A Semi-Open Learning Environment for Mobile Robotics

L. Enrique Sucar[1], Julieta Noguez[2], Gilberto Huesca[2] and Eric Rodríguez[3]
1 Instituto Nacional de Astrofísica, Óptica y Electrónica
2 Tecnológico de Monterrey, Campus Ciudad de México
3 Tecnológico de Monterrey, Campus Cuernavaca

***Abstract*—We have developed a semi-open learning environment for mobile robotics, to learn through free exploration, but with specific performance criteria that guides the learning process. The environment includes virtual and remote robotics laboratories, and an intelligent virtual assistant the guides the students using the labs. A series of experiments in the virtual and remote labs are designed to gradually learn the basics of mobile robotics. Each experiment considers exploration and performance aspects, which are evaluated by the virtual assistant, giving feedback to the user. The virtual laboratory has been incorporated to a course in mobile robotics and used by a group of students. A preliminary evaluation shows that the intelligent tutor combined with the virtual laboratory can improve the learning process.**

***Index Terms*—Tele-learning, virtual laboratories, intelligent tutors, remote labs**

I. Introduction

Practical experimentation in the laboratory is a key element in engineering education. This is a challenge for e-learning, where there is not direct access to laboratories. An alternative is the incorporation of *virtual laboratories*. In a virtual laboratory [10], a student interacts with simulated or remote equipment, performs experiments, and observes the results through visualization tools, cameras or other sensors. In this paper, we consider that a virtual lab could be a simulated model of certain apparatus, a real equipment that is accessed remotely or a combination of both. Generally, virtual labs are accessed via a web browser and internet.

Virtual laboratories assume that the student learns just by performing experiments and observing the results. However, this is not, in general, an effective and efficient strategy. It strongly depends on the learner ability to explore adequately and interpret the results of the experiments [7]. Many times the student repeats several times the experiment with unsatisfactory results, without detecting the cause of poor performance. An alternative is to integrate an intelligent tutor to the virtual laboratory, to guide and give help to the student, in a similar way as a laboratory assistant in a real lab. A tutor can help the student when the results of an experiment do not go well. It can give lessons to review the material that is critical for the experiment, and can select the type and level of difficulty of the experiments according to the experience and knowledge of the student. The tutor needs to deduce the student knowledge based only on the interactions when performing experiments. Thus, the students are evaluated indirectly based on the results of the experiments and the exploration behavior in the environment. Based only on the interaction with the virtual lab, the tutor has to deduce the state of the student, and decide the best pedagogical action. Given the uncertainty inherent in this task, a student model that can represent and reason with uncertainty is required.

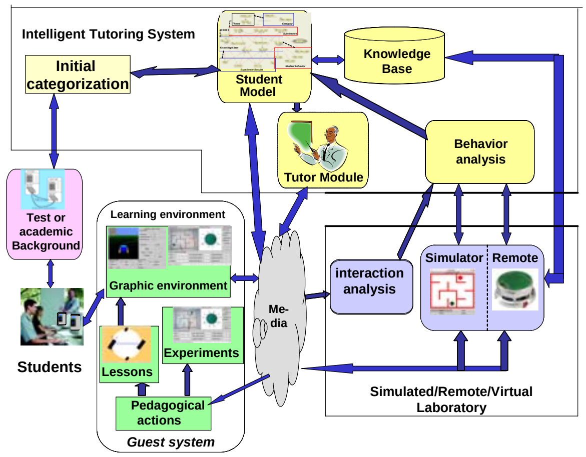

Figure 1. General architecture for virtual labs.

We have developed a semi-open learning environment for mobile robotics, to learn through free exploration, but with specific performance criteria that guides the learning process. The environment includes virtual and remote robotics laboratories and an intelligent virtual assistant the guides the students using the labs. The virtual and remote labs consider a specific task similar to robot competitions, in particular line following and a maze; and each one includes several experiments. The series of experiments in the virtual and remote labs are designed to gradually learn the basics of mobile robotics, from mechanical design and kinematics, to control and planning. Each experiment considers exploration and performance aspects, which are evaluated by the virtual assistant, giving feedback to the user. The key element of the intelligent assistant is a student model that uses a novel representation based on probabilistic relational models. The model keeps track of the students' knowledge at different levels of granularity, combining the performance and exploration behavior in several experiments, to decide the best way to guide the student in following experiments, and to re-categorize the students based on the results. The virtual laboratory has been incorporated to a course in mobile robotics and used by a group of students. A preliminary evaluation shows that the intelligent tutor combined with the virtual laboratory can improve the learning process.

