A Distributed Extension of the Hybrid PRS System using Video Processing to Command a Robot via Bluetooth

Mauro L. de Freitas†*, Victor E. L. Valenzuela†*, Mikhail Y. R. Gadelha†*, Waldir S. S. Júnior†, Lucas Cordeiro†, Vicente F. Lucena Jr†

† *Federal University of Amazonas*
* *M.Sc. students of Electrical Engineering*
Av. General Rodrigo Octávio Jordão Ramos, 3000, Setor Norte, Coroado I – Manaus, AM, Brasil

Summary
We describe and evaluate an extension of the hybrid architecture called Procedural Reasoning System (PRS) to control the navigation of a mobile robot in a dynamic environment. In contrast to other approaches that perform all route planning in pre-runtime and are sensitive to sudden changes, our hybrid architecture redirects actions based on dynamic changes of the environment. Experimental results show that our approach allows the robot to achieve approximately 95% of its goals.

Key words:
Hybrid System, Bluetooth, LEGO Mindstorms Robotics, Cell phone.

1. Introduction

As the micro-controllers become smaller, faster, cheaper and more reliable, their range of application tends to expand quickly. Nowadays, micro-controllers are used to control a wide range of systems ranging from simple mobile phones to entire product manufacturing. These micro-controllers interact directly with hardware devices in order to control a physical (or dynamic) process. The software that usually runs in the micro-controllers must react to events occurring in the control process and issue control signals in response to these events within a given timing constraints [1]. The software is thus embedded in some larger system and must respond, in real time, to changes in the system's environment.

In this work, we thus develop a (real-time) software that is capable of controlling the navigation of a mobile robot. The software is embedded into the LEGO® MINDSTORMS™ kit, which allows a wide variety of robot models to be built and can easily be programmed via the NXT™ micro-processor. In particular, we propose an extension of the hybrid architecture called *Procedural Reasoning System* (PRS) [2],[3] to allow the use of a secondary processing device with the purpose of increasing the computational power. This thus allows us to integrate more computational resources (such as memory, data and graphical processors, digital cameras and sensors) into the system. As a consequence, we can easily allocate expensive tasks of the system (in terms of computational effort) to the appropriate physical device to efficiently compute the results of our navigation algorithm.

Additionally, we have extended the communication infra-structure of the PRS in order to focus on the communication by messages to suggest objectives and to dynamically control the priority of the system. In this particular case, we assign the highest priority tasks to the reactive system so that we can ignore other actions that might impact the performance of the navigation algorithm and thus contribute to the integrity of the physical robot.

The second contribution is the system itself. More specifically, the extended distributed subsystem that we designed has two novel characteristics. The first one is the use of a low cost platform with hardware and software suitable for the proposed applications. In this case, the platform that we used consists of a phone camera running the Android operating system [4]. The second one is the communication technology that allows mobility and reuse of distributed devices. In this case, we implemented the communication protocol using the LEGO® MINDSTORMS™ kit, in such a way to conform to the necessary standards for data communication between the mobile phone and the robot. This model allows the replacement of the mobile phone by other device if they have the same operating system (Android), a camera and the same technology for wireless communication (Bluetooth) [5], [6].

This work is organized as follows: In Section 2, we discuss briefly the main related work. In Section 3, we describe our extended model with particular emphasis on the characteristics that impact the creation of the proposed system. In Section 4, we explain how the experiments were conducted in order to validate the proposed model. In Section 5, we summarize this work and present the future work.

2. Related Work

Guilherme and Avinash [1] describe three models of autonomous robots navigation. On the map-based

Manuscript received July 5, 2012
Manuscript revised July 20, 2012

navigation, the system requires that the environment representation (i.e., the geometric or topological) must be provided by the user. On the map-building-based navigation, the robot constructs their geometric or topological representation of the environment and uses them for navigation. On the maples navigation, no internal representation of the environment is used for navigation; it is based on the recognition of objects in the environment. The motion decisions are made based on image analysis.

