Remote Robotics Laboratory as Support to Teaching Programming

Thais Oliveira Almeida
Department of Science Computer
Federal University of Roraima
Boa Vista, RR, Brazil
Institute of Computing (Icomp)
Federal University of Amazonas
Manaus, AM, Brazil
thais.oliveira@ufrr.br

José Francisco de M. Netto
Institute of Computing (Icomp)
Federal University of Amazonas
Manaus, AM, Brazil
jnetto@icomp.ufam.edu.br

Marcel Leite Rios
Research Group in Technology Solutions (GoTec)
Federal Institute of Rondônia (IFRO)
Porto Velho, RO, Brazil
marcel.rios@ifro.edu.br

***Abstract*—This paper presents the development of a Robotics and Programming Laboratory with the support of Multiagent Systems (MAS) to help users learn programming skills using robots. The developed laboratory is available in a Learning Management System (LMS) allowing people from geographically distant locations to gain experience with experiments of this nature. The choice of Robotics Education (RE) as object of scientific experimentation linked to teaching and learning systems has shown to bring significant improvement in the learning process. The system makes available on the Internet a space where it is possible to practice programming using Lego Mindstorms robots in the Python programming language. The system suggests several challenges based on the score and profile of users. Its features were tested with promising results, where users have felt very interested and motivated to continue learning programming and robotics. The work developed presents advances over remote robotics laboratories found in the literature, since it allows the robots to be remotely programmed through the programming language NXT-Python, and not only by using command arrows.**

Keywords— Multiagent System; Learning Programming; Remote Laboratory; Robotic Education.

I. Introduction

Difficulties in learning introductory programming disciplines are constant as theme for various discussions and research, resulting in many proposals for methodologies and support tools. In addition to Computer Science students, Engineering students, and students of related areas usually take courses of this nature, but find it difficult to assimilate its contents and complete them successfully.

To be successful in introductory programming courses, students must learn a set of basic concepts, such as the syntax of a programming language, and to develop problem-solving skills. It is critical to engage students while teaching them in order to increase their enthusiasm for learning new concepts. Educational researchers have concluded that active engagement in learning is an important tool to improve students' comprehension [1].

The purpose of this work is to improve STEM (Science, Technology, Engineering and Math) educational programs for scholars and students alike, using robotics as the motivating element for learning. Robotic Education (RE) has been used as a learning technique allowing to perform activities, and as a tooto stimulate students' creativity given its dynamic, interactive and even playful nature.

Nowadays, it is easy to find a vast variety of robotic platforms, although costs are the greatest obstacle to most educational institutions who wish to include robotic platforms as a part of their educational programs [2]. The amounts charged for these environments constitute the main deterrent to the widespread use in schools, especially in developing countries. The same happens with didactic laboratories, due to its high installation cost.

With the advancements in technology, several remote laboratories have been developed in order to allow experiments with robots, like [3]-[6]. These works use the structure of a physical lab and make it available on the Internet to allow even geographically distant people to experiment with robots.

The Remote Robotic Laboratory arises from an investigation carried out in the context of existing remote laboratories. Additionally, the mentioned laboratories were analyzed in order to identify contributions and limitations for each one of them, thus serving as motivating element.

It is worth noting that this proposal is not intended to replace existing physical laboratories, but rather to allow remote access to real equipment. In this sense, the research question of this work is: How does a remote robotics laboratory motivate users to learn programming, from experiences that allow them to assimilate theoretical content in practice?

This work aims to develop and evaluate a Multiagent System (MAS) which helps improve motivation of users in the learning process of programming with the use of robots, in a remote robotics laboratory that provides physical equipment for conducting experiments. To report the research, this article is structured as follows: Section 2 describes the related works, Section 3 presents the work description, Section 4 describes the case study, Section 5 offers discussions and conclusions.

II. Related Works

In order to evaluate the state of research concerning this work, this section presents some research related to RE and

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

remote laboratories that allow manipulating or programming robots at a distance.