II. The Learning Environment

A. Architecture

We considered aspects of open learning environments, because the student needs to explore different parameters to observe their effects inside the simulated lab, but each experiment has specific objectives the student needs to achieve, enabling an effective assessment of the exploration behavior and learning goals that guide the learning process. The learning environment compromises 3 main elements:

- A virtual (simulated or remote) laboratory.
- An intelligent asistant that guides the students in the use of the lab and gives personalized help and lessons on the relevant concepts.
- A user interface (guest system) where the student interacts with the lab and visualizes the experiments.

These elements have been integrated within a general architecture for virtual laboratories depicted in figure 1. Based on this architecture, we have developed the virtual labs for mobile robotics, and are currently developing virtual labs for other domains. Next we describe the user interface, and below the other two main elements.

B. User Interface

We defined a standard human-computer interface for all the experiments, composed of 5 windows, shown in fig. 2:

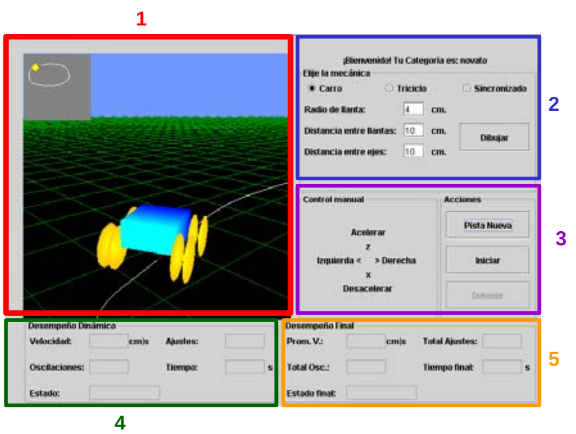

Figure 2. User interface. An example of the 5 elements in the user interface for an experiment in mobile robotics.

Experiment Visualization: the behavior of the experiment is displayed graphically, using 3-D graphics for simulated labs or video for remote labs.

Exploration Characteristics: the different options for available to configure the experiment are displayed in this area, and set by the student.

Interaction commands: the specific commands available, according to the experiments, are shown in this section.

Dynamic behavior: the results, in terms of exploration and performance are displayed during an experiment.

Final experiment performance: the final results of an experiment are depicted here. These are sent to the intelligent tutor at the end of each experiment to update the student model.

This interface allows the student to explore and control the simulated or remote environment, to observe the behavior of the equipment during the experiment, and to verify the results during an experiment (dynamic behavior) and at the end (final experiment performance). Through the interface, the tutor performs its *pedagogical actions*, which consist of: (i) *lessons* on a specific topic when required, and (ii) *experiments*, that is, defining the type and complexity of the next experiment suggested to the student.

III. The Intelligent Assistant

A. Intelligent Tutor

An intelligent tutoring system (ITS) tries to emulate a human tutor by adapting the learning experience according to the student. An ITS has usually 3 main components [8]:

Student model, which represents the student knowledge and preferences, to adapt the learning experience to suit the learner's perceived needs.

Tutor, which, based on the student model, guides the student in the learning process and provides personalized help

User interface, which provides the medium for interaction between the tutor and the student.

We have develop an ITS for virtual laboratories. The system is designed such that is can be easily adaptable for different experiments in several domains. Following the philosophy of virtual labs, the tutor is non-intrusive. That is, while a student is performing an experiment the tutor is transparent to the user, although it is following the experiment, considering exploration and performance criteria. At the end of an experiment, the tutor analysis the results in the context of the current student model, and does the following:

It displays to the student the results of the experiment, in terms of several parameters, related to the exploration and performance of the student in the experiment.

Based on the results and the student level, it provides, if necessary, lessons at the appropiate level of granularity of the relevent concepts that considers that the student could need to improve the results.

It updates the student model (see the next section).

It selects the difficulty of the next experiment (in the case of a simulated lab) based on the student model.

The key component of this tutor is a novel student representation based on probabilistic relational models, which we describe in the next section.