The work proposed by Mahit and Fevzi [7] present a speed control and positioning of an autonomous robot in a variable path. The robot has a camera for image capture (which is attached to the front part of the robot) and a transmitter. The images are sent to a computer along with the wheel angle and robot's speed information. On the computer, the images are processed using fuzzy logic [8] for extraction of path trajectory. Finally, the speed and trajectory information are sent to the robot.

A monitoring system for autonomous robots is described by Ju et al. [9]. There are essentially six devices: two cameras, a transmitter, a phone, a computer and a robot. The objective is to monitor the movements of the robot in a controlled environment. Note that in this proposal, there is no video processing. The images are captured by two cameras (located at the robot and above the environment) and the images are only displayed in the computer.

Robert [10] describes a robot that catches tennis balls. The author used a fixed camera located above the environment so that the robot and the tennis balls are always visible by one computer. The camera acquires the images and transmits them to a computer using a USB port. After that, the system uses the software Robot Vision CAD (RvCAD), which is responsible for image processing and for performing the robot movements. Finally, the commands are transmitted via serial port to the robot. Due to the robot's mechanical structure, which has a ball trap in front of it, the robot only needs to move over the balls to catch them.

3. Proposed System Model

Our proposed system consists of two main blocks. The first one, called *Secondary Processing Device*, is formed by a distributed video acquisition and processing platform. The second block, called *LEGO NXT Robot*, is formed by different components of the LEGO MINDSTORMS® platform (see Figure 1). The communication interface between the blocks is carried out through a Bluetooth transmitter. The system is designed to perform certain tasks that are designed effectively by the LEGO robot (as described in Table 1). A system that is able to efficiently handle these tasks is the so-called procedural reasoning system (PRS), which is of general reasoning and includes the specification of goals with reactive behavior (see [2], [3], and [12]). While traditional systems perform all route

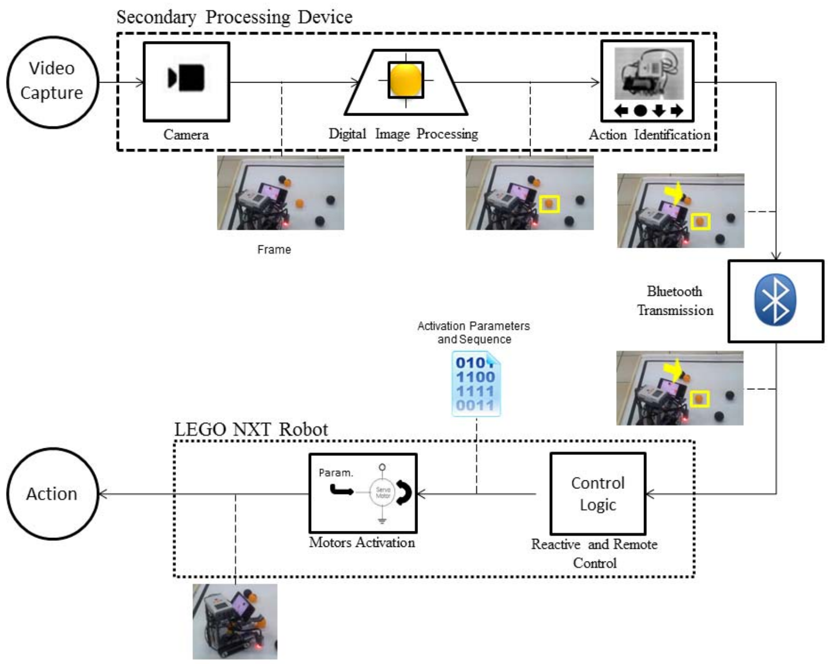


Fig. 1 Proposed System.

planning before running and are sensitive to sudden changes to the environment, the PRS redirects actions based on dynamic changes of the environment. This system is thus based on the principle that the interaction with the environment is always the same, i.e., the system already has all features to achieve in relation to the stated objectives. However, a system that employs the idea behind the PRS assumes that all layers are in a single processing device. In our proposed system, this idea is modified in order to process video frames in a distributed platform. We therefore modified the generation of goals to exploit the parallelism in our hybrid architecture.

