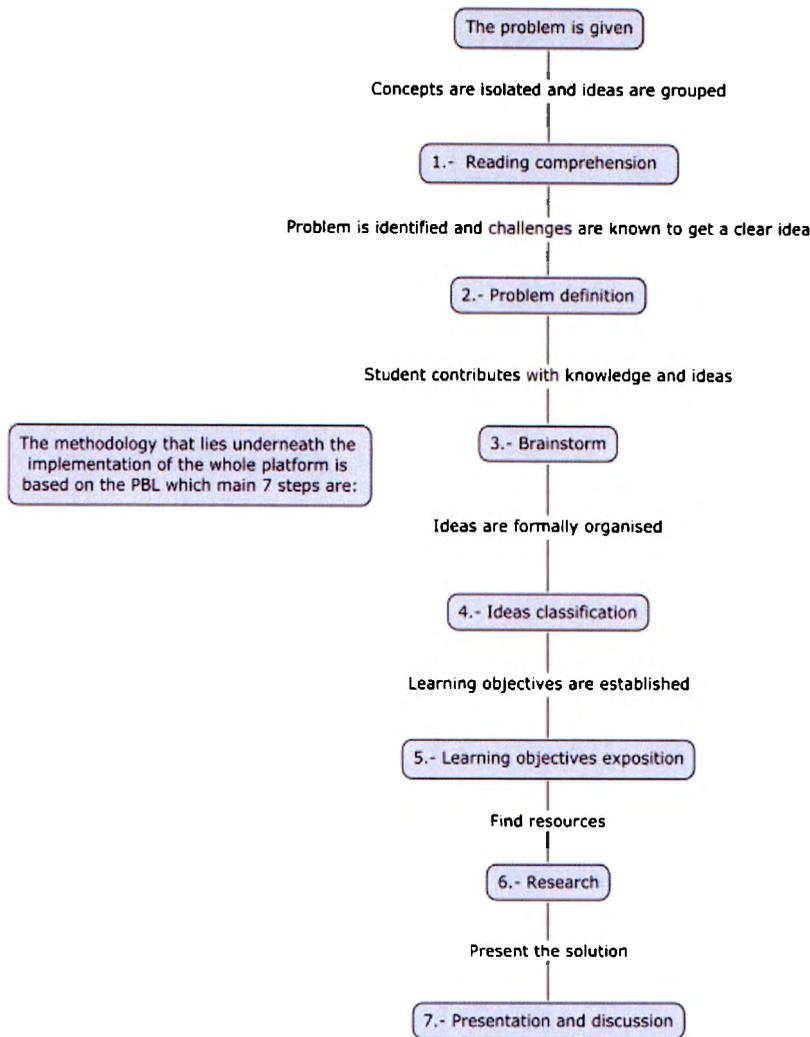


Figure 10.0 7 steps of the PBL

Incorporating the characteristics of problem based leaning in our system of educational robotics, we propose a new model of teaching in which each robot instructs, with concrete examples, different tasks to one or several students, themes and topics that relate among themselves. With this premise the comprehension and generation of knowledge of the concepts that are intended to instruct is more clear and concise. Due to the use of conventional control laws, intelligent control laws, artificial intelligence, neural networks and algorithms of optimization to obtain a certain desired behaviour in each of the members of the robotic community, this topics are proposed as a initial basic system to instruct. Besides these central subjects, a series of key examples are approached so they provide the student the most number of concepts when the implementation comes, some of them involve:

- The implementation of proportional controllers (P), proportional derivative (PD), proportional integral (PI) and proportional integral-derivative in a robot.
- The implementation of a robot that follows sound.
- The implementation of a robot that applies the concepts of the inverted pendulum.

It is clearly shown that while a natural evolution exists over the desire of the student to obtain a better behaviour from the robot he is designing, his knowledge is totally self-directed and independent.

10.2 Development

Once understood the methodology of PBL, below an example of the PBL methodology is given regarding the previous explanation.

- 1) Local and individual tasks are assigned to each robot member of a little community. The purpose of each of them is:
 - a) The solution of the task.
 - b) The transmission of the experience and learning patterns, generated due to the solution achieved, to all the members of the community.
 - c) The acquisition of knowledge received by other members.
 - d) Updating the way they fulfil their processes.
 - e) Await a new task.
- 2) A global task is completed with each of the contributions of the members from the community.
- 3) A hierarchical system of operation is established; where a member with superior hierarchy is assigned more complex tasks and therefore it count with a greater number of sensors.

A group of robots that interact between themselves to develop joint work are usually called collaborative robots and they present the most notable advantages in the operations in which the participation of one specialized robot is not enough, results very expensive, time-

consuming or its design is complex. Another fundamental advantage, of the use of collaborative robots, is that they present characteristics of tolerance to failures, due to the redundancy of information between the members that compose the system. A failure in the function of a robot can be corrected for the rest of the team.

Some of the members that compose this hierarchy are mentioned above (considering each "robot" as a system of: sensors, actuators, processing unit and a communication unit, which have a designated task):

a. - Leader robot: Receives all the information of the global task. Its objective is to be able to guide and train other inferior robots (with less sensors and lower cost) so that they can solve in the same way the global task. These tasks include solving successfully a labyrinth (this means without collide), or the perfection of a process in less time. The leader presents a robust architecture and a control that allows the performance of the functions that present a greater complexity such that any other member of the community could not be able to realize at the moment the task integration itself. That that reason it is provided with more extended capacities so it can adapt to the terrain and facilitate the advance of the rest in any means.

b. - Global task robot: These types of robots have more than one task assigned to develop in the work field. The work field is a place in which the assigned tasks will take place. They perform their action in a specific place following the instructions of the leader robot. They don't poses previous knowledge of the place, route nor the global task; therefore they are called "system operators" due to the fact that their performance is limited to the knowledge they acquire during the followed orders of the guide robot.

c. - Model task robot: its objective is based mostly in versatility while executing a task, such as the movement through any trajectory or given labyrinth. Specifically in this part of the learning process, terms with numerous amounts of concepts, at the moment of their implementation, are applied. It could be a robot that instructs concepts about stability and principles of the inverted pendulum. With said design, real fuzzy control could be appreciated as the complexity performance is done. This robot presents an advantage besides the global task robot. Its capacities are not limited only to follow the leader robot; it can perform by itself recovery and search tasks as it acquires enough knowledge or as it has received useful information. Finally, this robot could inclusively perform the role of the leader robot (when implemented with a neural network system

that will help generalize any problem) in the case any problem presented or its performance got interrupted due to external conditions.

d. - Specific task robot (or workstation): This series of robots are known as automats due to the fact that their tasks are previously defined, they don't present changes and depend on the operator to perform their task. They do not present intelligent control and they base their actions on traditional control.

It is observed that the interaction of a robotic community with a hierarchic scheme and with members with the previously mentioned characteristics, facilitate the exemplification of certain control tasks dynamically and very illustrative.

After the PBL understanding and the construction of the minimum system presented for the robots, the process should start describing the main scenario definition. For this case we proposed the global task for search and exploration. But previous to the concept idea, we may define the place where the work should be done. It is important to recall that the place may influence the learning process of a student.

The area where all the activities will take place is also one of the main aspects introduced by this educative methodology in order to establish some characteristics for the workplace to develop the whole bunch of activities. The idea is indented to implement a logistic for the learning process in which a systematic, well-defined and structured program is conceived to increase the potential of problem base learning to instruct some specific subject. For this reason is that we came out with this definition for the workplace, developed and validated during the entire process, which ended up with a simple all-in-one workspace capable of recreate unlimited number of scenarios to propose several examples and didactic activities for thousands of knowledge generation.

The development environment proposed consists of an area of 25 square meters, meaning a square of 5-meter-side. This space allows easy displacement of the robots and lets the activity define the configuration appropriated for its tasks, giving the possibility of an infinite number of routes to be traced with the different panels.

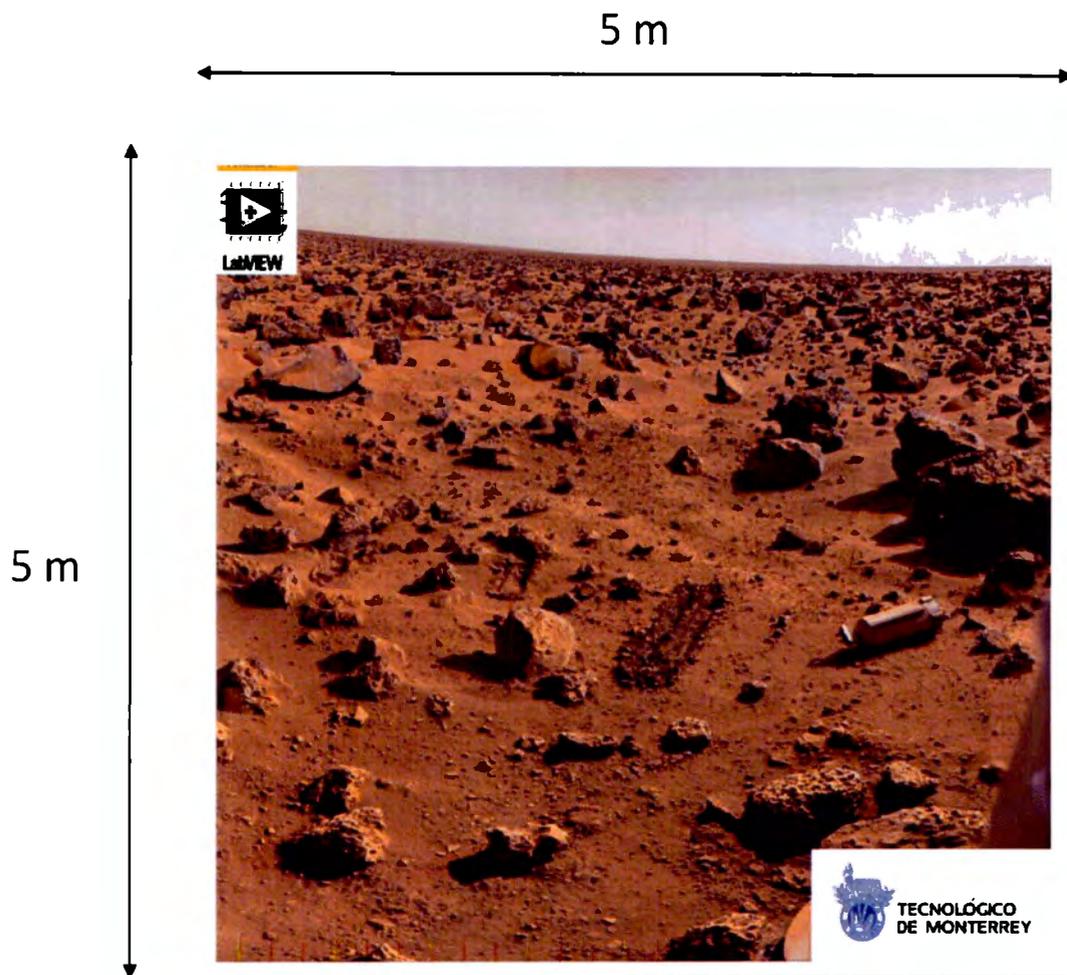


Figure 10.1 workplace for the activities

Panels for the labyrinth were established in two different sizes to have different configurations. They are versatile and let the construction be easy and quick for several arrangements. Each scenario can recreate a zone for each activity, and not only one for all of them.

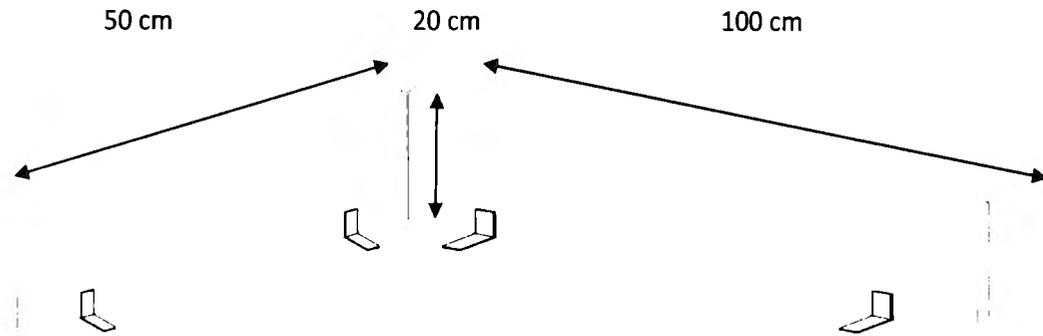


Figure 10.2 panels for the labyrinth construction

First of all we need to specify the workplace for the corresponding activity. A start zone and a work zone are established, where respectively a start point and end is denoted. The different trajectories must start and arrive from and to these two zones in order to describe a specific pad or route. Having this trace, we could apply qualitative analysis of the learning of the members of the community in different environments and possibilities.



Figure 10.3 start and work zones designated for the activities in the work place

When implementing this basic system, the individual is capable of learning in a practical way engineering knowledge, as an example of this learning platform should be fuzzy logic, neural networks, optimization systems, conventional controllers (P, PD, PI, and PID), systems and

schemes of communication between robots, some protocols and a programming language. During the experimental activities, tangible knowledge could be achieved, with a better understood and reasoning.