B. Student Model

The student model provides to the ITS knowledge about the each student, so its behavior adapts to the student needs. There are several ways to represent the student model [4], our representation is based on Bayesian networks [7], [9]. A Bayesian network is a directed acyclic graph, in which the nodes represent random variables and the arcs represent probabilistic dependencies, parameterized by the conditional probability of each variable given its parents in the graph. In the framework of Bayesian networks for student modeling, the random variables represent the student's knowledge of the different concepts related to the experiments, at different levels of granularity. For instance, in the robotics domain, we could have a node that represents the student's knowledge about *angular speed* (basic concept or knowledge item), another for *car configuration* (intermediate concept or sub-theme) and another for *kinematics* (high level concept or theme). The arcs in the graph represent the dependencies between these concepts, and their relationships with the information obtained from the experiments (performance and exploration results). Based on the values of the experiment result variables, the other variables are inferred using probabilistic inference [7]. We have extended this representation using probabilistic relational models. In this way we have a more flexible and efficient representation for student modeling for virtual laboratories. Next we present a general description of this model, for more details see [5], [6].

Probabilistic relational models (PRMs) [3] provide a new approach for student modeling, integrating the expressive power of Bayesian networks and the facilities of relational models. They allow the domain to be represented in terms of entities, their properties, and their relations. Based on probabilistic relational models, we designed a general structure for modeling students in virtual laboratory environments. The first step is to identify the classes in the model. The next step is to define the dependency model at class level, allowing it to be used for any object in the class and facilitating the understanding of the model. The classes and their relations provide a general *schema* for the student model. Based on this schema, a *skeleton* is derived, which integrates in a Bayesian network the relevant variables and their dependencies. Finally, a particular Bayesian network is obtained from the skeleton for each specific experiment in certain domain.

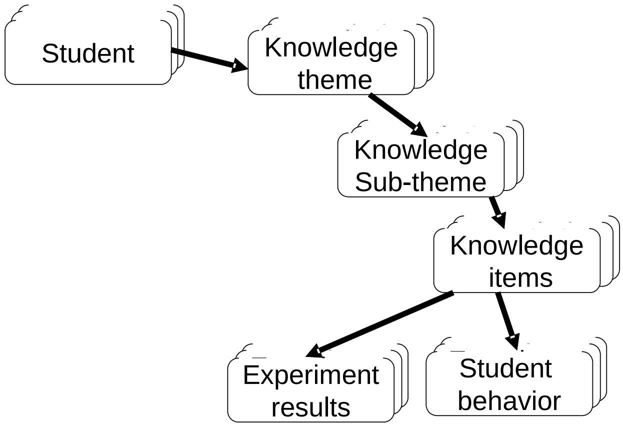

Figure 3. A general schema for the student model for virtual labs.

A general schema for the student model is depicted in figure 3. This model integrates behavior analysis (experiment results, student behavior), with the representation of the student's knowledge at different abstraction levels (items, sub-themes, themes and student category). For each class, a number of attributes (information variables and random variables) is defined. Once the model is specified at the class level, including the attributes and their dependencies, we can extract a skeleton, that is, a general Bayesian network model for a fragment of the model. A skeleton obtained from the model in figure 3 is depicted in figure 4. Based on this skeleton, particular instances (Bayesian networks) are defined for each experiment. In the case of the virtual robotics laboratories, we have 4 experiments, although two of them are similar, so we generated 3 different Bayesian network. The main advantage of the PRM student model is that, once a general schema is defined, it is relatively easy to specify the models for different experiments and even different domains, reducing the time and cost of developing intelligent tutors for different applications.

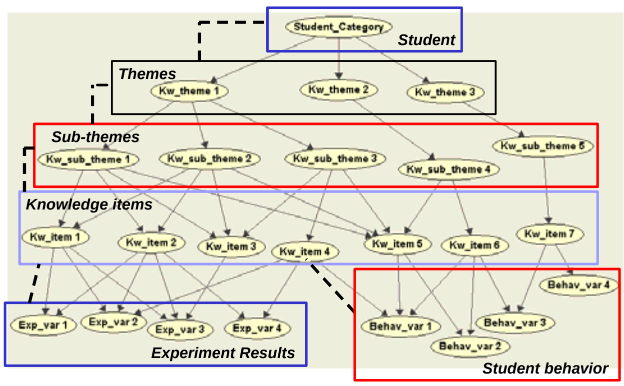

Figure 4. A general skeleton obtained from the schema in figure 2. This skeleton specifies a general model for any experiment, which is later instantiated to a particular experiment.