Table 1: Effective tasks done by the robot

Task	Layer
Left	Deliberative
Right	Deliberative
Forward	Deliberative
Backward	Deliberative
Stay in environment	Reactive
Harvest	Deliberative
Release	Deliberative

In our proposal, we decided to devise an algorithm for navigation of the robot in the environment movement. This algorithm uses information only from the video frame provided by the acquisition platform. The design of the algorithm takes into account the limitations of the available hardware (e.g., memory and processing speed). Note that there is a compromise between the design of the algorithm, the ability to perform complex tasks of the robot, and the processing of video frames.

To join goal-structures and reaction to external influences, we use hybrid architectures for robots (see [3], [12] and [13]). As an example, the goal of collecting the spheres has a lower priority to the goal of staying within the environment. Thus, the priority of tasks is determined according to its association with the corresponding layer. In this paper, we consider the so-called deliberative and reactive layers, used in the PRS system. According to [2] and [14], the tasks can be implemented with different priorities. In this work, however, tasks associated with reactive layer have higher priority.

As a result, the identification of the movements to be made in the environment is performed by the secondary processing device and sent over the Bluetooth connection to the robot, where these commands are processed and executed if the sensors do not indicate that the robot is leaving the environment. Otherwise, the robot will navigate and ignore all commands until it is within the permitted area. Figure 2 shows the logic used in the block of reactive and remote control. It shows the data-flow that is sent to the mobile phone. This block performs the selection between the execution of received external action and the internal action. Whenever the data are received, boundary sensors are checked. If they are active, it starts a process to return to the allowed region for navigation, otherwise, it performs the received action.

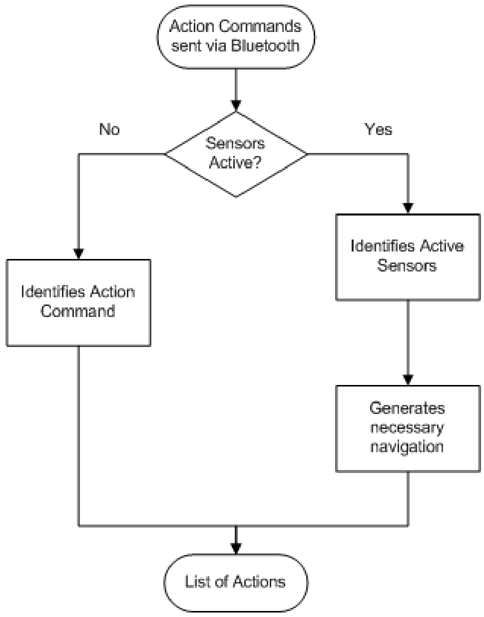


Fig. 2 Reactive and Remote Control Logic.

One of the advantages of the suggested system compared to other related systems (e.g., [7], [11]) is the distributed computing. At one hand, in these related systems, part of the processing is done by a computer which does not move around the robot, thereby, restricting the movement to the length of the communication cables. On the other hand, our approach does not restrict the movement area of the robot and also allows the distribution of the computational effort between the mobile devices. Note that this advantage is important because of hardware limitations of the LEGO platform. Specifically, it does not have accessory for capturing high resolution images (up to 1Mpixel), its processor (ARM7 AT91SAM7S256) has low speed (48 MHz) and memory resources are scarce (64kB RAM). Thus, processing the video frames would not be possible without attaching the mobile phone to the proposed system.

Another advantage of the proposed system is the interface between the blocks; in this case, we use the Bluetooth technology [15], [16] to connect both blocks shown in Figure 1. Note that the availability of a Bluetooth allows devices to exchange data with the robot [2], which thus gives us the possibility to use not only a mobile phone, but also a tablet. These devices have superior processing power in comparison to LEGO's platform. In Table 2, we show the main differences between our system and the one described in [10].

Table 2: Comparative of the proposed system and Robert[10].