In an individual robot, a basic sensor system must be implemented to provide abilities that allow the robot to interpret the environment. To understand which sensor is functional for a given member of the community, it is important to analyse the circumstances and challenges that everyone will face during their task contribution. Therefore is vital to process signals acquired from the environment through sensors to react to outside stimulus. The basic system proposed is presented below:

- a) Ultrasonic sensor, capable of determining the distance that exists between the robots and the obstacles (preferably from 0 to 255 millimetres with a precision of ± 20 mm). Also to help the detection of dynamic obstacles.
- b) Touch sensor, to alert the system of a physical stimulus in order to determine an intelligent reaction when pressed, released or bumped (E.g., when crashing against any obstacle the robot should be able to redefine and adjust the trajectory).
- c) Sound sensor, preferably with pressure sound levels up to 90 dB, capable of detecting an extended range of sounds [dB] or just human perception sound range (in adjusted decibels [dBA]).
- d) Light sensor, capable of determining the intensity of existent light in the environment (light/ darkness) or the intensity of the light reflection on coloured surfaces.
- e) Acceleration sensor, capable of determining in threes axes the acceleration and tilt with respect to certain reference point (preferably with a range of precision of ± 2 degrees).
- f) Gyro sensor, capable of determining the direction of rotation and the degrees per second per rotation.

The action took due to signals acquisition is developed by actuators. These actuators, besides of some additional mechanical systems, give the capacity of movement to each robot. For the basic system, it is proposed the use of DC servo motors with rotation sensors that permit determining the rotation of the motor in degrees (preferably with a precision of ± 1 degree) and the control of different velocities.

Although the basic system mentioned includes only seven different types of sensors and one type of actuator, there is no limitation to expand the system and implement as much inputs (sensors) and outputs (actuators) as possible. Recall that the previous system was a minimum and with new implementation the system can be expanded.

In order to process the signal we establish the first group of variables that will be considered for the intelligent control applied on each robot:

- a) Velocity.
- b) Distance.
- c) Contact.
- d) Angle.
- e) Intensity (E.g. light, sound).
- f) Rotation.

These variables are used for the fuzzy controller's implementation, which governs the behaviour of some members of the community (E.g. the leader robot). Figure 10.4 shows the flow diagram of the fuzzy controller system design proposed. The process to compose the fuzzy controller is presented below with a brief explanation:

- a) Data input due to sensors read variables.
- b) Signals fuzzification process.
- c) Rule evaluation with membership functions. The input data is fuzzified.
- d) Tuning and evaluating the inference rules (with IF-THEN rules previously established).
Several experiments are done to find the most efficient operating point.
- e) Inference method.
- f) Defuzzification. The outputs of the fuzzy controller are obtained. The results are used to control the reactions done by the actuators in the robot.
- g) Evaluation of the results of the controller.
- h) System outputs.

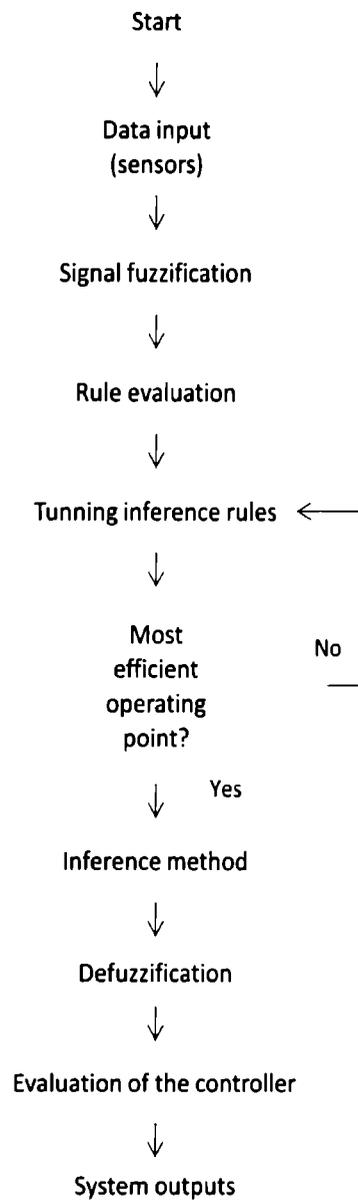


Figure 10.4 fuzzy controller system flow diagram

Figure 10.5 shows the process of the fuzzy controller for labyrinths solution and obstacle evasion. The explanation is described in the next steps:

- a) Linguistic variables are used to catalogue the data feed from the sensors, E.g. variables can be "D = Displacement", "R = Rotation of motors", "V = Velocity", among others.
- b) Linguistic terms are assigned to the linguistic variables, such as "VB = Very big", "B = Big", "I = Ideal", "S = Small", "VS = Very small".

- c) The signals fuzzification process takes place; it delivery degrees of membership in a range from 0 to 1, for the linguistic terms mentioned.
- d) The interference “If-then” rules are tuned and evaluated, such as “If the Velocity is Very big, then... <consequent>”
- e) The defuzzification process is done with the centre of gravity method to obtain the outputs of the controller.

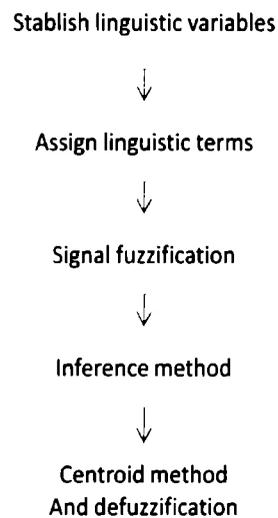


Figure 10.5 fuzzy controller for labyrinth solution and obstacle evasion flow diagram

We proposed another fuzzy controller but using the inverse model⁷ technique, due to the use of DC servo motors that let us easily model the behaviour of the robot according to the whole range of possibilities during the labyrinth solution.

The behaviour of the robot is model with respect to the angle needed to avoid obstacles and follow the trajectory of the labyrinth, setting the range from 0° to 360° in both clockwise and counter-clockwise directions. After obtaining the fuzzy model the inverse model is constructed. Inverse model implementation gives the advantage of having a faster and more precise⁸ response, also that there is no need of a mathematical model for the system.

After programming the inverse model and then substituting it with the real motor, the behaviour of the robot is above the average compared to other controls.

Figure 10.6 shows the flow diagram of the process to implement the inverse model. The steps for the modelling process are the next:

- a) We determine the use of 2 ultrasonic sensors, at front side rear of the robot which will provide the sensors read that will determine the inputs of the system (Input1 and Input2).
- b) With the readings from both sensors a fuzzy controller is designed similar to the one described in figure 2, where measures from the sensors are classified in the linguistic rules such as "distance of sensor" with linguistic terms: "small, medium and big" where expert knowledge is applied to specify them; and the outputs are expressed in degrees of rotation necessary to avoid colliding. An example of the rules of interference set used is: "If the distance of the ultrasonic right sensor is medium and the distance of the ultrasonic left sensor is small, then the angle of rotation of the robot will be 90° clockwise".
- c) The output obtained from this first fuzzy controller is the angle that the robot must apply.
- d) The output of the first controller is used as the input for the second fuzzy controller.
- e) The corresponding angle to the desired rotation of the robot is modelled with respect to the power of each motor which allows the movement; we propose only two motors for this system (left and right). E.g., if the robot needs to rotate 90° clockwise, the necessary voltage or enough rotation of each motor is found through several tests so it allows the trace of the desired movement.
- f) The outputs are obtained (Output1 and Output2) in terms of voltage or rotation for each motor.

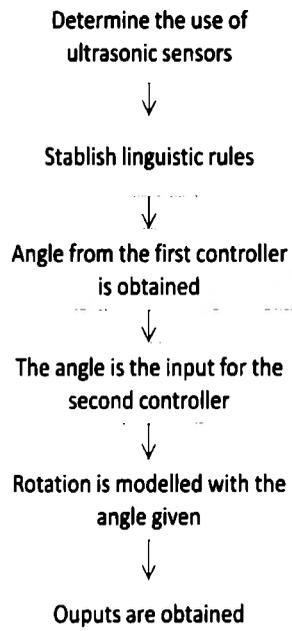


Figure 10.6 inverse model controller flow diagram



This project was more than a personal challenge as it involved many factors throughout its realization. The first big step was to learn artificial intelligent control, a very interesting science where studying about fuzzy logic, artificial neural networks, optimization algorithms and expert systems laid the groundwork for the development of the platform.

As a project, one can highlight the programming of algorithms concerning the above disciplines, which, as is the case of expert systems, required our expertise after extensive research and experience after carrying out numerous tests to finally adjust the parameters which would lead to better performance. One of the methods used was Mamdani-style inference method due to its versatility and for being one of the most explored and used nowadays.

As an educational platform, it leded the objective to generate a working methodology based on problem-based learning (PBL) methodology in a dynamic and didactic way that could be extrapolated to any field or branch of science for students from an early age. This led to the patent application having systematized the platform to support the principles of it. A key part was the use of PBL because it creates the theoretical framework to develop a series of examples or cases to solve. Also the minimal system is relevant as the resource to work with, referred to the robot and its components, actuators and sensors and finally the work area, the matt and panels that recreate various scenarios to solve new challenges to avoid monotony.

It is important to note that the use of Lego® MINDSTORMS® NTX kit facilitated the implementation of all the concepts and algorithms due to its software based on a powerful graphical programming language, LabVIEW. Even when it leaves the possibility to expand it into any system, we must recognize the advantages of these devices that integrate key functions such as Bluetooth communication for the robotics community.

As a small robotics community, it was possible to link the members to communicate and transfer learning patterns, allowing social learning to keep

cognitive consistency about their actions and generate knowledge and intelligence among all devices and assess the performance and evolution so that an error can be corrected by another and the capabilities inherited by another as well. For this reason the information was not specific routes or defined paths, but the robot parameters that will provide new skills to keep learning and improving performance in assigned tasks.

The achievements of the project were demonstrated in competitions and exhibitions where it was shown, with an innovative concept that proposed a different approach to traditional education with collaborative robotics and artificial intelligence. Better recognition could not have been achieved without winning the regional "Expociencias" as first and then second place in nationals with accreditation to attend the science summer camp in Lithuania. We performed the first phase of validation with youth from our same institution where their complete ignorance of the subject turned into interest when participating during the practices. It should be noted that as a team we could say that we were the first to validate the platform because we acquire lots of knowledge and experience as we worked to this excellent project.

- **Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems.** Guanrong Chen y Trung Tat Pham, USA 2001. CRC Press
- **Fuzzy Expert Systems and Fuzzy Reasoning.** William Siler and James J. Buckley, USA 2005. Wiley
- **FUZZY SETS IN APPROXIMATE REASONING AND INFORMATION SYSTEMS.** James C. Bezdek y Didier Dubois y Henri Prade, USA 1999. KLUWER ACADEMIC PUBLISHERS
- **FUZZY EXPERT SYSTEMS.** Abraham Kandel, USA 2000. CRC Press
- **PRACTICAL AND EXPERIMENTAL ROBOTICS.** Ferat Sahin y Pushkin Kachroo, USA 2008. CRC Press
- **Artificial Intelligence A Guide to Intelligent Systems.** Michael Negnevitsky, UK 2005. ADDISON WESLEY
- **Intelligent Systems for Engineers and Scientists.** Adrian A. Hopgood, USA 2001. CRC Press
- <http://Oieeexplore.ieee.org.millenium.itesm.mx/stamp/stamp.jsp?tp=&arnumber=932756&isnumber=20189>
- Sensors, Appliance Control, and Fuzzy Logic
<http://Oieeexplore.ieee.org.millenium.itesm.mx/stamp/stamp.jsp?tp=&arnumber=287530&isnumber=7165>
- La robótica como herramienta para la educación. (2000)
<http://www.donosgune.net/2000/dokumen/EduRobSp.pdf> Obtained on November 19th.
- <http://www.roboticaeducativa.com/sitio/modules/mylinks/singlelink.php?cid=2&lid=23>

Objective: Recognize the system which will be used for all the activities.

Material:

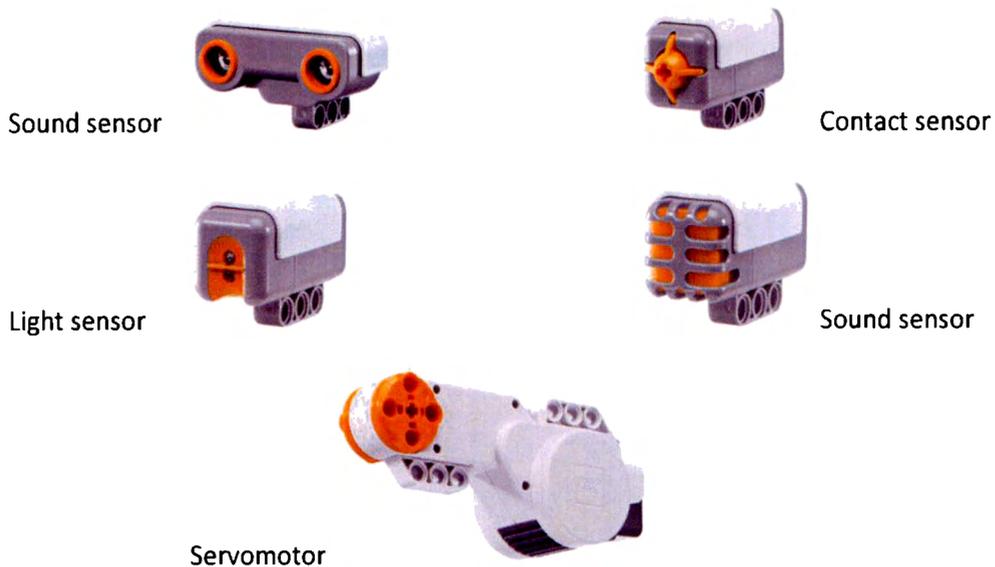
Lego® MINDSTORMS® NXT Kit (Basic)

Lego® MINDSTORMS® NXT Toolkit (required software for LabVIEW)

Procedure

Spread off the different pieces of the Lego® kit on the floor and try to separate them in categories due to their purpose, structure, colour or surface. Make a list of projects and ideas that came out as you analyse them and try to understand the connecting procedure for each one.

Locate the sensors, actuators and explain their function.





