

4-D MODELING APPLIED TO INDUSTRIAL AUTOMATION

Ivando S. Diniz, *ivando@sorocaba.unesp.br*

Diego Colón, *dcolon@sorocaba.unesp.br*

Fernando P. Marafão, *fmarafao@sorocaba.unesp.br*

Frederico J. Moron, *fjmoron@gmail.com*

UNESP – Univ Estadual Paulista

Group of Automation and Integrated Systems (GASI)

Av. Três de Março, 511, Sorocaba, SP – Brazil – 18087-180.

Abstract. *This paper presents a 4D modeling, monitoring and control system for an industrial plant. In the competitive markets, industries are required to improve significantly their productivity, reduce the waste of raw materials, stock's size, repairing time for (faulty) equipments and everything else that reduces the efficiency. Automatic control systems, monitoring systems and process' optimization have been increasingly used in the last years in order to accomplish these objectives. Today's industrial monitoring systems technology is based in 2-D graphic representation of the several plant components (tanks, pumps, pipes, and robots, to name a few) and the events related to them (for example, start / stop, warnings and alarms). As the events occurs in real time (and are registered in data bases) the name 3D supervisory system was adopted. The today's tendency, however, is to represent the several plant's and control subsystems (as well as the corresponding events) in three-dimensional figures, what coined the name "4D modeling, monitoring and control system". The main idea is to develop (in a computer) a virtual reality plant (representing the real one) in an environment very much like the "first person" computer games (like "Counter Strike" and "Doom 3") in order to provide to the system's operators some desirable characteristics, like: exact subsystems locations (for the easiness of access) and more intuitive navigation, by avoiding a great number of screens and windows. In this paper, a simplified 4D monitoring and control system developed at the University Estadual Paulista (UNESP) is presented. The plant is a fan (driven by a DC motor) mounted in a table. The motor controller is implemented by means of a microcontroller (MICROSHIP PIC18F87J60) and the supervisory system is a home PC. The communication between them is performed via an Ethernet network and TCP/IP protocol (with CRC error correction). Besides the fact that it is a nondeterministic network (no packet delivery time guaranteed, which is necessary for real time applications), for simple and noncritical applications, it could be used. In more critical applications, the protocol could be easily modified. The 4D supervisory software and the program were developed in C++ language, using standard libraries such as OpenGL (the library used for game developers) and Winsock (for network communications). In order to maintain the graphical performance independent of the network traffic, several threads (parallel execution units) were implemented. The microcontroller software was programmed in the interface MPLAB IDE 8.02 and all the plant's starting procedures are controlled by it.*

Keywords: *Virtual Reality, 4D modeling, Control systems, Automation, OpenGL.*

1. INTRODUCTION

The technological evolution has contributed to the improvement of industrial process control, and the advent of high-speed computers and networks are the key factors for this improvement. The today's monitoring and controlling systems, which are called 3-D, have greatly benefited from the evolution of digital electronics, allowing a great reduction (in size) of the supervisory equipment (from the huge analog panels to a reduced set of industrial computers with flat monitors), as well as the control equipment (from the analog/pneumatic to digital computer as well). Much improvement, however, are expected in the near future, ranging from further miniaturization and performance enhancement till more user friendly human-machine interfaces. The 4-D modeling, supervisory and control systems come as a promise to partly fulfill these improvements, as the three-dimensional representation of the plant's components allows: 1) easiness of location of faulty equipments, and 2) simultaneous presentation (to the system's operators) of the global plant, to name a few.

The main idea of a 4-D supervisory system is to represent graphically (in computer monitors), and in three dimensions, the whole industrial process, which includes: 1) its machines (and geographical position), and 2) the events occurring along the operation phase (like, for example, connection/disconnection of a machine, temperature variations, high level alarms in a tank). Some research activities have also started in this area, as for example the virtual laboratory for non-motorized travel (developed in the UNC Highway Safety Research Center) and the training and monitoring software, under development by São Paulo University, for Vale do Rio Doce Company . In the area of 4-D modeling, some unexpected endeavors, such as the "Walt Disney Concert Project Hall", have allowed that, by means of a 4-D model of a big playhouse, many issues concerning accessibility, security and emergency health assistance were addressed (by simulation of crowd and vehicle dynamic motions in critical situations in buildings). Thus, this paper is divided as follows: in section two, an historical perspective of the graphical representation of supervisory and control systems are presented. In section three, the technology used in 4-D supervisory systems are described. In section four,

the system developed at UNESP Sorocaba is presented, in section five, some experimental results are shown and in section six, conclusions are drawn.