	Our Proposed System	Robert [10]
Mobility	The device to make decision is attached to the robot.	The device to make decisions is a PC on a fixed position.
Wireless connections	The communication between the robot and the cell phone uses Bluetooth.	The communication between the robot and the PC uses a wired serial port.
Vision device	The camera is responsible for acquiring images, which is located at the mobile phone (attached to the robot).	The camera is responsible for acquiring images, which is located at the ceiling of the environment.

memory and processing limitations. This algorithm can be represented as the block diagram shown in Fig. 3. First, we have the input that consists of several frames of video, encoded in three channels (RGB) of 8 bits each and resolution of 640x480 pixels, captured by the acquisition distributed platform. There is a transformation of frames into a color space called by YCbCr [17]. In this case, three channels are generated (8 bits each). This change provides robustness to changes in illumination. After that, a reference threshold is assigned to each array that represents the three channels observing the values obtained for the spheres that constitute the objects of interest. Finally, a mask is applied between the matrix resulting in a single binary matrix which shows the orange ball position. Note that all other patterns, i.e., objects that are not of interest (black spheres) are excluded.

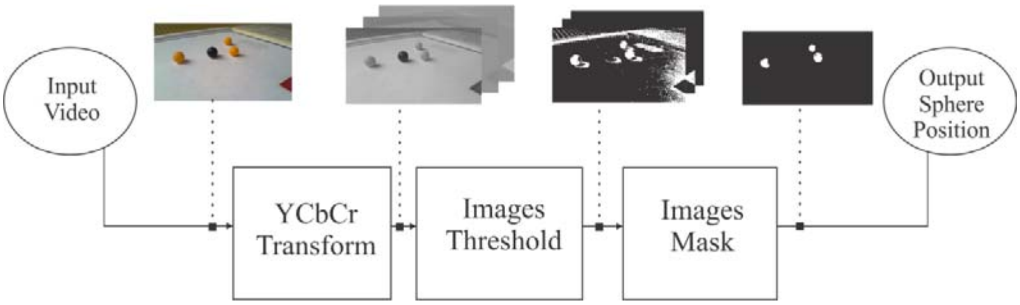


Fig. 3 The proposed system algorithm.

An important task performed by the robot is moving within the limits imposed by the environment. If it comes to move beyond these limits, it would be too complex to replicate the states in the remote device (e.g., computer, mobile phone, or tablet). In our proposed system, however, this was solved using the concept of reaction to external influences [14], i.e., actions activated by the sensors of the robot to preserve the integrity and to remain within the limits imposed.

An important part of the proposed system consists in processing the video frames in order to achieve, for example, recognition of the orange spheres and discard spheres with black color. The design of the algorithm for video processing (performed on a distributed mobile platform) is crucial because the decisions sent to the robot are determined by the output of this algorithm. In this paper, we adapted the algorithm developed by [17], which provides an efficient approach for classification and tracking of objects in video frames for platforms that have

4. Experiment Procedures

4.1 Experiments setup and comments

For this work, the programming language used is called NXC (*Not exactly C*), which was developed specifically for the LEGO Mindstorms platform [18]. Similar to the C language, it allows the creation of jobs, which is necessary for the robot to manage the operation of two layers: *reactive* and *interpretive*. An important aspect of this language lies in the fact that it does not need a custom firmware for execution, which means that it uses the original firmware provided by the manufacturer.

The development environment to create the LEGO ® NXT's program was the Linux Operating System (OS) with the building tool called *Next Byte Codes* (NBC) [19]. This tool allows the compilation of source code and also writes binary files through the USB connection to the NXT device.

The distributed computing platform consists of a Samsung Galaxy SII mobile phone [20] with Android OS v2.3. The development environment was Linux OS with Eclipse Integrated Development Environment (IDE) and the development tools for Android OS (SDK) [4]. This platform (Samsung Galaxy SII) captures video through the camera. The algorithms for processing the video frames are designed using the Open Source Computer Vision Library version 2.3 (OpenCV 2.3) [21] in C++ programming language. The integration of Android (Java) to OpenCV (C++) is done through the Java Native Interface (JNI) using the Android Native Development Kit (NDK).