The work developed by [3] features a remote robotics laboratory that allows students to define a complete controller for a mobile robot. Thus, making it possible to interact with both the environment and other robots present in the arena (possibly conducted by different users). Before working with real robots, users can test their algorithms in a virtual environment where test algorithms are replicated.

[4] introduce SyRoTek, an e-learning platform for mobile robotics, artificial intelligence, and control engineering. This environment provides and manages a set of autonomous mobile robots in an arena, so users can run the experiments. Robots can be controlled remotely via ActiveX technology, or by a program written in C ++, Delphi or Java.

A virtual, remote robotics laboratory is presented in [5]. The authors use EJS (Easy Java Simulation), Matlab, and LabView technologies for laboratory development. The application allows you to work with simulated robots (virtual lab) and real robots (remote lab). Students can create algorithms to control the robot, simulate it in the virtual lab and deploy it in the remote lab for testing purposes.

[6] present a remote access laboratory for experimentation with a team of mobile robots. The robots are built with Lego Mindstorms technology, and users explore the Matlab environment for the development of robot team control strategies. The lab can be accessed through standard Web browsers and a graphical interface that allows you to select the type of experiment you want to perform.

All the works presented in this Section have a significant recognition in the academic environment and served as motivation for the development of this research. In this article we present the integration of the MAS with a Learning Management System (LMS), so that users have the help of Intelligent Agents (IAs) to program the robots remotely in a programming language, which is what distinguishes this work from others.

III. WORK DESCRIPTION

This section is intended to provide an overview of the system, operation, and tools used in the work carried out. In this sense, Subsection A presents an overview of the developed laboratory, explaining the system architecture and a brief introduction to the agents. Subsection B addresses the scheduling process. In Subsection C, the challenges room and the remote manipulation process are presented.

This laboratory provides a space, on the Internet, where it is possible to perform the programming practice using Lego Mindstorms robots in the Python programming language, mediated by MAS technology. The strategy adopted to reach the objective of the work, covered steps such as the initial specification of the AIs using the Unified Modeling Language (UML), and the Prometheus methodology in the creation of the graphic models and the implementation of the laboratory.

A. System Overview

The laboratory is available in a LMS, and can be accessed via Web, allowing even geographically distant people to gain experience in programming, using robots. In this environment, the students have challenges proposed to be overcome with the support of a MAS to assist in their realization. In this way, users who have access to computer labs, but do not contain robotic kits, can enjoy experiences of this nature.

To access the robotic experiments, it is necessary that the user is registered in the system. In order to perform the experiments, a prior scheduling is also necessary in view of the availability of robotic kits for use. The programming of the robots must be performed in the programming language NXT-Python (which allows to control the Lego NXT robots, using Python programming language).

Communication with the robot can be done via USB or Bluetooth. As we use Bluetooth technology in this laboratory, it was also necessary to install Pybluez on the server for the correct functioning of the system, a Python module that allows access to Bluetooth resources. The MAS offers a helpdesk for user registration, confirming access to the virtual room, making reservations, scheduling experiments, and assisting in the remote programming of the robot. Figure 1 shows the architecture of the laboratory.

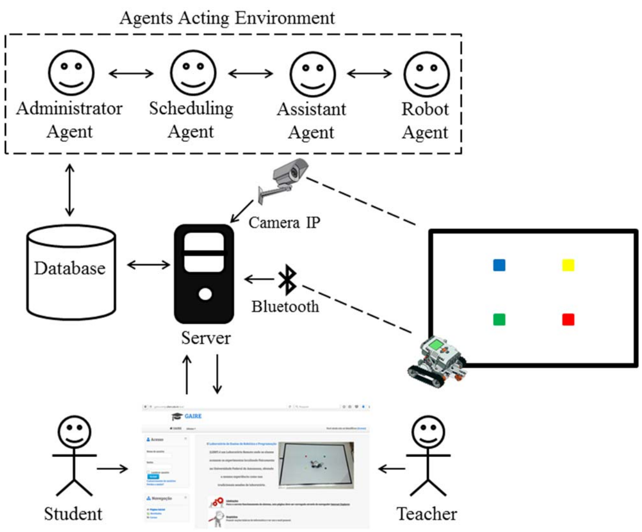


Fig. 1. Architecture of laboratory

The system architecture consists of four agents, each with its function, namely:

- Administrator Agent: responsible for releasing access to the AVA and storing user information.
- Scheduling Agent: responsible for managing the schedule for conducting experiments in the laboratory.
- Assistant Agent: Assists users in the process of robot programming. Suggests challenges based on the history of each one, which helps to estimate the user's learning environment.

- Robot Agent: responsible for executing the robot movements as programmed by the user, and presenting it in the LMS. It also performs the user-generated syntax analysis of the source code before submitting it to the server.

The agents act on the environment by exchanging messages while the user performs the experiments, as well as offering support regarding the programming contents. Figure 1 also shows the tools used in the communication process between the user and the robotic kit. To that end, the Moodle LMS was used as the graphical interface of the system. The MySQL database, the web server, the IP camera that captures the physical lab images, the Lego Mindstorms Robotic Kit, and a USB 2.0 Bluetooth connector were also used. The Jade framework and the Java programming language were used in the development of the agents.

B. Scheduling process

For access to the virtual room and robotic experiments, it is necessary for the user to register. User information is stored in the laboratory database, which uses the Moodle table structure, as well as some additional tables. The laboratory is available on the Internet and can be accessed by any user at <http://gaire.icomp.ufam.edu.br/ava>.

It was necessary to adopt a scheduling policy in the system in view of the quantity of robotic kits available for use. In this specific case, a user can have access to the robotic experiments according to the scheduled time for a specific day, as long as the available schedules per day, in Table I, are observed.

TABLE I. AVAILABLE SCHEDULES PER DAY

Schedules	
1	09:00 a.m. – 09:40 a.m.
2	10:00 a.m. – 10:40 a.m.
3	11:00 a.m. – 11:40 a.m.
4	02:00 p.m. – 02:40 p.m.
5	03:00 p.m. – 03:40 p.m.
6	04:00 p.m. – 04:40 p.m.
7	05:00 p.m. – 05:40 p.m.

It is not possible to schedule for past dates and times, nor for times already reserved by other users (the Scheduling Agent is responsible for resolving these conflicts).

It is the responsibility of the Scheduling Agent to attempt to make all reservations requested by users. The Scheduling Agent operates in the environment checking for new reservation requests, and for available scheduling vacancies. After confirmation of the schedule, an email is sent to the user, allowing him to follow up the requested reservations.

C. Challenges room

The developed laboratory enables users to conduct experiments with the Lego Mindstorms robotic kit. In this sense, students can program the robots through the NXT-Python programming language, instead of only manipulating it through control arrows. The works presented in Section 2 of this paper collaborated and motivated its development.

The system has five challenges to be achieved by users, and can be accessed after scheduling through the links available in the virtual room (Figure 2).

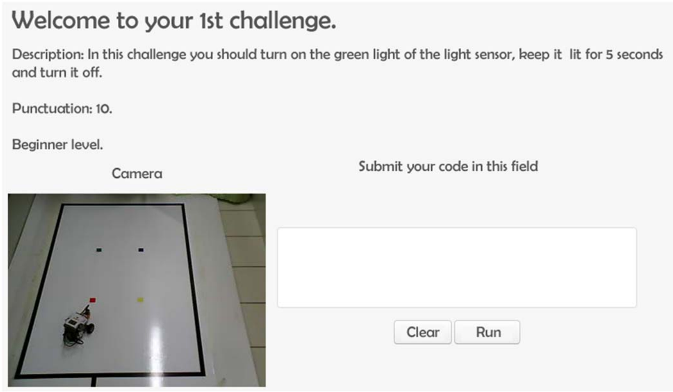


Fig. 2. Challenges room

The length of stay of the user in each challenge is in accordance with what was stipulated in Table I (a maximum of forty minutes). It is important to note that by loading a challenge and completing it successfully, a score is assigned to the user.