2. A HISTORICAL PERSPECTIVE

The original supervisory systems were based on huge analog panels (with lamps, gauges, switches and buzzes) indicating some restricted set of variable and alarms, hardly indicating the position of the faulty component. As the computers evolved, 3-D supervisory software started to be developed, as the one presented in figure 1 (in fact, it is just a screen among many others). The subsystems were represented in mnemonic ways, and several graphical resources (such color changes, alarm windows and switches) could be associated to these objects. Also, a little geographical information could also be represented.

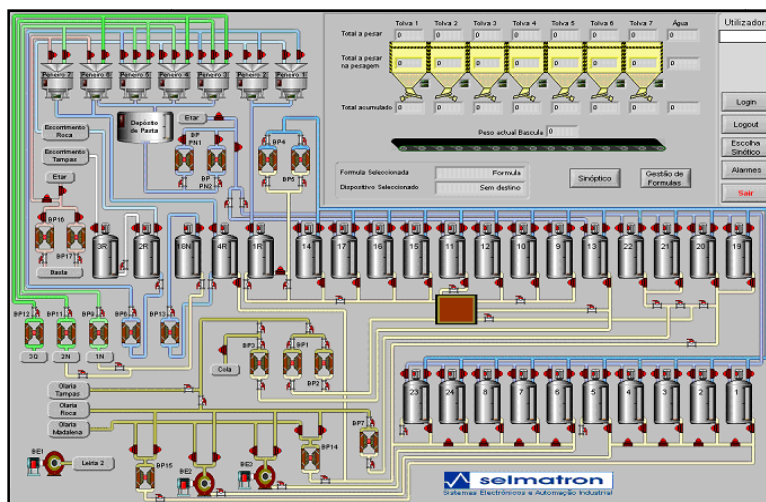


Figure 1 – Industrial process represented in 2D images. (Selmatron courtesy).

At the same time, some 3D modeling (three space dimension, without time) started to be used, as the one presented in figure 2. In this figure a complete industrial plant is represented in isometric perspective. Some geographical and supervisory information is in fact presented in this representation. However, many equipment characteristics, as their relative sizes and exact positions were absent. In order to see equipment details, other windows should be opened, which could overload the image and difficulty the reading. Besides that, the concept of geographical navigation is completely absent in this representation.

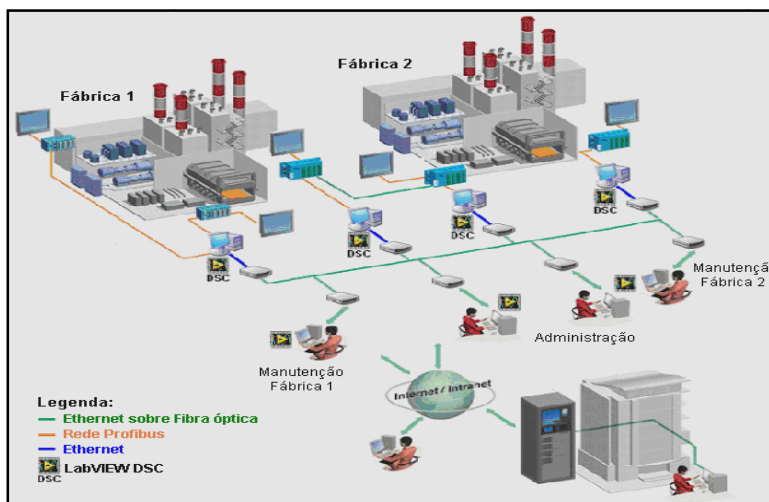


Figure 2 – Industrial process represented in isometric perspective. (NI courtesy).

3. THE TECHNOLOGY OF 4-D REPRESENTATION

4-D modeling and supervision consists of graphical (three-dimensional) representation of the process to be controlled, as well as the real-time supervision (which accounts for the additional dimension in the name) in much the same way as in virtual reality (Siscoutto, 2008). This means that the process' operator (that is, the system's user) could navigate in the "virtual plant", as he/she would do in a 3-D videogame (Schuytema, 2008). The equipments, as well as their keys, lamps and rotating parts, must be represented with real dimensions, and their sizes must proportionally change, as the operator gets closer or farther. Also, the interaction between the operator and the virtual equipment should cause the effect, in the real plant, that is represented in the screen. The interaction with the physical plant, as would be expected, should cause the same effect in the virtual platform.