As mentioned in Section 3, the proposed system is designed to effectively execute various tasks by having the robot as the main tool. Action commands are sent from the mobile phone to the NXT device through the Bluetooth communication protocol [15] and there are no return messages from the robot. These commands thus allow the tasks to be executed.

4.2 Simulations Experimental Procedures

The test environment was a white colored arena with $2m^2$, orange and black colored spheres, that were randomly arranged. We also marked a starting position, which is painted in red as shown in Figure 3. The side stripes, which indicates the permitted limit zone for navigation, were made of 2cm thick black tape.

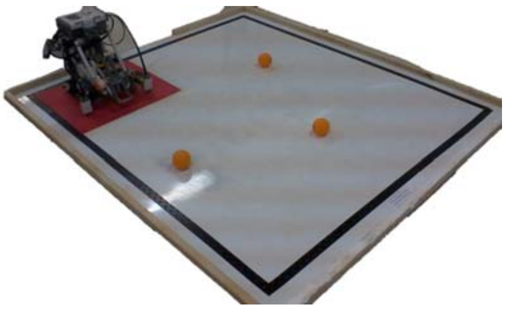


Fig. 4 Test Arena

The system performance was evaluated by observing the results of 40 simulations carried out in the test arena. On each simulation, the robot was positioned on the red marker and the spheres were arranged randomly. The objectives of the robot were: capturing three orange spheres, return to the red zone and depositing there all spheres. The robot was not allowed to capture the black spheres, i.e., it should ignore them. Also it was allowed to push the black spheres while moving in the arena. The results of each simulation were evaluated in a binary manner: *success* or *failure*. A result was successfully categorized if the robot fully executed all of its goals, otherwise the result was categorized as failure. In Table 3, we can see the results that we obtained. Note that the performance of the proposed system was satisfactory. In 95% of the cases the robot performed all of its objectives.

Table 3: Experimental results used to evaluate the performance of the proposed system.

Status	Number of experiments	Number of experiments (%)
Success	38	95%
Non-Success	2	5%

4.3 Results Summary

This paper proposes a modification of the PRS hybrid architecture. The purpose is to consider an extension that provides the execution of two separate processing centers. The extension was implemented using a mobile phone to capture and process the video, thus communicating through an exchange of messages by suggesting new goals to the NXT robot system. During the experiments we observed that:

1) The movement sequences were executed correctly in different situations in which the robot tried to move out of the region allowed. Thus, we guarantee that the robot maintains its integrity by not leaving its area demarcated by the black stripes on the edge of the environment.
2) Regarding the identification of the spheres contained in the frames captured by the camera phone from the Samsung Galaxy SII, the image processing algorithm carried out correctly the segmentation and identification of the spheres. In particular, we cite the example in which two spheres are very close to each other. In this case, the algorithm behaves as expected, i.e., it identifies the two spheres with success. Another situation is that the segmented spheres images were obtained with shadows. In this case, the shadows did not change the performance of the algorithm. In tests with balls of different colors of the specified tests and smaller physical sizes, they were not captured, as expected.
3) After the continuous processing of video frames captured by the mobile phone, instructions for collecting the spheres were sent to the LEGO NXT Robot until the collection itself was performed by the robot. This procedure was performed in different ways (a total of 40 times)

until the robot was able to collect three spheres in a row. In all cases, the collection was successfully performed, i.e., no mechanical error influenced in the accuracy of the collection and, consequently, an error count of stored spheres.

4) In respect to the return procedure of the robot to the starting position after the collection if three spheres, we observe that in all tests the robot behaved in the manner which we expected, i.e., it returned to the starting position specified by the environment and released the collected spheres. The end of its path, although the sphere arrangement was random, in all cases was the same.
5) The robot just captures one kind of spheres and ignores others. This feature made it possible that spheres were displaced by the robot during its trajectory. This situation was considered acceptable in this project.
6) The experiments resulted in a rate of 95% success and 5% non-success. The last was related to all cases in which the robot did not realize the goals in its entirety for any reasons, e.g., the spheres are together and the image processing algorithm did not recognize them.