The assigning of the user profile is related to their performance in the room of challenges when programming the Lego NXT robot. It should be noted that all user actions in the laboratory are stored in specific DB tables. It is the Professor's duty to accompany the students during the accomplishment of the challenges, and upon completion of the same, "approve" or "not approve it."

The server and the robot are connected through Bluetooth technology. After the program is sent to the server by the user, the Robot agent looks for the code in the database, generates the executable file and performs the syntactic analysis of the code, checking for errors. If there is no syntactic error, the running program, sent by the laboratory, controls the robot remotely.

Figure 3 shows the layer structure that was developed to allow integration of the technologies.

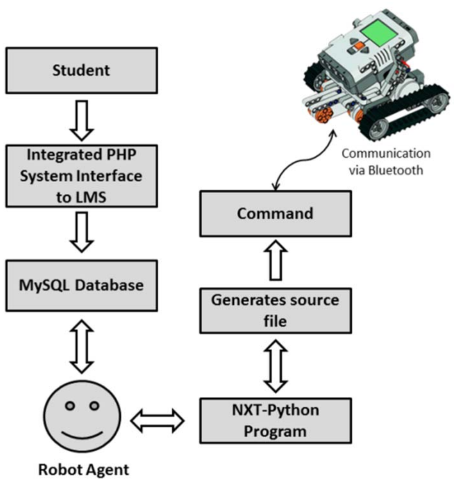


Fig. 3. Remote manipulation process

By default, Moodle uses the layer structure to store all occurrences in the DB. The user enters the program written in the NXT-Python language from the graphical interface which stores this data in the DB through scripts developed in the PHP language.

The MAS running on the server was developed in JADE and requires the Java language to run. The developed agents are responsible for assisting in the registration of users, scheduling of experiments and manipulation of the robot in the system.

The three lines below should be included at the beginning of a program written in NXT-Python, so that the remote manipulation process works correctly. The ID variable is used to control the robot. The object is started from the class defined as BlueSock, as follows:

```
1. from nxt.blue sock import BlueSock
2. from funcoes import *
3. ID = '00:16:53:11:25:73'
```

The first line imports the "BlueSock" class from the "NXT Blue-Sock" module. This class must be used for commands to be sent correctly through Bluetooth technology, which uses a 48-bit MAC address. The second line imports the functions that will assist users in meeting the challenges. To use them, it is only necessary that the user invoke them in his program by stating the name of the program followed by the arguments. The third line generates the object "ID", where the parameter '00:16:53:11:25:73' is an example of the MAC address of the robot. As the "nxt.bluesock" class hides how Bluetooth communication is performed and variables are updated, users can develop their programs unaware of the internal details.

The robot can use the light sensor to light a color (red, blue or green). It can also be moved forward, right, or left according to the functions available to users. The speed at which the robot will walk, as well as the angle that it should follow to turn, are also controlled by the program. Table II lists the predefined functions that users can use in their programs to meet the challenges.

TABLE II. PREDEFINED FUNCTIONS

Functions	Meaning
light(brick, string_color)	Illuminates light sensor.
turnoff_light(brick)	Turns off light sensor.
waiting_time(time)	Causes the robot to wait for a while in seconds, until a new action is taken.
run_for_time(brick, velocity, time)	Travels a distance based on the given speed time.
spins(brick, velocity, number_rotation)	Rotates the robot.

IV. CASE STUDY

A. Project description

This section presents the tests performed with the system and the results obtained. For the development of the proposed MAS, we used a Desktop computer that worked as a Web server, a Lego Mindstorms robotic kit, a rechargeable battery for the Lego RCX, a USB 2.0 Bluetooth connector and an IP camera.