Several pieces

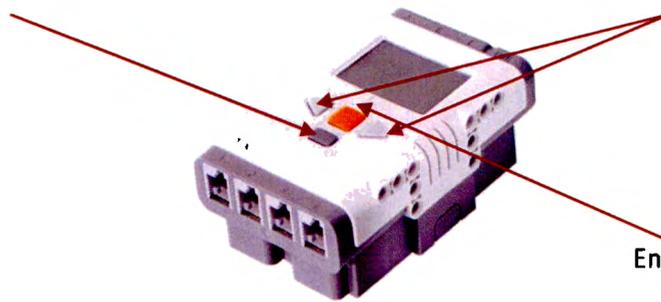


Cables

Once a complete view of the kit was done put the batteries necessary and turn on the brick (main process unit). Navigate through the different menus. Take your time to adjust the volume and determine the NXT version.

Back button

Cursor Buttons



Enter Button

Objective: Recognize each type of sensor. Explore its capacities and limitations to understand their operation mode in order to choose the indicated for each next particular project.

Material:

1 Brick NXT

1 Cable

1 Ruler

1 Ultrasonic Sensor

1 Light Sensor

1 Sound Sensor

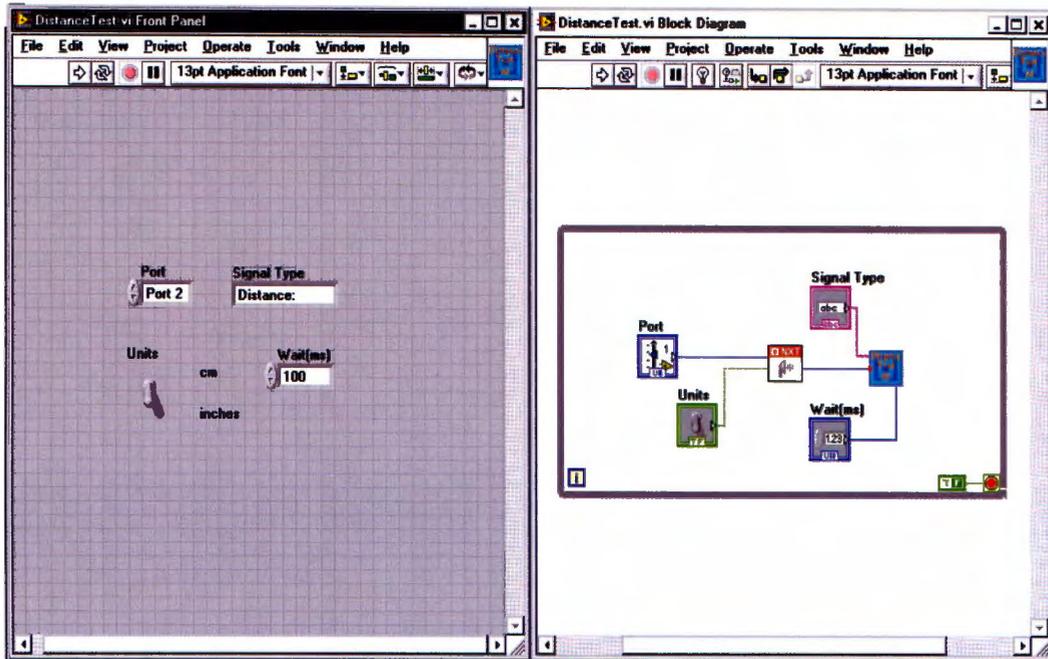
Procedure

1. Identify each type of sensor listed on the materials.
2. Test each type of sensor and answer the questions:

a. Ultrasonic Sensor

Connect one cable from a desired input port of the NXT to the sensor and turn the brick on. Place the ruler perpendicular to the wall or any object over the table.

Open a new file in LabVIEW and program the simple code shown in the figure below for the distance test:



Compile and download. Run

Take the corresponding measure with the ruler and answer:

What is the sensor resolution?

Determine the operating range.

What are the units of the distance?

Is it possible to change the units? How? Which ones are they?

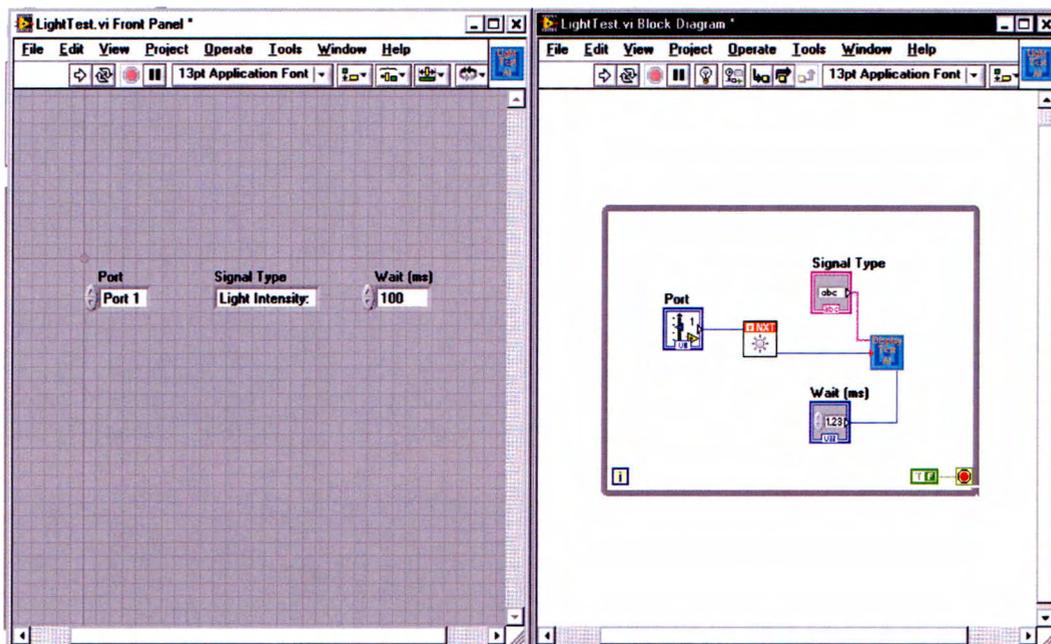
Is there any difference with each scale?

What happens if the object placed to measure the distance changes? Try placing your notebook, your hand and some other thing.

b. Light Sensor

Connect one cable from a desired input port of the NXT to the sensor and turn the brick on.

Open a new file in LabVIEW and program the simple code shown in the figure below for the light intensity test:



Compile and download. Run

Place different objects towards and away the sensor and answer:

What is the sensor resolution?

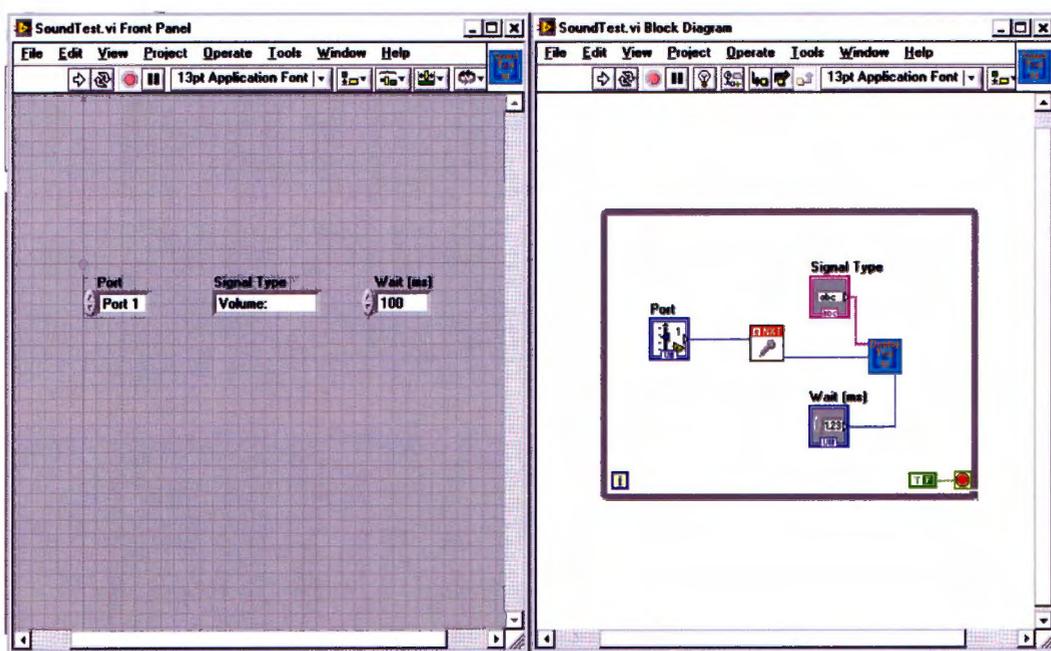
Determine the operating range.

What happens if you point different light sources to the sensor? Try different colours at 10cm and fill in the intensity

c. Sound Sensor

Connect one cable from a desired input port of the NXT to the sensor and turn the brick on.

Open a new file in LabVIEW and program the simple code shown in the figure below for the sound test:



Compile and download. Run

Use a sound source such as the computers speaker, place the sensor 10cm away from the source and turn up and down the volume.

What is the sensor resolution?

Determine the operating range.

Measure the volume displayed in the brick due to: a) claps b) talk c) shouting d) hit the floor.

Objective: Identify the actuators structure. Understand the different types of motor control; identify the advantages and disadvantages for several purposes.

Material:

1 Brick NXT

3 Cables

3 Motors

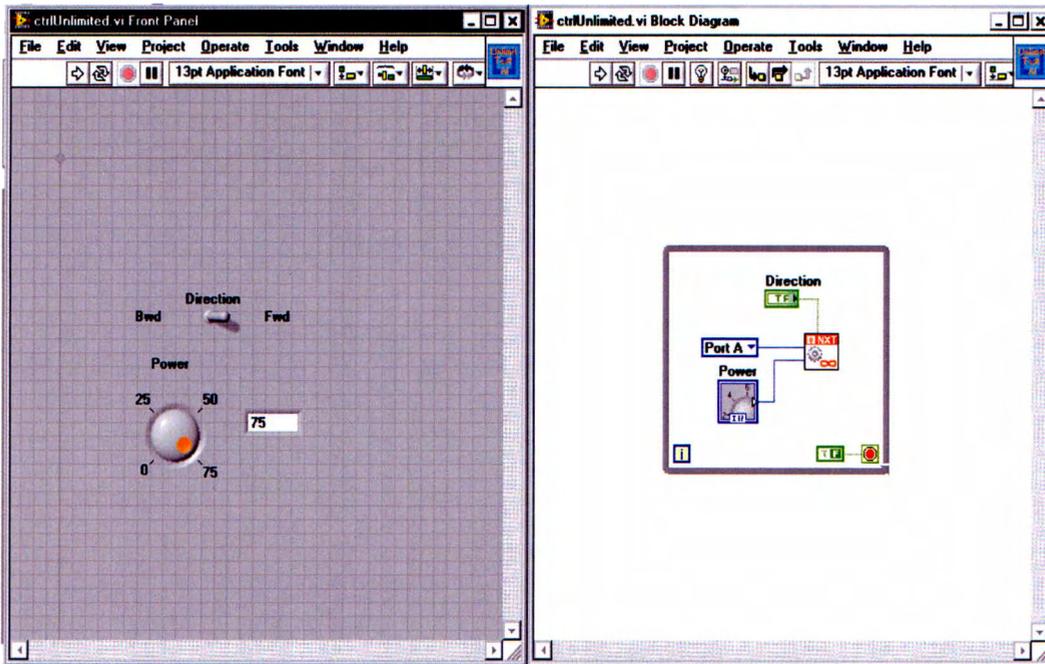
3 Wheels

Necessary pieces

Procedure

1. Take 3 long cables and connect each motor to the different output ports on the brick. Connect one wheel on each motor as convenient.
2. Find the code for unlimited, distance, time and all.

- a) Open a new file and select the VI “ctrlUnlimited”, compile and download. Run and observe the behaviour of motor A.

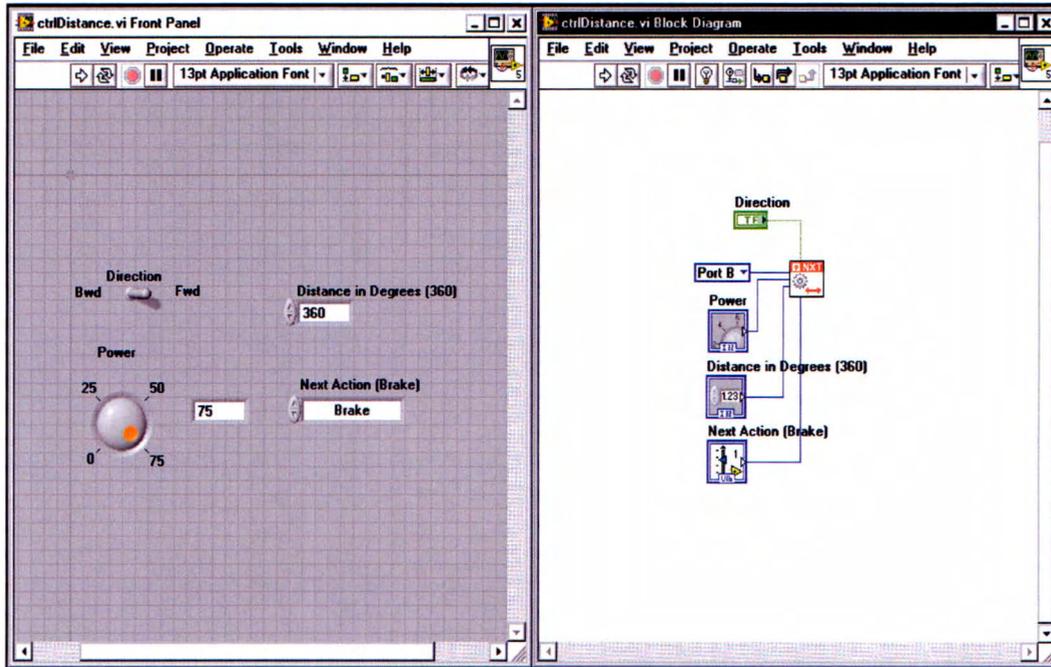


After you test the program running ok, try these little changes (REMEMBER: For this activity you will only change values in the Front Panel with the available controls):

Open the VI and modify the power for the assigned motor (WARNING: Do not change Port A, you will need them!). Observe what happens with the motor. How could you explain the new behaviour?