After each experiment, the PRM student model is updated based on the results of the experiment. Based on the evidence form exploration and performance variables, the relevant knowledge items are updated using probability propagation, and from these upwards to the sub-themes and themes. When the student selects another experiment, he/she is re-categorized considering the previous results and the new knowledge items related with this new experiment.

The results are used by the *tutor module* to decide the best pedagogical action, such as help or lessons. Based on the values (probabilities) of the relevant nodes in the student model (knowledge at different levels of granularity), a set of rules is used to decide the best pedagogical action. The possible actions are: (i) present another experiment with higher difficulty level, (ii) present another experiment with lower difficulty level, (iii) present another experiment with the same difficulty level, (iv) give help specific for the experiment, (v) display a lesson, that could be at different levels of granularity –concept, sub-theme or theme, (vi) apply a quiz, and (vii) present another experiment of different type.

IV. VIRTUAL ROBOTICS LABORATORIES

We have developed a learning environment for mobile robotics that incorporates several virtual labs. The environment considers simulated and remote labs, for individual and collaborative learning. The different labs are designed to support different stages of the learning process. Currently, we have the following labs:

- Simulated robotics laboratory
- Remote robotics laboratory
- Distributed competition

These laboratories are integrated with the intelligent assistant and have been incorporated to a basic course in mobile robotics [6].

A. Simulated Lab

The *simulated robotics laboratory* is a simulated lab for individual learning. It is designed so that the students can explore the aspects related to the design of a mobile robot: mechanical configuration, kinematics and sensors, as well as for experimenting basic control algorithms and programming. It includes 4 experiments:

- *Experiment 1*. Mechanical design and kinematics: explore different types of robots and physical dimensions.
- *Experiments 2 & 3*. Sensors: explore the use of infrared sensors under different configurations.
- *Experiment 4*. Control: learn about control algorithms and programming.

All the experiments in the virtual robotics lab consider a line following competition, in which the simulated robot must follow a circuit (white line over a dark background) as fast and as close as possible. Each experiment includes several parameters that the students can explore (type of robot, dimensions, position of the sensors, etc.), and at the same time performance goals (time to complete the circuit, minimize oscillations, etc.). The user interface for experiment 1 is shown in figure 2.

B. Remote Lab

The remote lab for mobile robotics allows remote experimentation by programming a real robot via Internet. The objective is that students are aware of the differences from simulation to the real world, in particular the uncertainties inherent in the controls and sensors in real robots. They also practice more advanced concepts in robotics, including trajectory planning and search. The lab is based on a *Kheppera II* mobile robot. In the lab a student has to program the robot to solve a maze. The student writes a *Java* program and then sends it to the server via the laboratory web page. The program is the executed by the robot, and the student can observe the experiment in a video obtained from a ceiling mounted camera, see figure 5.

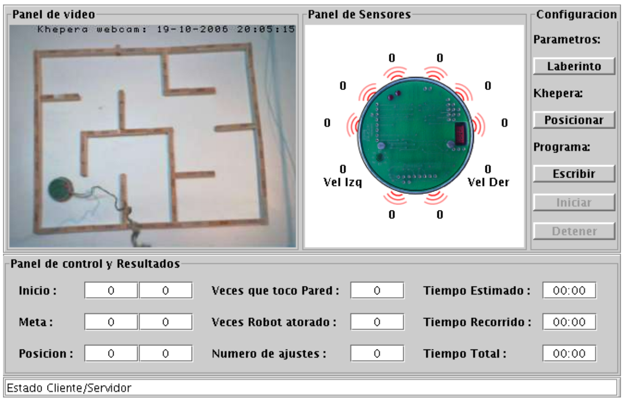

Figure 5. Remote lab interface. It shows a top view of the experiment (top left), the sensors (top middle), the commands (top right), and the dynamic variables (bottom).

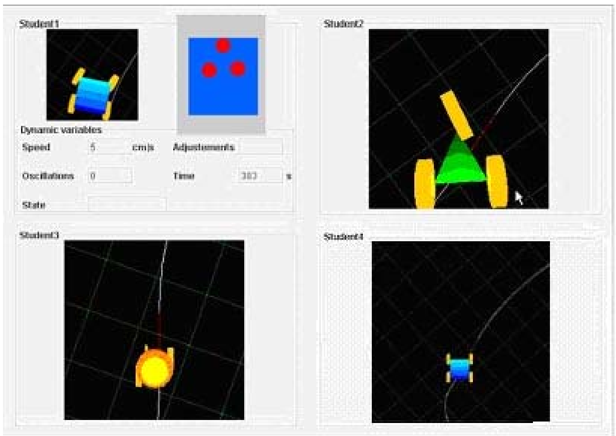

Figure 6. The contest interface as seen by one of the participating students. The participant observes the dynamic variables associated to her robot (top left) and only the simulated robot for the others.