With these results at hand, we have strong evidence that the modified hybrid architecture achieved a satisfactory performance because during the experiments the robot did not exceed the defined areas. The segmentation and identification of the spheres in all video frames captured by Samsung Galaxy SII occurred satisfactorily, even though there was proximity between the spheres. Most importantly, the return procedure of the robot with collected spheres was successful. Another important point lies in the fact that there was no collection of spheres of smaller physical sizes and colors not specified in the experiments.

5. Conclusion

In this work, we have proposed an extension of the hybrid architecture called *Procedural Reasoning System* (PRS) to allow the use of a secondary processing device with the purpose of increasing the computational power, which thus allows us to integrate more computational resources (such as memory, data and graphical processors, digital cameras and sensors). Two major contributions were described. First, we have extended the communication infra-structure of the PRS in order to focus on the communication by messages to suggest objectives and to dynamically control the priority of the system. The second contribution is the system itself. In particular, the extended distributed subsystem that we designed has two novel characteristics. The first one is the use of a low cost platform with hardware and software suitable for the proposed applications. The second one is the communication technology that allows mobility and reuse of distributed devices. We developed the proposed system and the results were promising. In 95% of the cases, the objectives were fully achieved.

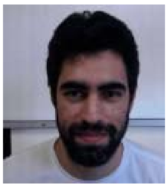
Acknowledgments

To National Council for Scientific and Technological Development (CNPq - Brazil), for providing funding to conduct this research.

References

[1] Hermann Kopetz. "Real-Time Systems: Design Principles for Distributed Embedded Applications". Kluwer Academic Publishers, Second ed., 2002.
[2] F. Ingrand, R. Chatila, R. ABami, F. Robert, “PRS: A High Level Supervision and1 Control Language for Autonomous Mobile Robots”, Proceedings of the IEEE International Conference on Robotics and Automation Minneapolis, Minnesota, pp.44-49, April, 1996.
[3] F. Ingrand, O. Despouys, “Extending Procedural Reasoning to ward Robot Actions Planning”, Proceedings of the 2001 IEEE International Conference on Robotics & Automation, Seoul, Korea, May, pp. 21-26, 2001.
[4] Android Open Source Project, [Online]. Available: http://source.android.com/source/using-eclipse.html. [Access in May 15th, 2012].
[5] J. Lee, Y. Su, C. Shen, “A Comparative Study of Wireless Protocols Bluetooth, UWB, ZigBee, and Wi-Fi”, The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)pp.5-8, Taipei, Taiwan, Nov, 2007.
[6] E. Ferro, F. Potortì, “Bluetooth and Wi-Fi Wireless Protocols: A survey and a Comparison”, IEEE Wireless Communications, pp 12-26, February 2005.
[7] Mahit G., AFevzi B. “Speed and Position Control of Autonomous Mobile Robot on Variable Trajectory depending on its Curvature”, Journal of Scientific & Industrial Research Vol. 68, pp. 513-521, 2009.
[8] K. R. S. Kodagoda, W. S. Wijesoma, E. K. Teoh, “Fuzzy speed and Steering Control of na AGV”, IEEE Transaction on Control Systems Technology, vol.10.
[9] D. Ju, R. Zhong, M. Takahashi, “Development of Remote Control and Monitor System for Autonomous Mobile Robot Based on Virtual Cell Phone”, Second International Conference on Innovative Computing, Information and Control, pp. 291, 2007.
[10] J. W. Robert, “Object Capture with a Camera-Mobile Robot System”, Robotic and Automation Magazine, vol.13, pp. 85-88, 2006.
[11] Guilherme N. S., Avinash C. K., “Vision for Mobile Robot Navigation: A Survey”, Pattern Analysis and Machine Intelligence, vol.24, pp.237-257, 2002.
[12] T. Bräunl, Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems, Germany: Springer-Verlag Berlin Heidelberg, 2006.
[13] R. Granot, *Introduction to Robotics – Deliberative and Hybrid Control,* Israel: Haifa University, 2008.