Open the VI again and change the direction. Observe. Is there any difference in its speed or anything else? What does the direction means respect to the behavior of the motor?

b) Open a new file and select the VI "ctrlDistance", compile and download. Run and observe the behavior of motor B.

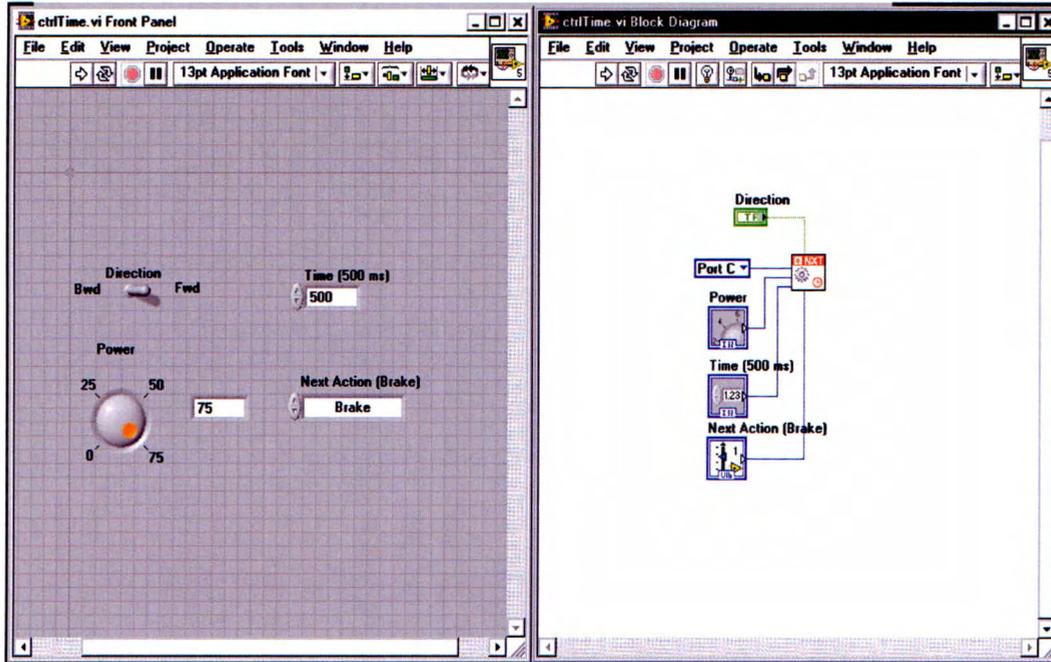


After you test the program, try these simple variations:

Open the VI and modify the distance in degrees and try the new code. Observe the new behavior, how could you explain the change?

Change the next action for the motor and watch carefully, is it important to consider?

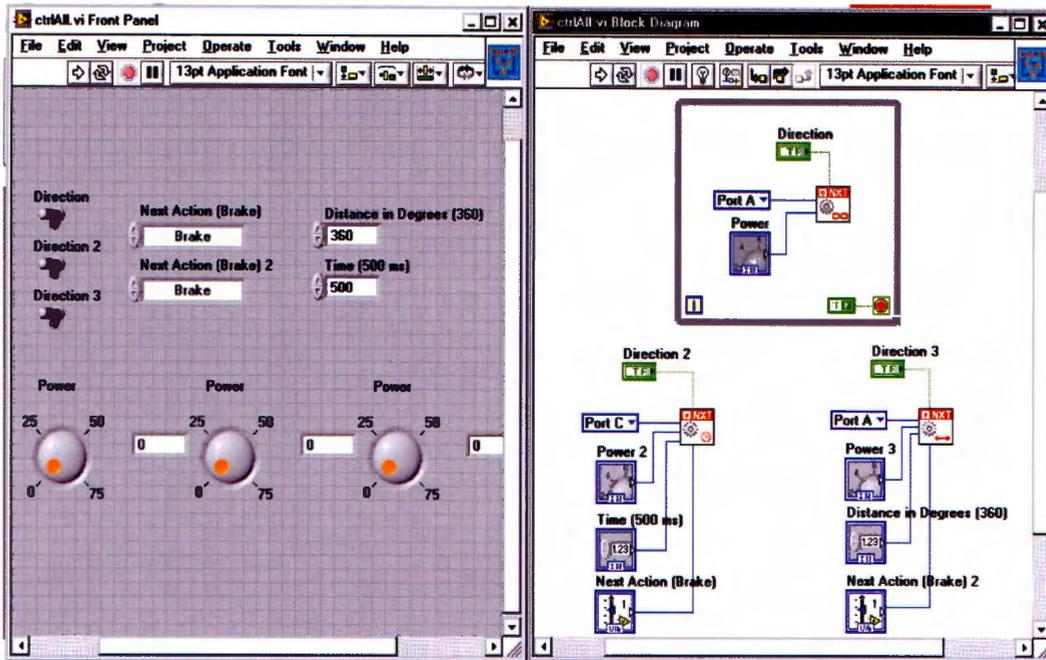
- c) Open a new file and select the VI "ctrlTime", compile and download. Run and observe the behavior of motor C



After you test the program, try these little changes:

Modify the time value in milliseconds, compile and download. Run and observe the behaviour of motor C. Is there any way to increase the time the motor is turn on? Propose a solution and discuss.

- d) Open a new file and select the VI "ctrlAll", compile and download. Run and watch in detail the three motors running at the same time.



Make a comparison between performances of the three motors paying attention on their speed, time they are turned on, resistance due to pressure from your fingers and other important issues you might perceive. Think about the possible uses for each motor, advantages and disadvantages, could we use all of them for the same purpose?

Objective: build the first robot prototype which will be extensively used during almost all the activities. This robot will allow the student to apply all the knowledge acquired and learnt during the whole course.

Material:

1 Brick NXT

4 Cables

2 Motors

2 Wheels

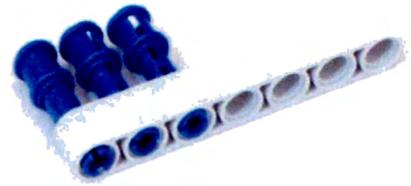
Necessary pieces

Procedure

1. Put all the material needed to build the robot together in a reachable zone for arms and organize it. Remember that you will be using several small pieces and it is recommended that you have a clean workspace before you start constructing the prototype.
2. Observe the built robot and then follow the building instructions. (Note: there is always the possibility to make little changes to your own creation, dear to make something new!).



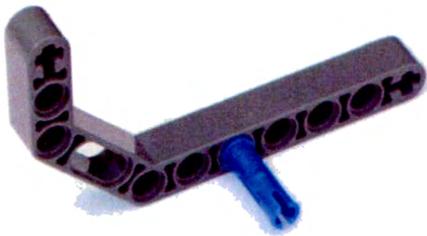
Take three blue double connector pegs w. friction and join them in the last three holes of the seven-white-lever.



Take four black connector pegs w. friction and one blue connector bush w. friction/crossale and put them together in an L-shape-lever as shown.



With the J-shape-lever join one blue double connector w. friction.

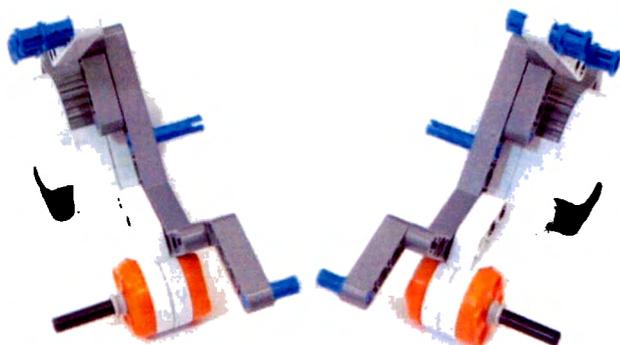


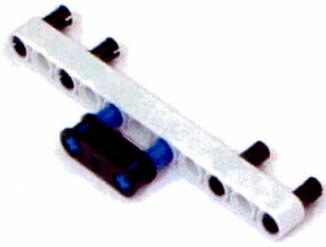
With a V-shape-lever, one connector peg w. friction and one blue connector bush w. friction/crossale do the next



Make the assembly of all previous subassemblies for the side structure as shown in the picture.

Remake the first part but this time changing the direction of the connections to get a right-side structure for the main structure.

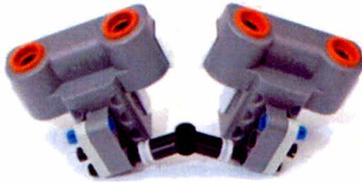




Assembly the next pieces in an eleven-white-lever with necessary connectors. Both left and right pictures are the same component.



The next step is to assembly the support for the ultrasonic sensors as shown in figure with the angle element.



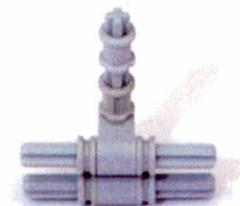
Take the two ultrasonic sensors and join a lever and two connectors to make the support for sensor itself. Make sure that each sensor has the lever pointing out, otherwise the sensors will mismatch.



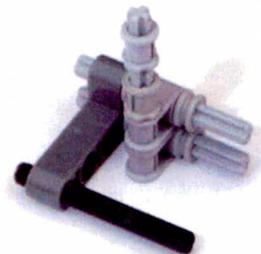
Take the piece made with the angle element and join the entire sensor mount.



The next assembly is made to guide the robot with the direction given by the servo motors, and it is important to recall that this assembly could be substituted with another solution.

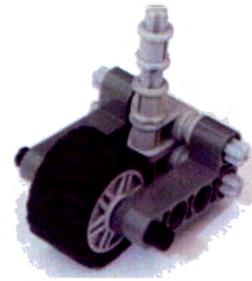


Take another L-shape-lever and pass an axle through the first crossale hole. Take the past assembly and join them as shown in figure.

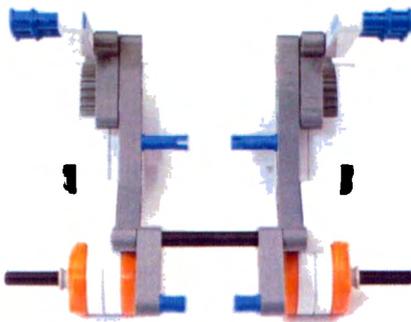
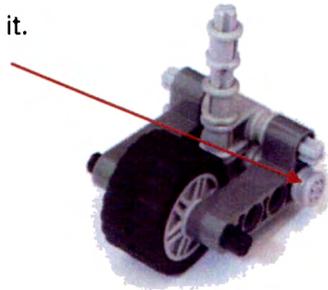




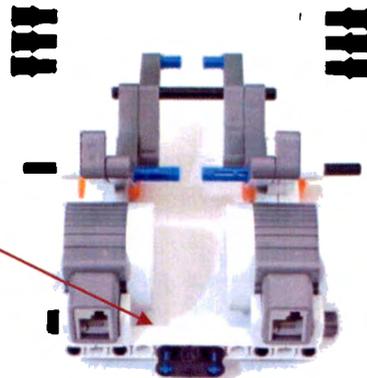
Pass the wheel through the axle and join the other L-shape-lever



Put one bush on each side of the structure to reinforce it.



Join the two motor structures with an axle.



Flip backward the structure and join the component shown



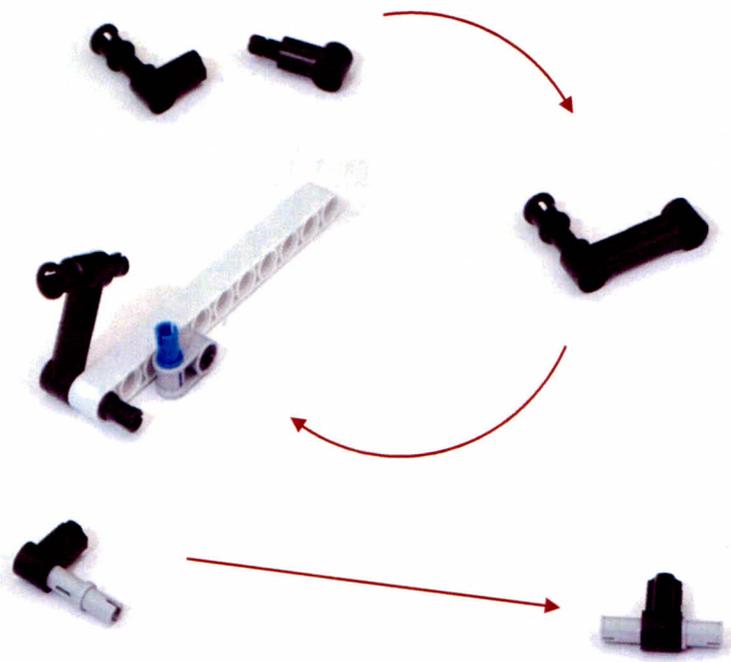
Now the structure for the contact sensor is presented, just make the subassemblies and put all together.

This is one side lever, be aware on the connection peg w. friction of one, just follow the pictures.