In order to provide these capabilities to the supervisory and control system, many software tools have to be used, as for example C4D, Maya, Game Studio, 3D Max (Aguiar and da Silva, 2007), and Blitz3D, Panda3D, DirectX and OpenGL as cited in (Shreiner *et al*, 2007).

3.1 Virtual reality

Applications of virtual reality started to be used in the 50's, with the development of flight simulators for the United States Air Force. A major advance occurred with the development of the stereo vision helmet by Philco in 1958. Soon after, the Sensorama appeared for the first time, which is a combination of 3-D movies with stereo sound, mechanical vibrations and even aromas and induced wind. It was patented by Morton Heilig. In 1965, Ivan Sutherland presented for the scientific community the idea of using computers to create drawings in the screen, which is the beginning of the computer-aided design (or CAD) era, as well as graphical computation. In the end of the 70's and beginning of the 80's, the first digital gloves appeared and in 1989, the Autodesk Company presented the first system of virtual reality based in a personal computer. In Brazil, the Laboratory of Integrated Systems (LSI) in the Polytechnical School (at USP) started the research and development of the Digital Cave. In (Siscoutto and Costa, 2008) presented techniques of increased virtual reality in the X Virtual Symposium on and Augmented Reality, in João Pessoa.

3.2 OpenGL

OpenGL is a software routine library (API) for free use in computer graphics. It is intended for developing applications charts, 3-D environments, videogames, among others (Shreiner *et al*, 2007). This interface consists of about 150 distinct low-level commands/routines used to specify the objects, as well as their movements and interactions in three-dimensional applications. OpenGL is designed as a streamlined, hardware independent interface to be implemented on many different hardware platforms (Shreiner *et al*, 2007) and (Schildt, 1997). Those commands allow one to construct complicated three-dimensional objects (like automobiles, parts of the body, airplanes, or even molecules) based in a small set of geometric primitives, such as points, lines and polygons. Sophisticated library functions, for constructing those complicated objects, should be built using those primitive commands of OpenGL (Schuytema, 2008).

3.3 Ethernet

Ethernet was developed in Xerox PARC by Robert Metcalfe, who, years later, left the company to promote the use of computer networks and create 3Com (former U.S. Robotics). Metcalfe, in 1980, managed to convince great companies as Intel and Xerox to cooperate and promote the Ethernet, which in September, 30 was transformed in a international standard. Then, it started to compete with proprietary systems, as token-Ring and ARCNET. Today, Ethernet is the most popular network standard in the world (Torres, 2001). In an Ethernet local network, the computers broadcasts their messages (packages) to the target computer, and collisions are detected by measuring voltage changes in the cables. If it occurs, the message is retransmitted some seconds afterwards. No time of deliverance is guaranteed, *a priori*. Other network standards, for industrial applications, were developed in order to solve this problem, but in some noncritical applications, Ethernet could be used.

3.4 TCP/IP

The TCP/IP protocol is currently the most popular in applications (mainly in the Internet) and is frequently used along with Ethernet for supplying high speed connection (up to 1Gbps in the Giga bit standard). TCP/IP is a trustworthy connection, which means that it guarantees the package deliverance (retransmitting in case of errors by means of confirm deliverance and collision detection system). In this work, the connection between the real and the virtual plants are made by a net Ethernet 10BaseT, with TCP/IP network protocol, by using the Winsock library (for C programming).

4. THE 4-D SUPERVISORY SYSTEM DEVELOPED AT UNESP SOROCABA

This work presents a 4-D supervisory and control system developed by the Group of Automation and Integrated

Systems (GASI) at UNESP Sorocaba. Firstly, it has been developed a simplified physical plant to be controlled and monitored (a DC motor with manual control), followed by the development of a PC based supervisory and control system, representing the plant, the physical controller (microprocessor based) and the environment where everything occurs. The communication between this controller and the supervisor is based on an Ethernet network. The manual control is used to increase or to diminish the motor speed, and to change the direction of rotation. The complete system is represented in figure 3.