The Web server uses the Ubuntu operating system, with Apache server version 2.4.7, MySQL database version 5.5.46 and PHPMyAdmin version 4.0.10. The LMS Moodle version 2.8 was used as the graphical interface of the laboratory. The JADE framework version 4.2.2 and the Java programming language were used in the development of the agents.

The approach developed in this project is intended for use by people with little or no computer programming skills, where just having a personal computer with Internet access and basic computer abilities will be enough.

Several tests were performed to detect the presence of system failures. The analysis of the records stored in the database and the feedback from the users involved in the research collaborated for the correction of these faults. The implementation approach used was prototyping. With each new feature added to the system, specific tests were performed to improve software quality.

B. Test in a real class

Tests were carried out with students from the College of Application of the Federal University of Roraima (CAP/UFRR), which showed that the use of the laboratory to motivate users to learn programming and robotics appears to be very promising.

Initially it was necessary to contact the pedagogical coordinator of the chosen school and give an explanation of the project and its objectives. The coordinator then requested parents to sign a document authorizing their children to take the tests outside of school hours.

Several students were interested in taking the tests. In view of the limited space and time, we selected 20 (twenty) middle school students from 7th and 8th grades. At first, a brief explanation about the laboratory was given to show the environment and how the remote manipulation process occurs. On this day, it was also requested that they register to access the system.

All the students were able to register on the system without much difficulty. After registering, the students were instructed to schedule the experiments (each student scheduled a day and a time).

1) User profile

A questionnaire was made available in the virtual room, to characterize the users participating in the laboratory tests. The importance of characterizing the user lies in the fact that it establishes a more critical look at the result of the questionnaire used by the user to analyze the work developed. The criterion chosen to summarize the questionnaire was that of general applicability, in other words, no specific questions were included in this questionnaire.

Of the interviewees, 62% were men and 38% women. In relation to the age group, 90% were 12 to 14 years old, and 10% were 15 to 17 years old. As for the level of schooling of the interviewees, all of them had incomplete middle school education. Figure 4 shows the graphs corresponding to these questions.

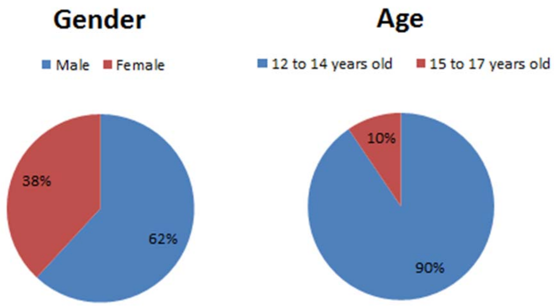


Fig. 4. User profile

Respondents were asked about computer use. 100% said they had used it before. Of these, 71% had used it a lot and 29% had not used it as much. No user reported having used it just a little. When asked about the knowledge they had in computing, 57% rated it as regular, 24% said they had good knowledge, and 19% rated it as optimal. Figure 5 shows the graphs corresponding to these questions.

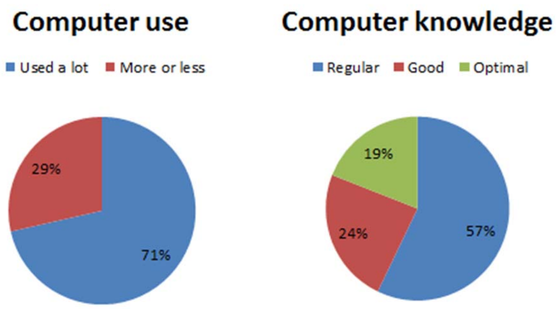


Fig. 5. Computer knowledge

Overall, since all students had basic computer knowledge and knew how to surf the Internet, it leads to the belief that they would not have much difficulty accessing the remote laboratory and conducting experiments.