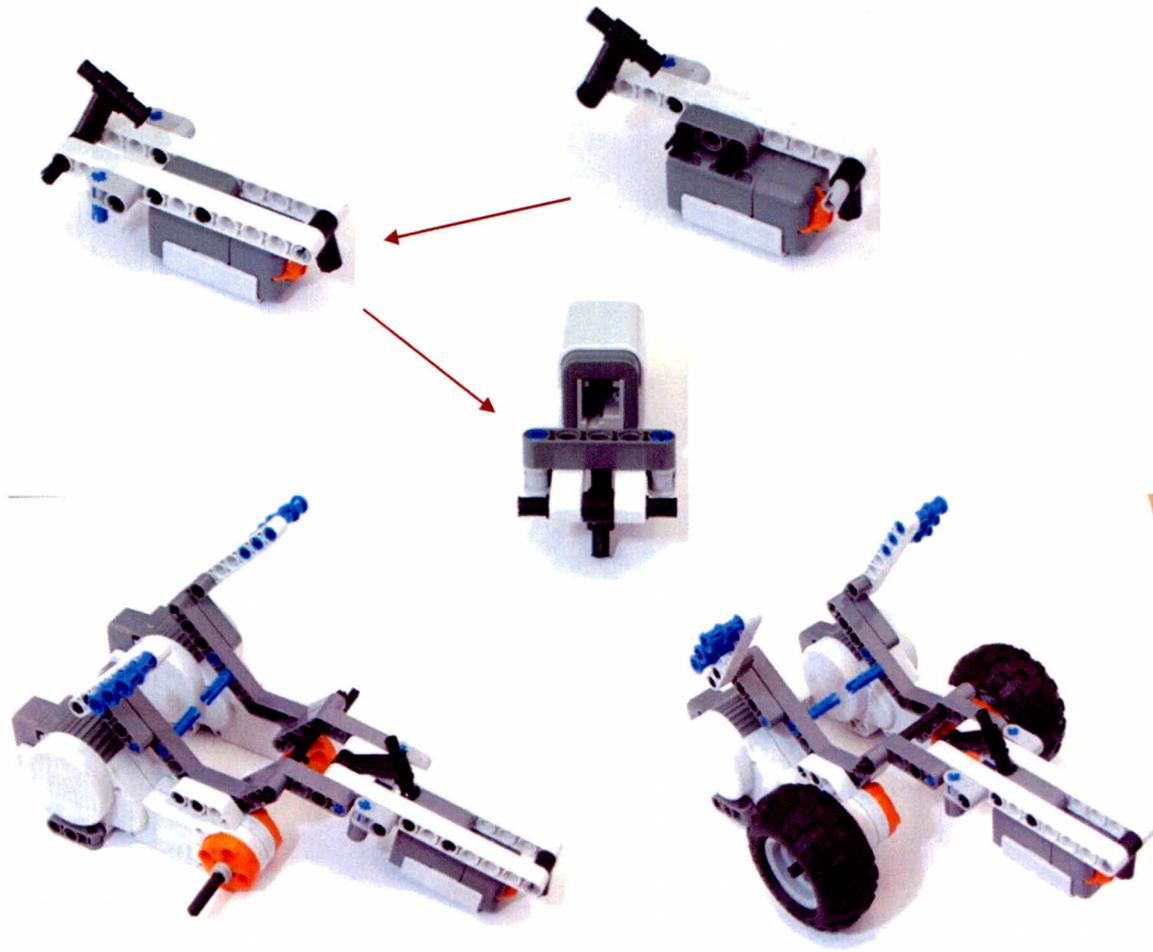
2



1



Join the pieces built into the structure, flip the structure and put the five-gray-lever.





Finally, join every assembly into the robot as shown. Do not forget the wheels and the backside free wheel.

Connect the sensors and the brick before you connect the cables for each input and output. The robots should look like this unless you enhanced your prototype!



Objective: Explore the LabVIEW programming environment and get familiarized with the NXT Toolkit and NXT Module.

Material:

1 Robot

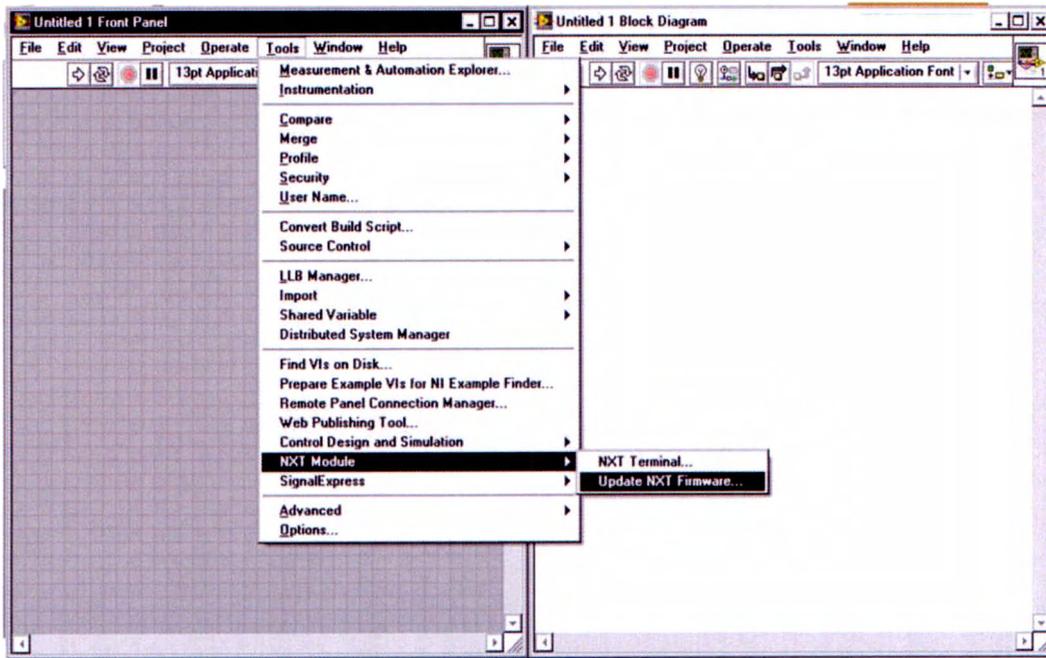
1 USB - Cable

1 Laptop with LabVIEW installed

Procedure

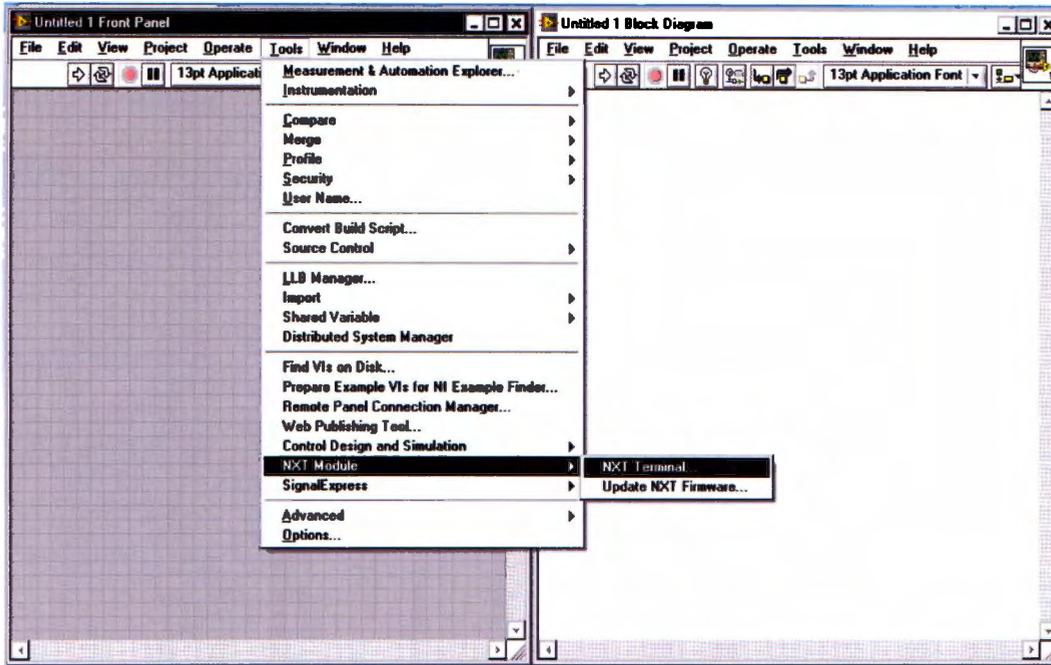
1. Follow the simple steps to get all configured and explore in detail every section carefully once you've done all the previous settings.

In order to test and develop new programs in LabVIEW, you must remember that previous to connect the NXT to the computer the NXT Toolkit must be installed. It is important to recall that the firmware must match the Toolkit version, if not, try to update both of them. Plug in your Robot with the USB cable and proceed. To set this view press Ctrl+T or go to the Window menu, where you can find more options.

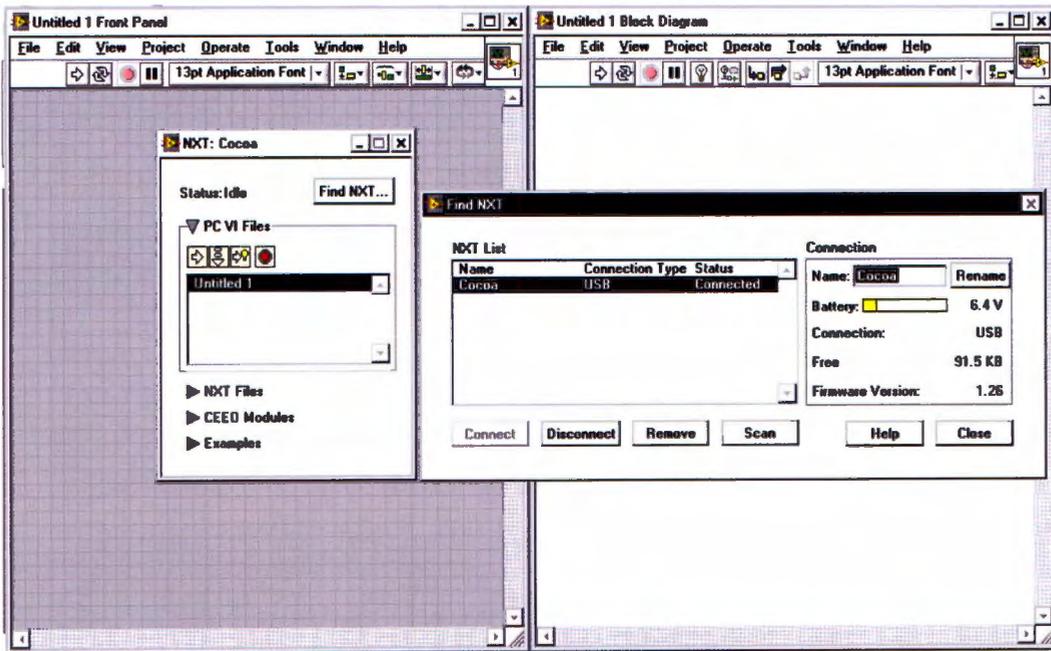


Once you have installed the Toolkit and you confirm that the Firmware matches proceed to rename your own Lego®. This name will identify the brick among the others while communicating and with the computer when plugged in or scanned by Bluetooth.

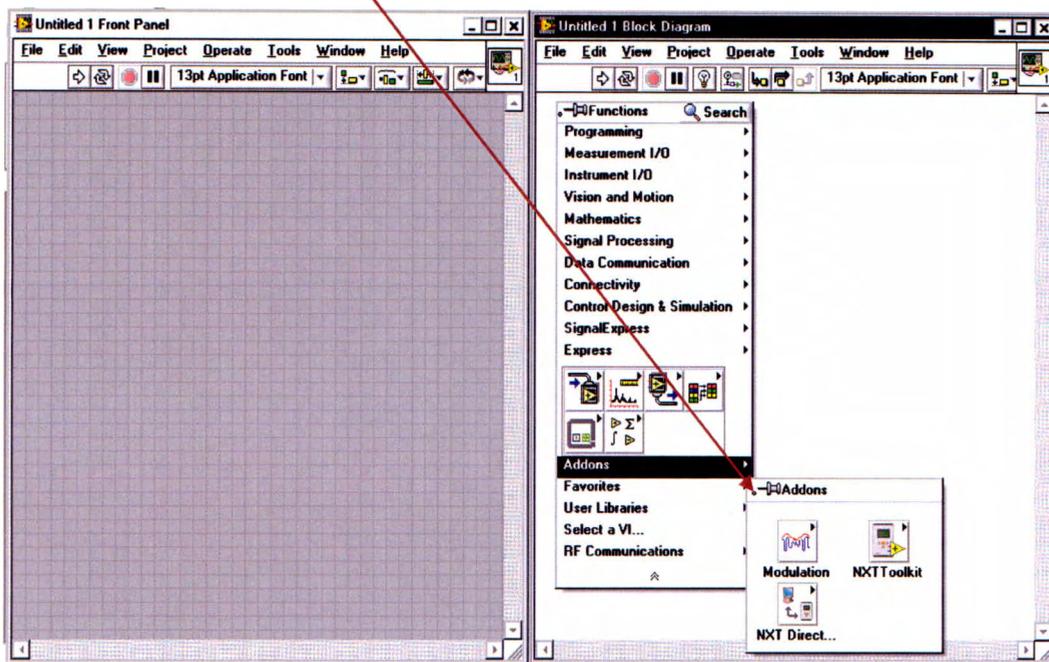
To rename your Lego® go the NXT Module, open the NXT Terminal, click “Find NXT ...” and you will be prompted to a window where some basic configuration for the Lego® NXT Brick could be done. Type the new name and click “Rename”



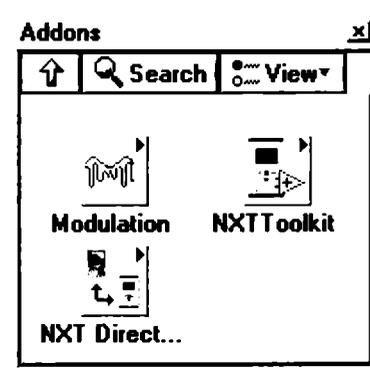
Find NXT Window



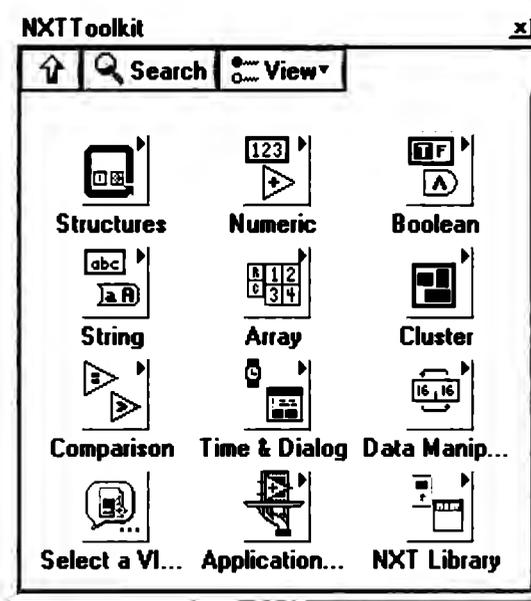
Start placing the mouse over the Block Diagram and right-click the screen and the Functions Palette will prompt. The Lego® NXT Toolkit is found in the Addons section, go a check what is in there. To set the Addons Palette click the tack at the upper left corner.