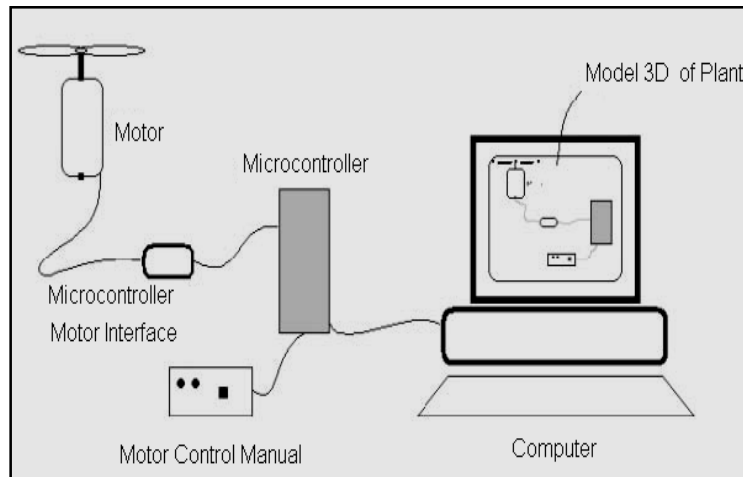


Figure 3 – Schematic representation of the physical system.

The computer must manage the communication between operator and plant, and it must be done as fast as possible in order to be imperceptible for humans. In order to achieve this, the network routine and the graphical routine were implemented in different threads (set of commands to be executed in parallel). The motor controller is implemented by means of a PIC microcontroller, which is also responsible to the communication with the PC (using a buffer of 8 Kbytes of memory).

The speed of the Ethernet adapter is 10Mbps, which is fundamental in order to avoid delays. The virtual environment is composed by a chair, a table, a DC motor (with a fan attached to it), the start/stop panel and an engine room (where everything is located), as shown in figure 4. The viewpoint can be changed by the user by using the mouse (or other especial key).

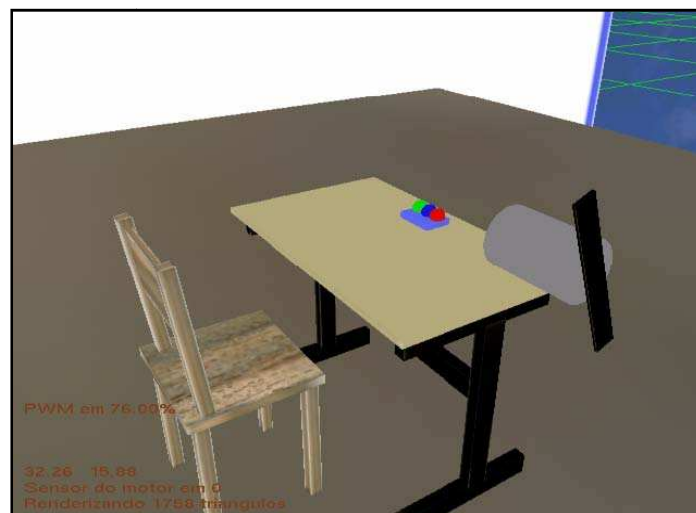


Figure 4 – Virtual environment composed by manual control, table, chair and dc motor

4.1 Development of virtual environment - PC

The virtual environment has been developed using the OpenGL and Winsock libraries, as well as standard C and C++ language libraries. The main program, executed in the PC assumes that the PC is the server in the TCP/IP protocol

and, as it was said, a specific thread, for network communication, was written so that the graphical program does not suffer from delays caused by traffic jams. In figure 5, it is shown the main program's flowchart. When executing the program, the computer loads the 3D archives for the memory and wait until the connection with the microcontroller is completed. As soon as the load of the 3D archives is completed and the connection established, the operator starts to operate as if he/she were in a computer game (see figures 10 to 13) and the virtual controls (that start and stop the motor) can be accessed with the mouse.