2) Experiments performed

Although this is not a desirable situation, some students who agreed to participate in the tests did not show up on the scheduled day and time. In a conversation with the pedagogical coordinator, she informed they were writing tests that week, and as a result some chose to give up, since participating in the tests would not influence their grades. Of the 20 (twenty) students who had enrolled in the system, 14 (fourteen) attended the tests. Table III shows the relation of the number of participating students, with the percentage of the challenges they completed within the timeframe for the experiment.

TABLE III. CHALLENGES COMPLETED IN EACH EXPERIMENT

Number of students	Challenges Completed	Percentage
1	2	40%
2	3	60%
7	4	80%
4	5	100%

Each experiment had a maximum duration of 40 (forty) minutes. As shown in Table II, 4 (four) students completed the 5 (five) challenges available in the virtual room within the stipulated time. 7 (seven) students completed 4 (four) challenges in time. Two (2) students were only able to complete three (3) challenges within 40 (forty) minutes, as the Internet was very slow and hampered the outcome. Only 1 (one) student was able to complete just 2 (two) challenges within 40 (forty) minutes, as he arrived 25 minutes late to take the tests.

In general, the students did not have difficulties in complete the challenges, considering the explanation given about the system and the operation during the first contact with them, as well as how the functions that were available for usage to complete the challenges were presented.

3) Results obtained

At the end of the experiments, an opinion poll was offered in the virtual room, so that users could evaluate the system after having performed the tests. This questionnaire aimed at analyzing the impression of the users after having experienced the programming of robots in the remote laboratory. In this sense, it was tried to establish if the proposed methodology helped as a motivating element in the learning of programming.

The first questions in the questionnaire sought to assess the effectiveness of the Lego Mindstorms kit to support programming learning. When asked about learning with relation to programming in the laboratory 43% of users said that it was very easy, and 57% thought it was facilitated. When asked about learning with relation to the content presented, 71% of users rated it as great, and 29% as excellent. The graphs showing the result of these questions are shown below in Figure 6.

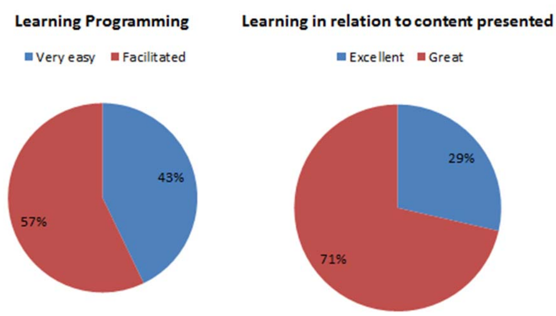
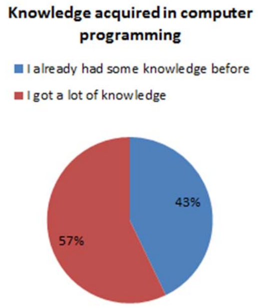
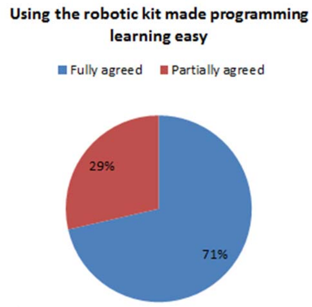


Fig. 6. Lego Mindstorms kit do support programming language

On this occasion, we tried to evaluate the methodology used for learning, where 100% of the users evaluated as very important. Regarding the knowledge acquired in computer programming, 43% said they had some previous knowledge, and 57% had great knowledge. z When asked whether using the robotic kit made learning computer programming easy, 71% of the users fully agreed, and 29% partially agreed. Figure 7 shows the graphs with the results of these questions.





When asked if the use of robots made learning programming fun, 86% of users agreed fully, and only 14% agreed partially. All users felt motivated and interested in learning more about programming using robots.