C. Distributed Competition

The distributed competition lab allows students to develop their robot in a shared environment, where they can compete with other students [9]. The student needs to write her control program previously, taking care of the mechanical and sensor aspects which were explored previously in the simulated robotics laboratory. This laboratory hereby acts as a contest scenario, and thus allows the student to learn by watching other people's work. For this experiment, the instructors setups a line following competition, in which up to 4 students can participate, and others can watch. Each student can observe the 4 robots in the virtual circuits, and the performance parameters for his robot. The interface as seen by one of the students is depicted in figure 6. To participate in a contest, the students need to load his/her automatic control program. The system verifies its syntax, if there are no errors, at contest time the system shows the robot movements based on the control program of each participant.

D. *Evaluation and Results*

To evaluate the learning environment, we have incorporated this environment to an undergraduate course in mobile robotics [6]. The virtual labs are used in the first part of the course, to practice the basic concepts in mechanical design, sensors, control and programming. In two occasions, we have used these labs on a formal course at *Tecnológico de Monterrey* and evaluated the impact of the learning environment. In the first case we evaluated the simulated lab, in particular the incorporation of the intelligent tutor. In the second case we evaluated the distributed competition, with emphasis on the impact of the competition on learning. In both cases we did a controlled experiment. The class was divided into two groups, control and experimental groups. After experimenting in the lab we measured the learning gain based on a post-test in the concepts related to the labs. We then compared the results of both groups in terms of the post-test grades. Next we present the results of both experiments.

E. *Experiment 1: SimulatedLlab*

In this case, the *control group* experimented in the virtual laboratory without the tutor, while the *test group* had the advice of the intelligent tutor. Both groups had the same time period to experiment in the lab, and in both cases, lessons were available in the laboratory's web page. A pre-test was applied to all students in the concepts related with the experiments. After 2 weeks of experimentation in the virtual lab, a post-test was applied in the same topics. Figure 7 summarizes the results of the post-test for both groups, the pre-test results are also shown for comparison. The graphs of the pre-test, with tutor and without tutor represent the students' grades in the post-test, in ascending order per student. The results show that the students that practice in the virtual environment with the help of the tutor have a better performance. This gives empirical evidence in favor of the need of a learning environment that integrates a tutor to guide the students in the virtual laboratory.

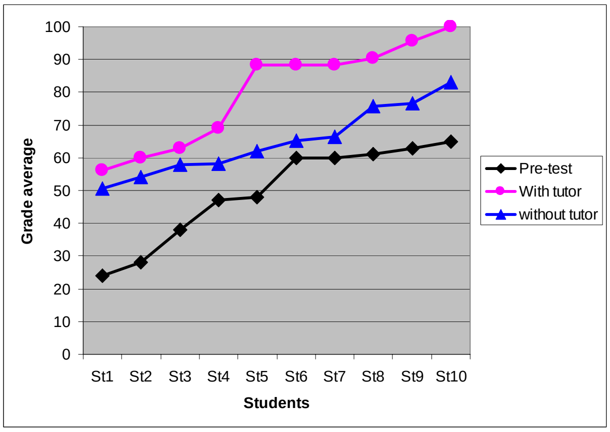

Figure 7. Post-test results for the simulated lab, comparing the scores of the experimental (with tutor) and control groups (without tutor). It shows also the pre-test for comparison.

F. *Experiment 2: Competition*

A total of 20 subjects enrolled in a robotics basic course participated in this experiment. We divided them randomly into an experimental group that used the distributed environment participating in several competitions; and a control group that only practiced with the individual experiments in the virtual laboratory. The experimental group had the opportunity to participate in several contest, while the control group only observed the contests. After the experiments, a post-test was applied to all the subjects, It consisted of 10 questions related with the knowledge objects practiced in the experiments, and of a ten item questionnaire targeted at students' opinions about their virtual laboratory experience. Figure 8 summarizes the results for the control and experimental groups. It shows the grades in the post test per group, in ascending order per student. We observe a significant improvement in learning for the students that participated in the contests with respect to those that did not.