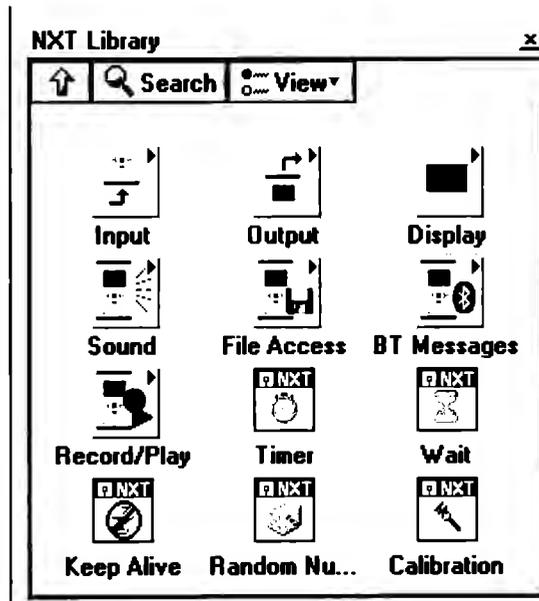
Once you set the Addons Palette en the Block Diagram Window, you will be able to explore easily the Toolkit without being prompted out from the Palette.



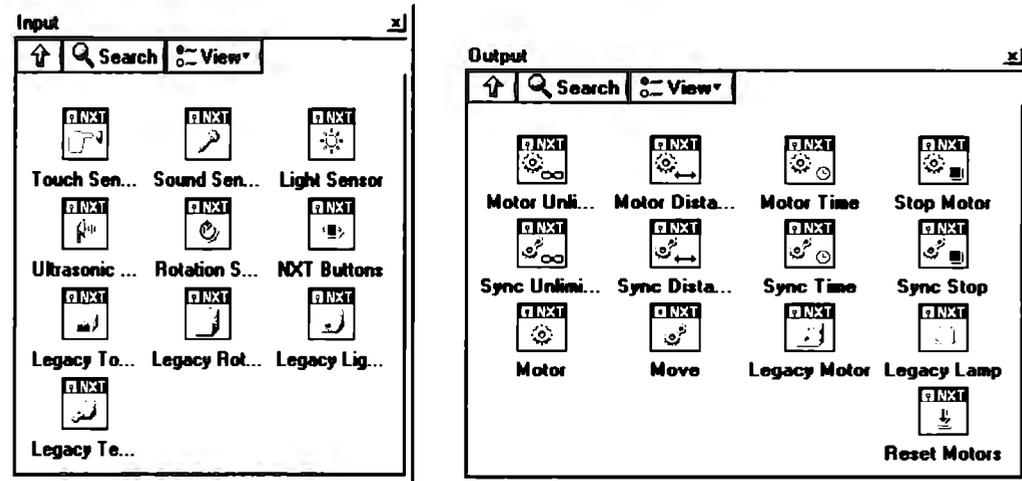
Click the NXT Toolkit and identify each sub Palette inside it.



Try searching in the NXT Library first.



Check out the Input and output Palettes.



Keep searching beneath all the Palettes available in the Addons palette and mainly in the Toolkit Palette. You do not need to understand all the blocks out-of-the-box, with the next activities you will gain the expertise programming with LabVIEW and developing new programs.

Objective: Test a simple code in LabVIEW to understand the VI's integration as well as some useful functions for more complex control.

Material:

1 Robot

1 Sound sensor

1 Laptop with ® installed

Necessary pieces

Procedure

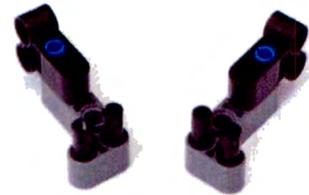
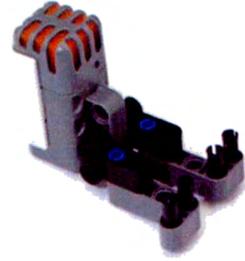
1. Follow the instructions to implement the Sound sensor in your robot.
2. Test the robot with the given code.

The control for this activity requires that you implement one Sound sensor at the top of the robot.

Follow the pictures step by step to build the structure.



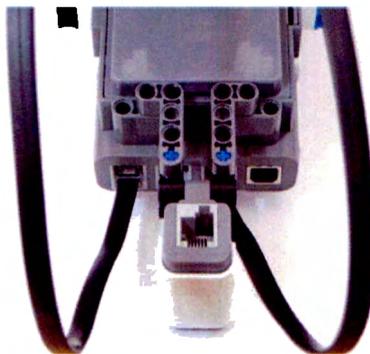
Join two double connector pegs w. friction in the soundsensor.



Use spread parts to arrange this configuration; both of them are necessary, left and right. Put all together and take your robot from the back.



These 4 holes at the top of the brick will be used to attach the Sound sensor. The reason for the Sound sensor to be in this position is that being at the top no sound would be attenuated by other parts of the system.

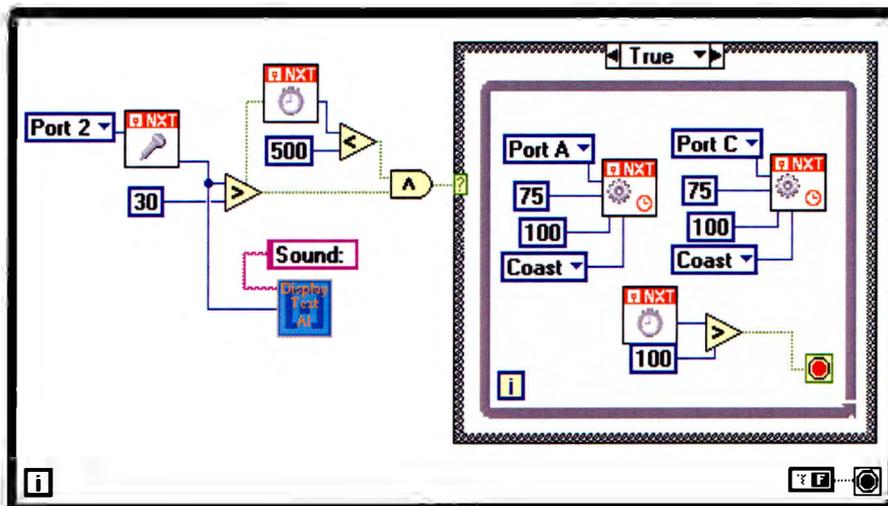


Your robot should look like this



Connect one cable from input port 2 of the NXT to the sensor and turn the brick on.

Open a new file and select the VI “On-OffTest”, compile and download. Run and watch the behavior of the robot due to the next actions.



Note that this code includes the well known VI “Display text” from the past activities with the same purposes, control and debugging.

Clap your hands at a reasonable distance from the robot. What happens? What happens if you clap louder? And faster?

Start a conversation in front of the robot and see what happens if you low your voice and if you raise your voice.

What does the value "30" means? Explain.

After you test the robot, try this little changes and see what happens. Change the motor power, the comparison volume, the time the motors are on and manage the timers' values.

Objective: Understand the action of the P controller for a first approach to conventional control. Manipulate the control to achieve the best performance.

Material:

1 Robot

1 Ultrasonic sensor

1 Laptop with LabVIEW installed

Necessary pieces

Procedure

1. Follow the instructions to implement the Ultrasonic sensor in your robot.
2. Test the robot with the given code.

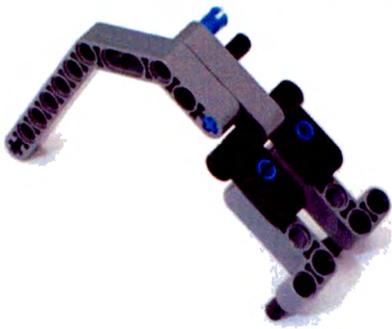
The control for this activity requires that you implement one Sound sensor at the top of the robot.



Follow the pictures step by step to build the structure. You can recycle the structure from the previous activity by just making few changes for the ultrasonic sensor.



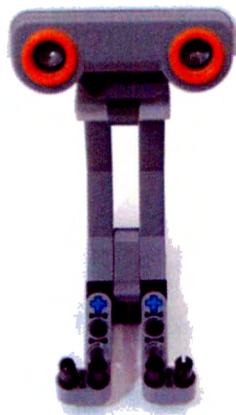
Instead of the Sound sensor, change the structure to another L-shape-lever and connect the given connector pegs.



Join the first J-shape-lever to the assembly. Put the two levers, the V-shape lever and the 5-white-lever as shown in the figure.



Join the second J-shape-lever to the other side. Then join the Ultrasonic sensor and finish the structure with another 5-white-lever in the others.



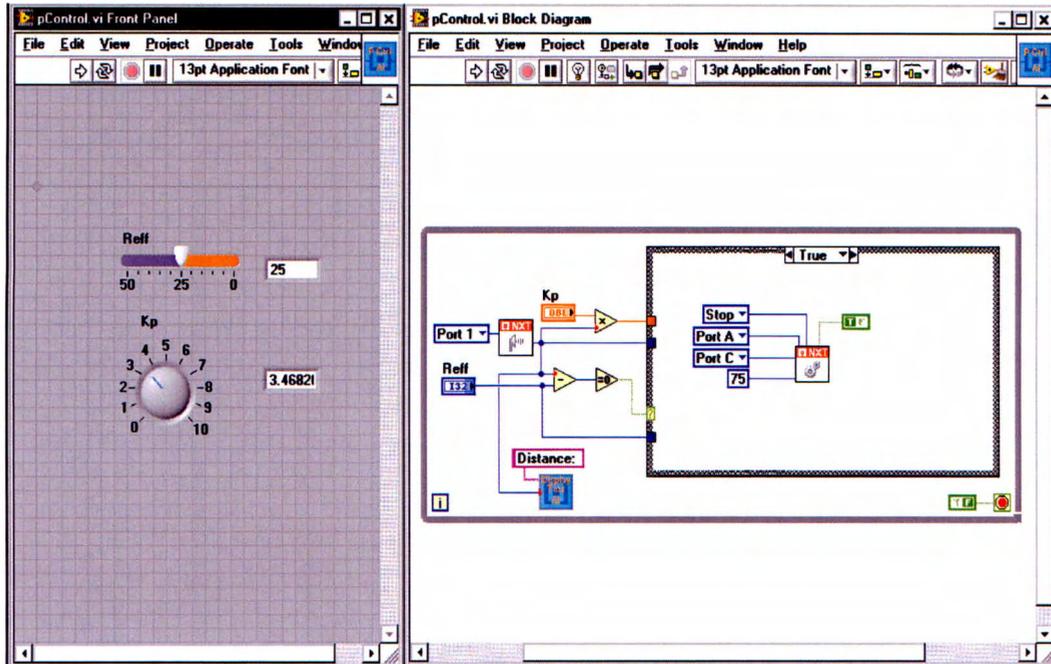


The same holes for the Sound sensor will be used to attach the ultrasonic sensor as the basic structure is the same. Then your robot should look like this



Connect one cable from input port 2 of the NXT to the sensor and turn the brick on.

Open a new file and select the VI "pControl", compile and download. Run and watch the behavior of the robot.



Place the robot about 60 cm from a wall and run the program. What happens with the robot? What is it trying to do?

What happens if you place your hand in front of the sensor? Does it change the robot behaviour?

Why the robot does never stops?

After you test and think about the answers try these changes in the program.

Change the "Reff" value to change the distance the robot will try to stop at. Compile and run the program to see what happens.

Now is your turn to tune the Kp gain to improve the robots behaviour. Make a few tests with different values for Kp and determine the best value for the controller. Is it really the best result the robot could obtain? What else could be done if the P control does not fill the operation needs?

Objective: Learn basic theory for the introduction of the intelligent control for the rest of the activities. Understand the meaning of “Artificial Intelligence” and introduce the “Expert Systems” concept.