[14] A. Luca, L. Ferrajoli, “Exploiting Robot Redundancy in Collision Detection and Reaction”, IEEE/RSJ International Conference on Intelligent Robots and Systems Acropolis Convention Center Nice, France, Sept, 22-26, pp. 3299-3305, 2008.
[15] LEGO The Group, [Online]. Available: http://mindstorms.lego.com/en-us/support/files/default.aspx. [Access in May 15th 2012].
[16] J. Lee, M. J. Huber, E. H. Durfee e P. G. Kenny, “UM-PRS: An Implementation of the Procedural,” *AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service, and Space*, pp. 1-8, 1994.
[17] Fredrik Kristensen, Peter Nilsson, and Viktor Owall. Background segmentation beyond RGB. *Lecture Notes in Computer Science*, 3852:602–612, 2006.
[18] J. C. Hansen, “LEGO Mindstorms NXT Power Programming: Robotics in C”, Second ed., Variant Press, 2009.
[19] J. Hansen, “Next Byte Codes Compiler,” 2011. [Online]. Available: http://bricxcc.sourceforge.net/nbc/. [Access in July 23rd, 2011].
[20] Samsung Galaxy S II, [Online]. Available: www.samsung.com/global/microsite/galaxys2/. [Access in May 16th, 2012].
[21] Willow Garage, [Online]. Available: http://opencv.willowgarage.com/wiki/Welcome. [Access in May 13th, 2012].



Mauro L. de Freitas was born in Manaus, AM, Brazil, in 1987 received the B.S degree in Computer Engineering, postgraduate at Video Systems and started the M.E degrees at the Federal University of Amazonas (UFAM) in 2009, 2010 and 2011 respectively. During 2009-2011 studied and worked with IC design and digital signal processing. His research interests are digital communication, signal processing and embedded systems.

Victor E. L. Valenzuela received B.E. degree from Amazon State Univ. in 2009 and started M.E. degree program at the Federal University of Amazonas (UFAM) in 2011. Worked developing Industrial Automation Systems Solutions in 2010. His research interests are Robotics, Industrial Automation Systems and Embedded Systems Design.

Mikhail Y. Ramalho Gadelha was born in Manaus, AM, Brazil, in 1988, received the B.E. degree in Computer Engineering in 2010 and started the M.E. degree at the Federal University of Amazonas (UFAM) in 2011. Worked developing mobile applications for Maemo and Meego in 2010 and developing SBTVD (Brazilian Digital TV standart) in 2011. His work focuses on software verification, bounded (and unbounded) model checking, mobile applications development and embedded systems.

Waldir Sabino da Silva Júnior received the B.S. degree in electrical engineering at the Federal University of Amazonas (UFAM), Manaus, AM, Brazil, in 2000, and the M.S. degree in electrical engineering at the Federal University of Rio de Janeiro (COPPE/UFRJ), Rio de Janeiro, RJ, Brazil, in 2004 and Ph.D. degree in electrical engineering at the Federal University of Rio de Janeiro (COPPE/UFRJ), Rio de Janeiro, RJ, Brazil, in 2010. Since 2006, he has been with the Federal University of Amazonas, as Full Professor. His research interests are in the fields of data compression, as well as in mathematical morphology, pattern recognition and digital signal processing in general.

Lucas Cordeiro received the B.Sc. degree in electrical engineering and the M.Sc. degree in informatics from the Federal University of Amazonas (UFAM), in 2005 and 2007, respectively. He received the Ph.D. degree in computer science from University of Southampton in 2011. Currently, he is an adjunct professor in the Electronic and Information Research Center at UFAM. His work focuses on software verification, bounded (and unbounded) model checking, satisfiability modulo theories, and embedded systems.

Vicente Ferreira de Lucena Jr. received his Ph.D. degree (Dr.-Ing) in 2002 from the University of Stuttgart – Germany. Since 1990, he has been a Faculty Member with the Electrical Engineering Department at the Federal University of Amazonas. Since 2003 he is also with the CETELI a special interest group that works with consumer electronics. His research interests include the development of automation systems, new software engineering approaches and the development of embedded systems, electrical drives and system identification.