The results obtained with the tests performed were satisfactory and promising, because in the end we realized the proposed system facilitated the learning process in programming. In addition, the use of robots served as a motivating element and made learning fun.

The results of the tests carried out in this article show that the system is feasible to be used by anyone who has an interest in learning programming and robotics, has basic computer skills and knows how to surf the Web. The only limitation to accessing the challenge room correctly is that the lab page must be accessed through the Internet Explorer browser, for the correct display of the camera image using the ActiveX plugin.

Overall, users were interested and motivated to learn more about programming using robots and to accomplish more challenges. They also asked whether they could schedule other experiments and practice what they had learned, and whether there would be other challenges available.

V. DISCUSSION AND CONCLUSION

This article presented a programming teaching approach supported by Multiagent Systems that served as a motivating element for learning with the use of robots in a remote laboratory, giving users the opportunity to have contact with experiments of this nature.

The proposed approach proved adequate to motivate users with basic programming knowledge, and even those who did not have any knowledge. Our purpose through this research was to investigate whether students would be motivated to learn programming and robotics.

Even with the reduced number of students attending the tests, the results were satisfactory because the system was able to spark interest and motivate them to learn programming and robotics. Activities were made available and developed by levels of difficulty. This way, the students felt motivated to overcome the new challenges that were controlled and monitored by MAS.

It was necessary to get them familiarized with the technologies related to computers and robots, to give them support on the functioning of the system, to explain the contents and the methodology to be used, among other features related to people with no knowledge in this area. In the tests performed, we became aware of improvements and adjustments that can be made to optimize the system, with the purpose of improving the learning process.

As in the development of the prototype the laboratory interface used the LMS Moodle, its tools were not included in the tests, since there was already a great deal of information about them in the available scientific literature. The tests explored the registration of users in the environment, confirmation of access to the virtual room, reservation, scheduling of experiments, release and control of the section in the challenges room, and remote programming of the robot.

The approach presents advances in robotics remote laboratories found in the literature, because it allows for the selection of different challenges, programming the robots in a programming language and uses a user/system interaction with the support of the Intelligent Agents. The developed system is functional, and with small adaptations can be extended to accept other programming languages. In the same way, it is possible to modify it and make it available on a Web page, making it independent of the structure of LMS Moodle.

Because of the distributed nature of the problem, MAS tracks user actions in the environment, and suggest challenges based on each person's background. The implementation of the system is a consequence of this study.

ACKNOWDLEGE

We would like to thank the Support Program for Scientific Initiation (PAIC) of the Foundation for Amparo and Research of the State of Amazonas (FAPEAM), for the financial support provided for the realization of the project in which this paper belongs.

REFERENCES

[1] F. Akim, V. Kapila, and M. Iskander. "Using robotics to promote learning in elementary grades." 119th ASEE Annual Conference and Exposition. 2012.

[2] M. Pedro, R. Elio, F. German, and G. Manuel. “Robotic Educational Tool to engage students on Engineering”. In Frontiers in Education Conference (FIE), 2016 IEEE (pp. 1-4). IEEE.

[3] G. Raffaele, R. Falconi, and C. Melchiorri. "Unibot remote laboratory: A scalable web-based set-up for education and experimental activities in robotics." IFAC Proceedings Volumes 44.1 (2011): 8521-8526.

[4] K. Miroslav, C. Jan, K. Karel, K. Tomás, F. Jan and P. Libor. (2013). “SyRoTek—Distance teaching of mobile robotics”. IEEE Transactions on Education, 56(1), 18-23.

[5] C. Dictino, C. Jesús, L. José Antonio, and D. Sebastián. (2013). “Virtual and remote robotic laboratory using EJS, MATLAB and LabVIEW”. Sensors, 13(2), 2595-2612.

[6] M. Casini, A. Garulli, A. Giannitrapani, and A. Casini. (2014). A remote lab for experiments with a team of mobile robots. Sensors, 14(9), 16486-16507.