Material:

1 Robot

1 Laptop with LabVIEW Installed

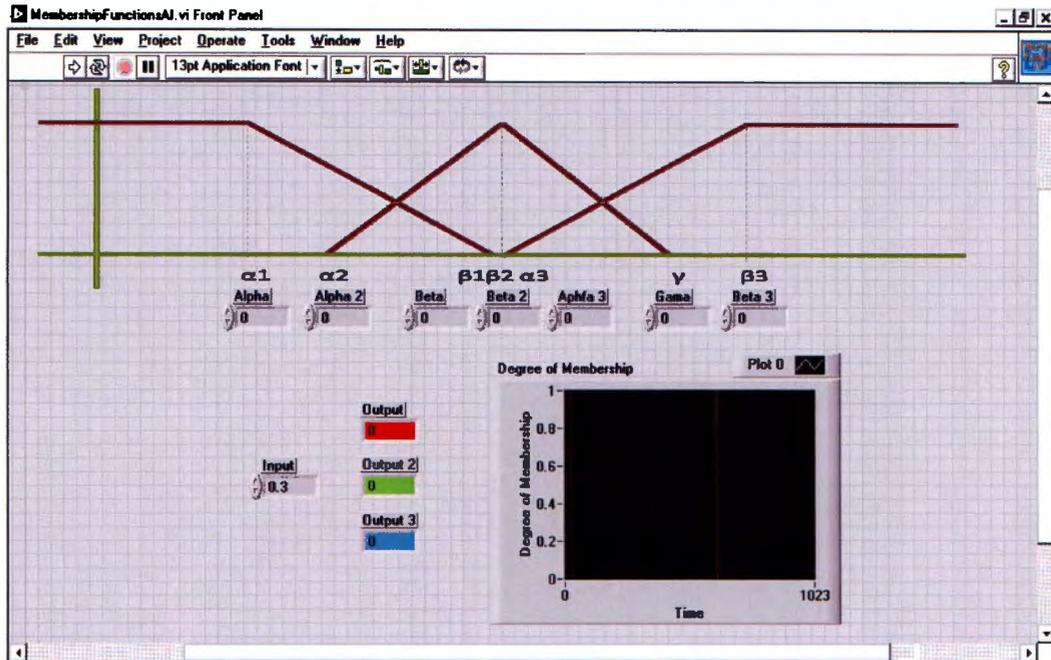
1 Ultrasonic sensor top mounting (done in the activity 6)

Necessary pieces

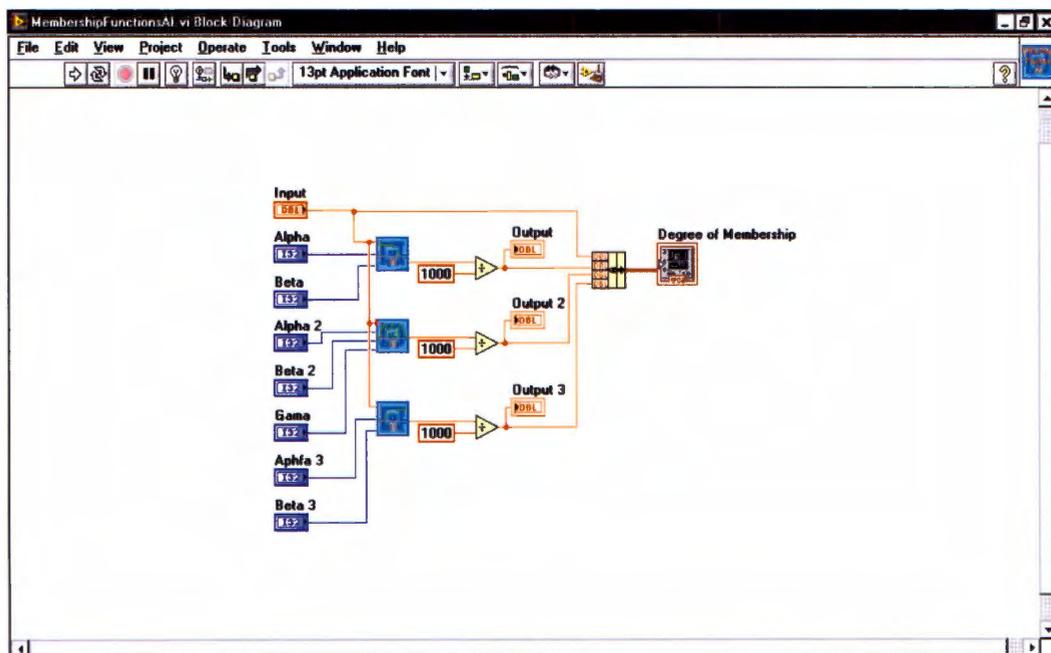
Procedure

1. Follow the instructions to complete all the code tests and modifications

Open the VI "MembershipFunctionsAI". This time you will use first the LabVIEW environment to test the code to understand the introduction of the Fuzzy Logic control and specifically the membership functions and the process of fuzzification.



Block diagram



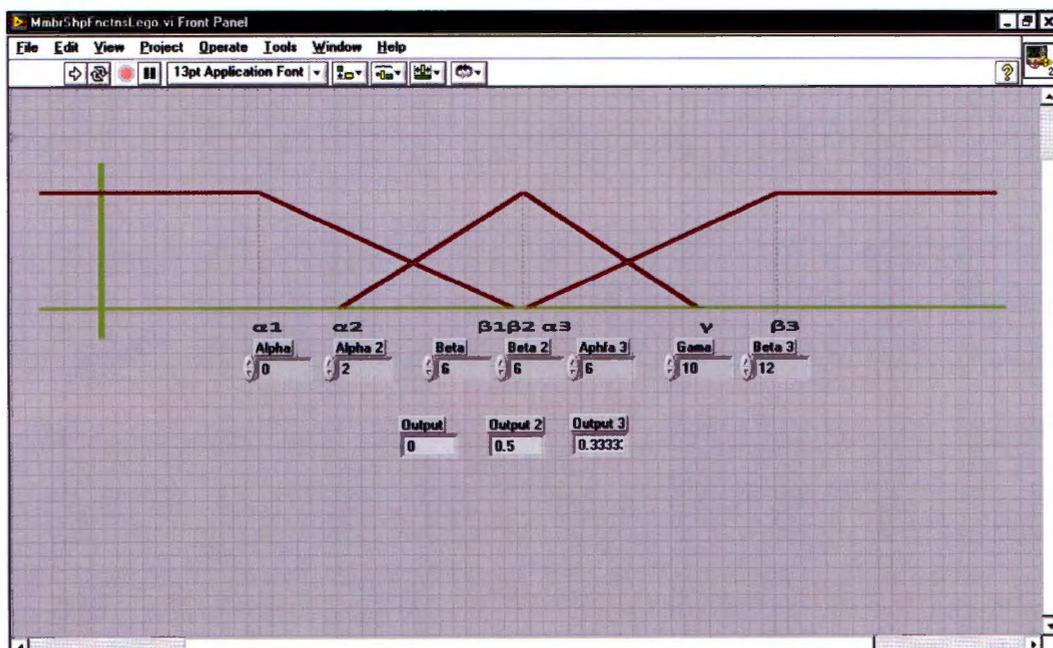
Set the values for α_s' , β_s' and γ to:

- $\alpha_1 = 0$
- $\beta_1 = 6$
- $\alpha_2 = 3$
- $\beta_2 = 5$
- $\gamma = 7$
- $\alpha_3 = 4$
- $\beta_3 = 10$

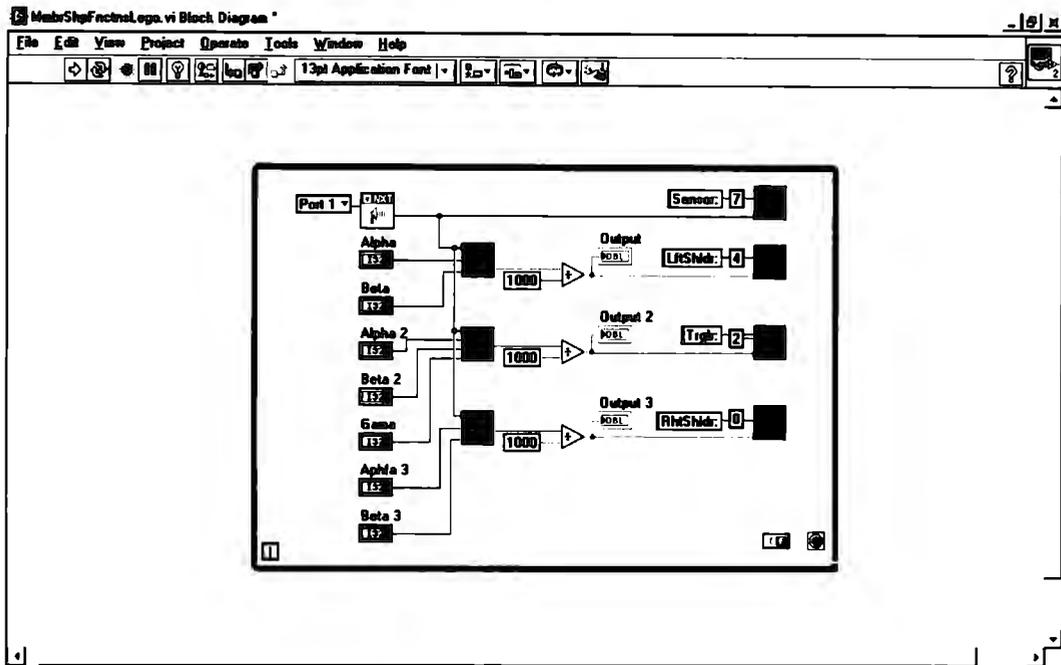
Run continuously the program and keep modifying the value of the input. Observe what happens with the chart and the outputs boxes? Does it make any sense to you? Write a brief explanation of what is going on with the program.

Try to modify the values for the α_s' , β_s' and γ to set a new universe of discourse for the membership functions. Run the program and check that everything is running as desired.

After doing the process of fuzzification with theory, open the VI "MmbrShpFncnsLego". For this real approach use the Ultrasonic sensor top mounting done in activity 6, connect the sensor to Port 1. Recall that the Lego[®] Ultrasonic sensor has a specific operation range. Taking this consideration, determine the universe of discourse for the membership functions to guarantee that the values read from the sensor will be useful to them. Set the values for the α_s' , β_s' and γ .



Block diagram



Compile and download to the NXT brick. Run the program and watch the display. Answer the questions.

What is it doing?

What are the values displayed?

Is it the same as in the LabVIEW test?

Objective: Understand the whole process of the Fuzzy Logic Inference and apply the technique in the robot's behaviour. Prove that the expert knowledge is vital for the performance of the controller. Understand how the linguistic rules are determined and evaluated.

1. Fuzzification
2. Rule evaluation
3. Aggregation of the rule outputs
4. Defuzzification

Material:

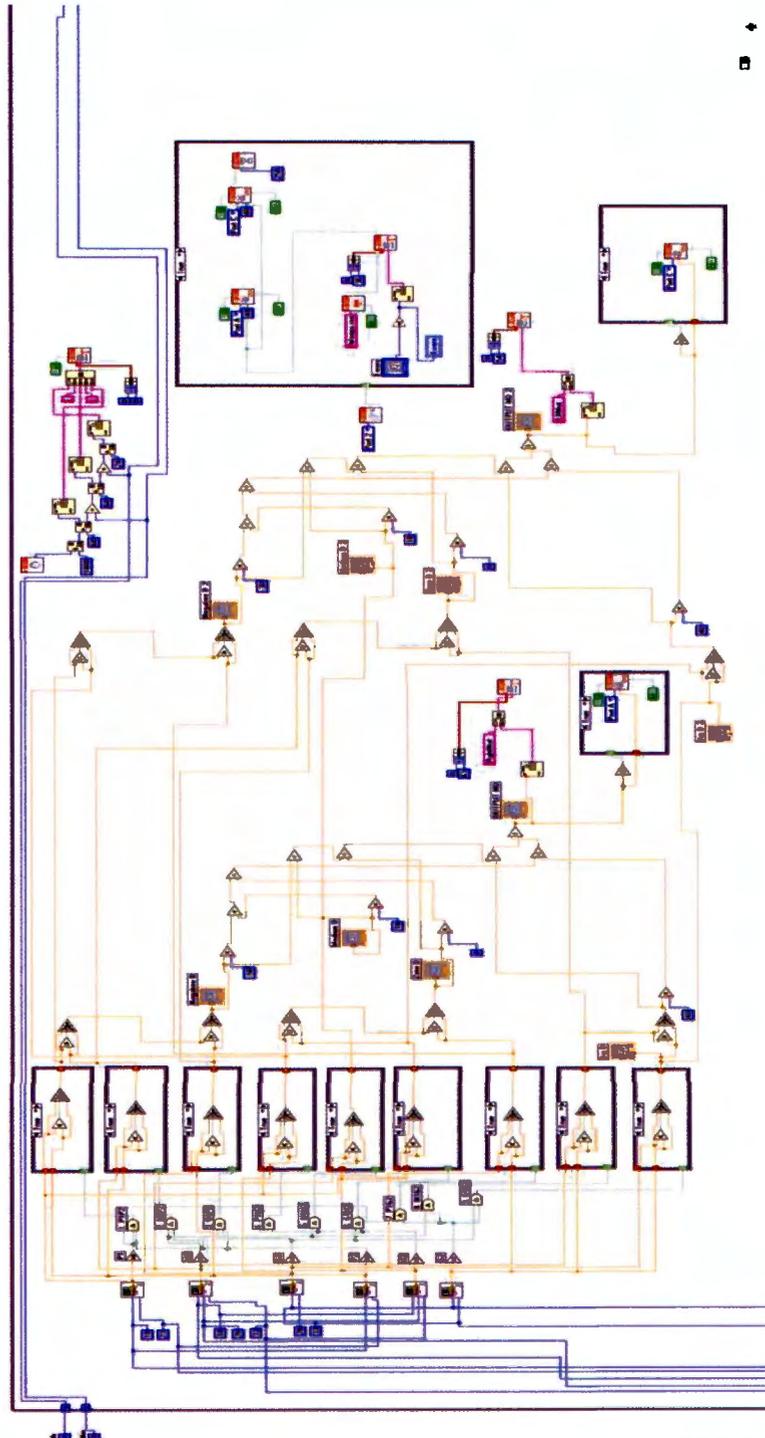
1 Robot

Necessary Panels

Procedure

1. Built a labyrinth with different panels, try to include curves, straight lines and zigzags to test the robot through different situations.
2. Test the robot's behaviour with the given algorithms.