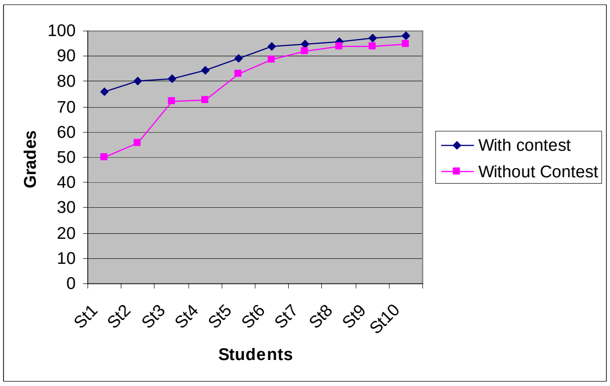

Figure 8. Post-test results for the competition, comparing the scores of the experimental (with contest) and control groups (without contest). The grades are in ascending order per student.

V. Conclusions and Future Work

We have developed a semi-open learning environment for mobile robotics, to learn through free exploration, but with specific performance criteria that guides the learning process. The environment includes virtual and remote robotics laboratories and an intelligent virtual assistant the guides the students using the labs. We have incorporated this environment to an undergraduate course in mobile robotics, and have performed some control experiments with two groups of students. The results give evidence of the impact of the tutor and the competitions on learning. We are currently designing an online course on mobile robotics, to which we are incorporating this learning environment. In the future we want to evaluate the impact on an e-learning scenario, considering all the labs. We are also working on extending these ideas to other domains, such as physics and medicine.

References

[1] Bunt A. and Conati C.. "Probabilistic Student Modeling to

Improve Exploratory Behavior". *Journal of User Modeling and User-Adapted Interaction* 13 (3), 2003, pp. 269-309.

[2] Huesca G. "Laboratorio Virtual de robótica móvil en esquemas de coordinación concurrente". Master thesis. Tecnológico de Monterrey, Campus Cuernavaca. 2006. (In Spanish).

[3] Koller. D. "Probabilistic Relational Model". *9th International Workshop Inductive Logic Programming* 1999. Saso Ozevosky & Peter Flach (Eds). Springer Verlag. 1999, pp. 3-13.

[4] Mayo M., Mitrovic A. "Optimizing ITS behavior with Bayesian networks and decision theory". *International Journal of Artificial Intelligence in Education*. Vol. 12, Number 2, 2001, pp 124-153.

[5] Noguez J. Sucar E. "A probabilistic relational student model". *VI Sixth Mexican International on Computer Science. ENC05. Computer Society/IEEE.* Puebla, México. September 2005, pp. 2-9.

[6] Noguez J. Sucar. L.E. "Intelligent Virtual Laboratory and Project Oriented Learning for Teaching Mobile Robotics" .*International Journal in Engineering Education.* Special Issue on "Robotics Education". Tempus Publications 44 (4), 2006, pp. 743-757.

[7] Pearl J. "Probabilistic Reasoning in Intelligent Systems". *Morgan Kauffman*. San Mateo California. 1988.

[8] Self J. "Formal approach to student modeling**".** In J.E. Greer and G.I. McCalla (eds.). *Student Modelling: The Key to Individualized Knowledge-Based Instruction*. Berlin. Springer Verlag, 1994, pp. 295-352.

[9] VanLehn K., "Bayesian student modeling, user interfaces and feedback: a sensitivy analysis". *International Journal of Artificial Intelligence in Education*. 2001, pp. 154-184.

[10] Wagner B. "*F*rom Computer Based Teaching to Virtual Laboratories in Automatic Control**".** *29th ASEE/IEEE Frontiers in Education Conference.* Session 13d6-6. San Juan Puerto Rico, 1999.

AUTHORS

Dr. L. Enrique Sucar, INAOE, Computer Science, Luis Enrique Erro #1, Tonantzintla, Puebla, México, esucar@inaoep.mx

Dr. Julieta Noguez, Tecnológico de Monterrey, Ciudad de México, Puente No. 222, Col. Ejidos de Huipulco, México D.F., México, jnoguez@itesm.mx

M.Sc. Gilberto Huesca, Tecnológico de Monterrey, Campus Ciudad de México, Puente No. 222, Col. Ejidos de Huipulco, México D.F., México

Mr. Eric Rodríguez, Tecnológico de Monterrey, Campus Cuernavaca, Reforma No. 182-A, Temixco, Morelos, México

Manuscript received May 2nd, 2007.