Open the VI "SpeedCntrlSndr(Graph006)". Download and compile to the NXT brick. For this activity it is not required to understand the code presented for the Bluetooth communication, it will be used in activity 11 so do not pay attention to that part of the code. Run the program and let the robot make its first run through the labyrinth. Answer the questions.

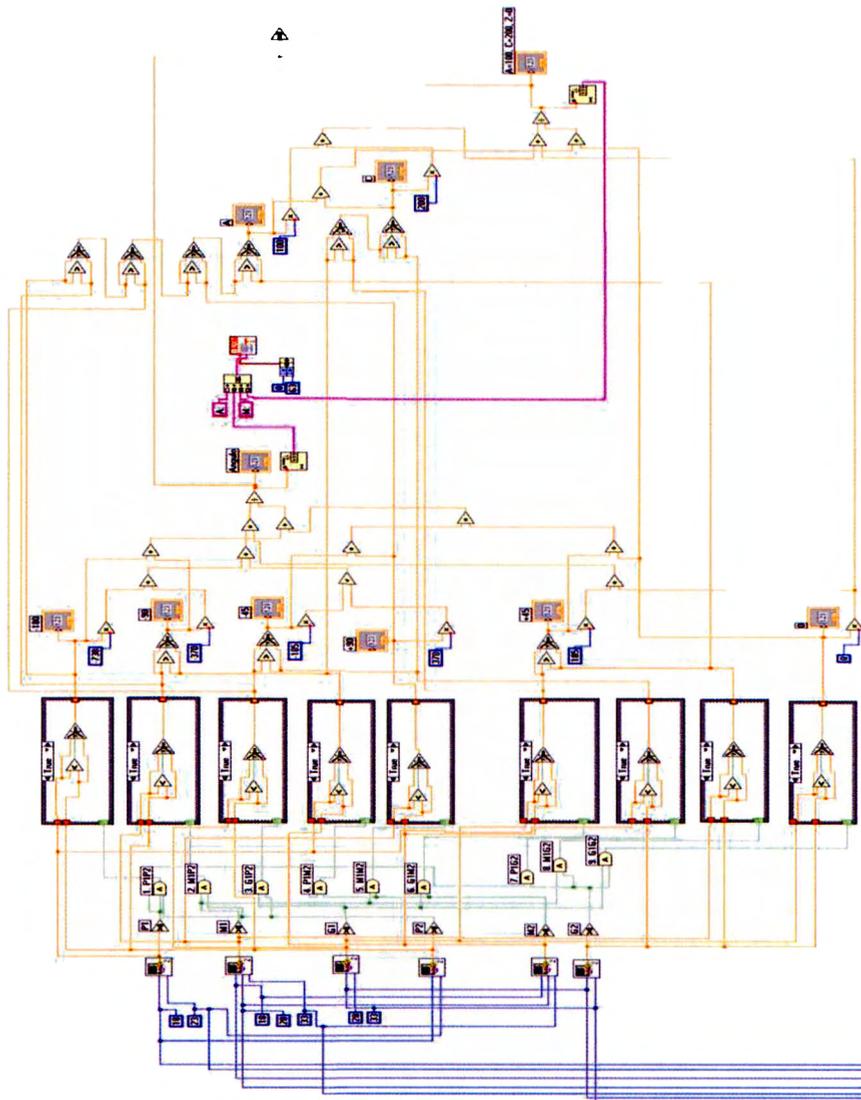
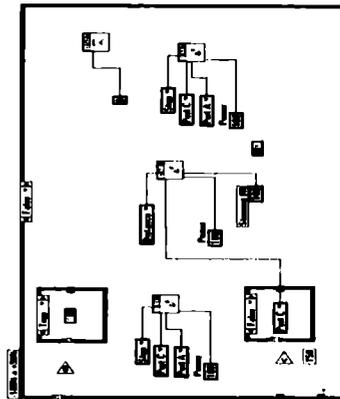


Watch the chronometer displayed in the brick as soon after the robot get to the end of the labyrinth. How much time does it take to the robot to get to the end?

How smooth was the route traced by the robot?

Do you think you can improve its displacement? Try to make a little change in the code varying the membership functions' universe of discourse. Download and compile the code. Run again and take your notes. Compare the results with the previous run.

Open the VI "InverseCtrlSndr(Graph016)". Download and compile to the NXT brick. Run the program and let the robot make its first run through the labyrinth. Answer the questions.



Watch the chronometer displayed in the brick as soon after the robot get to the end of the labyrinth. How much time does it take to the robot to get to the end? Was it the same as the Speed Control? Explain.

How smooth was this time the route traced by the robot? Was it better or worst than the previous Controller?



Objective: Realize the need of communication between the community to keep knowledge consistency and spread the learning patterns within the members to increase "intelligence".

Material:

Available Robots (two or more)

Labyrinth

Procedure

1. Pair your robots following the instructions to do so.
2. Built a labyrinth with the panels
3. Run the given programs to use the Bluetooth communication between the robots.

To establish a communication between robots you must first pair each other. This process will be willing to establish a connection and pairing devices is made only for the first time. Recall that every time you turn off the brick you will have to re-establish the connection, but it will be as simple as looking in your contacts as the robot "already knows" its pals.

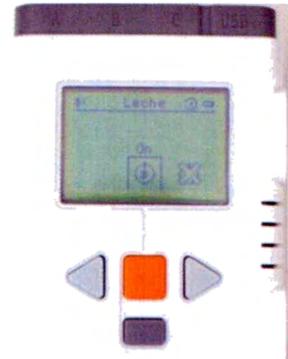
In order to accomplish the connection, one device should be the Master and the other the Slave. The Master is the one who will search the Slave.



Go to Bluetooth section on. After the Bluetooth is on, try



through NXT menu and turn it searching devices.

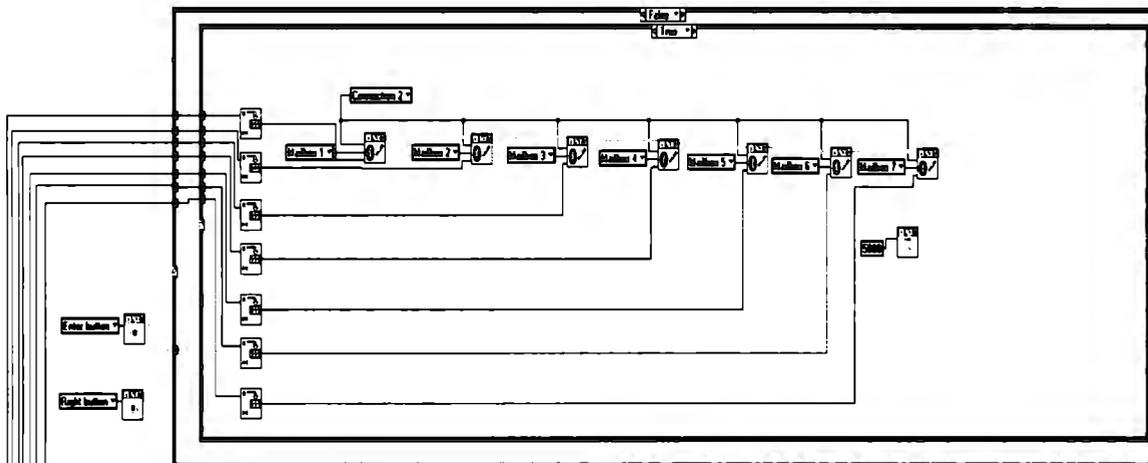


Select the NXT that you are looking for and select the connection for it.

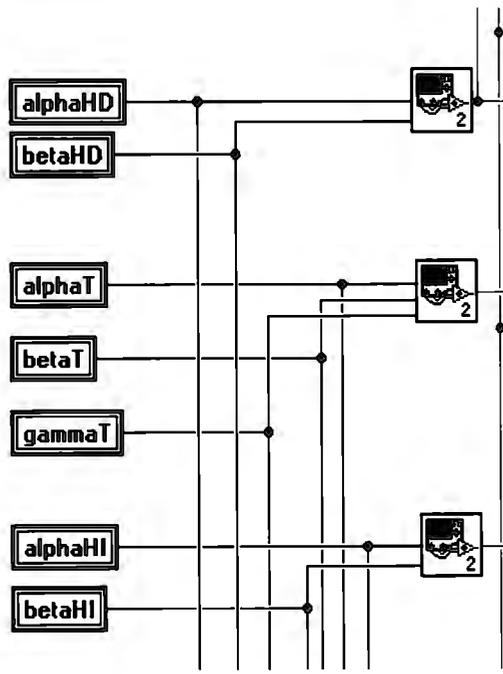
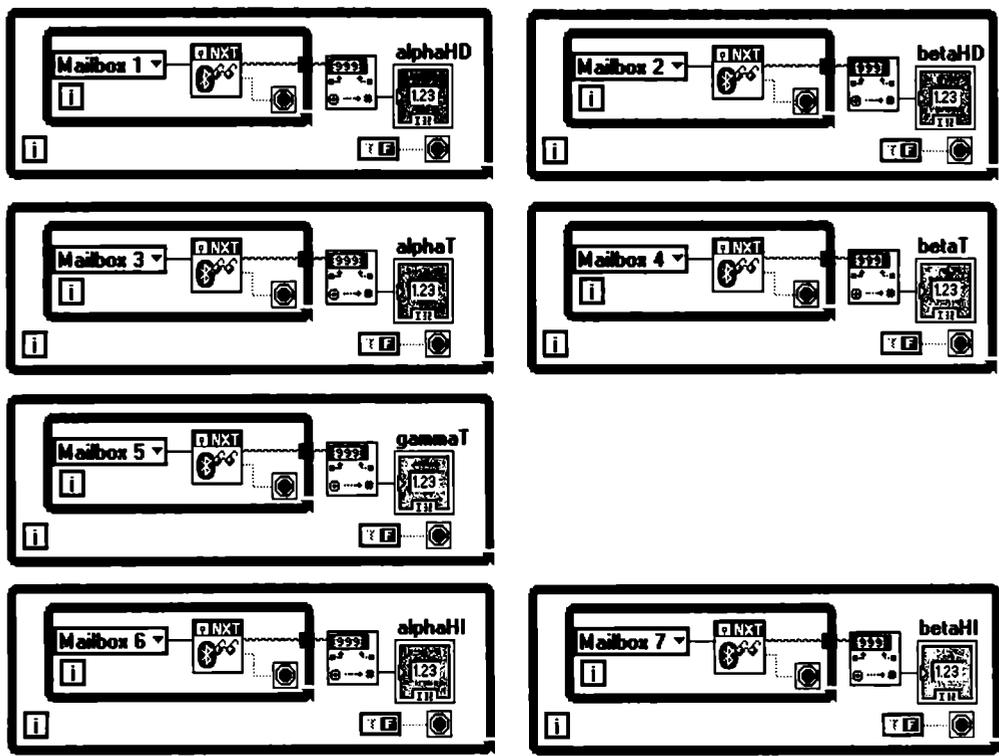


Now a Passkey will be asked for, but remember this is only for the first time. Type the passkey in both devices and the pairing issue will be done. The subsequent times you just have to look for its pals in the "My contacts".

Open either the VI "SpeedCtrlSndr(Graph006)" or "InverseCtrlSndr(Graph016)" as you desire and observe the code where Bluetooth programming is found. Note that the control has seven mailboxes used for this application, but recall that the NXT supports up to ten mailboxes. These mailboxes are destined to transfer the learning patterns from one robot to other. Compile and download the code to one NXT.



Open the either the VI "SpeeCtrlRcvr(Graph018)" or "InverseCtrlRcvr(Graph019)" and try to understand the code for the Bluetooth communication. Observe that this code the communication is "going in" rather than in the previous "going out". Compile and download the code to another NXT.



Put the robots at the entrance of the labyrinth and run the programs in both. Note that only the first code will be executed immediately after you pressed the button to run. Wait until the robot finishes the labyrinth and grab it then check the time elapsed and quickly press the set button and observe the other robot. Answer the questions.

What time does it take to the first robot to get to the end?

What time does it take to the second robot to get to the end? Is there any difference?

What was exactly the learning patterns transfer?

Objective: Develop a mini community to apply the concepts learnt in all the activities. Integrate the skills acquired to create a new global task and find out how to solve the new problem with your own expertise.

Material:

Robots (three or more)

Computers with LabVIEW installed

Procedure

1. Propose a brainstorm to collect data for the possible new global task.
2. Once the problem is defined, ask the advisor to write down the problem clearly.
3. Every student must contribute with her or his ideas.
4. Make the brainstorm for the problem solution.
5. Classify the ideas and organize them.
6. Establish the learning objectives according to the problem.
7. For this case the research will be done in the previous activities to isolate the concepts and knowledge needed for this next challenge.
8. Make the whole process of implementation and programming for each robot, as well as the pairing issue to create the mini community and develop the new global task.

VIDAVILIC

Esta obra se Imprimió y
Encuadernó en Vidavilic
55 54 43 09

E-mail: victor_vidavilic@yahoo.com.mx