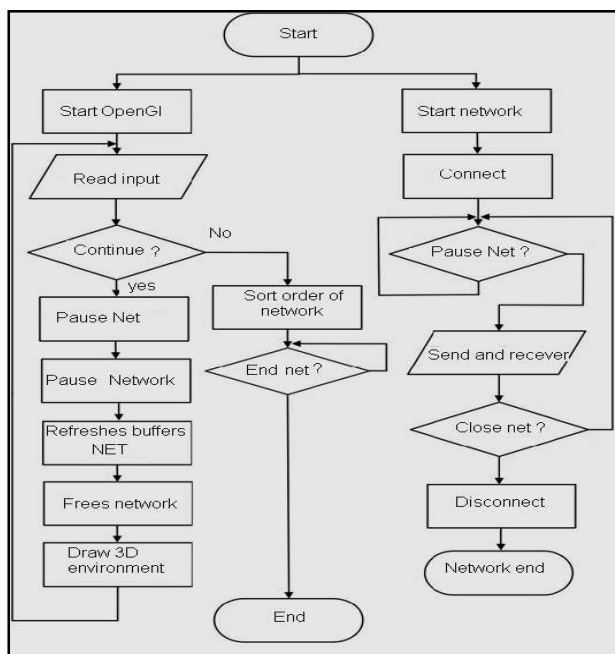


Figure 5 - Simplified flowchart of the program in the PC.

4.2 Physical controller design: hardware and software

The microcontroller PIC18F87J60, shown in figure 6, contains: 1) an internal Ethernet adapter for interface with the PC, 2) a PWM drive for the DC motor, and 3) a timer for digital input and output control (for the indication LEDs).

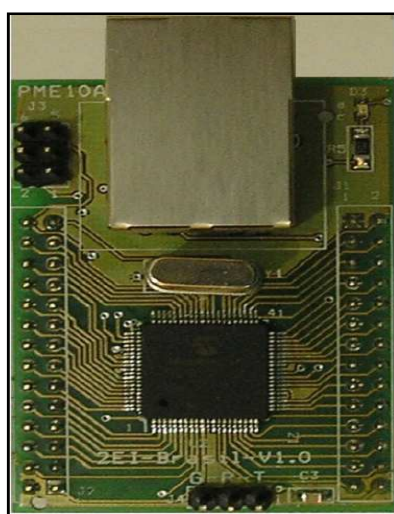


Figure 6 – Microcontroller PIC18F87J60 with Ethernet. (2EI courtesy).

The main program in the PIC microcontroller, with flowchart presented in figure 7, initializes the plant and the PC communication (Pereira, 2008). The program is an infinite loop which performs the following operations: network

connection control, plant information collection, communication with the supervisory plant actuation. Even if there was a TCP/IP connection fail, the PIC would reestablish it. Even during a network failure, plant’s manual control continues to work. There is an angular velocity sensor continuously read by the PIC’s program, which is presented in the virtual environment (see figures 10 to 13). The PME10A (PIC18F1896) card, with couplers and H bridges are used as interface between the motor and the microcontroller.

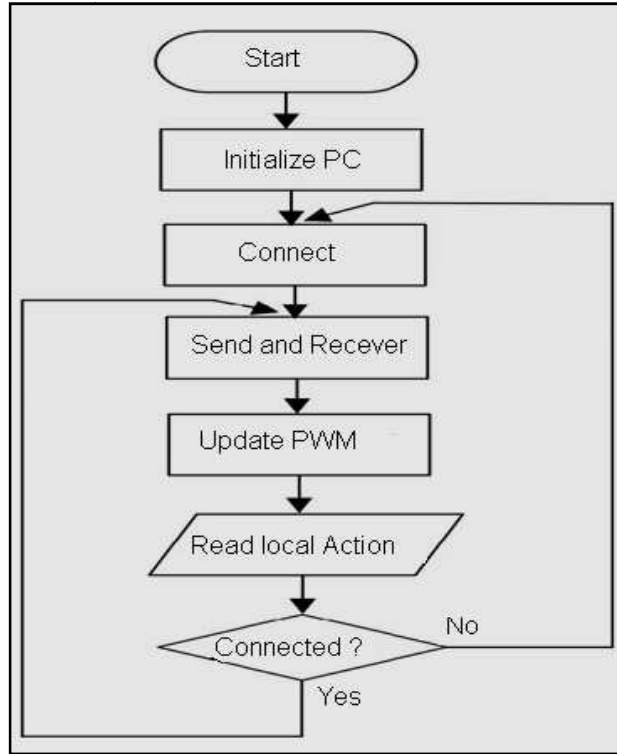


Figure 7 - Simplified flowchart of the PIC’s program.

4.3 Manual control: interface between the real and virtual world

Manual control panel consists of four pulse buttons (i.e., push buttons without lock) to turn the engine on or off and another the rotation speed (by steps). By a characteristic of the H bridges, the motor speed is limited between 18% and 90% of the maximum nominal speed. Figure 8 presents the electric scheme of the manual panel’s.

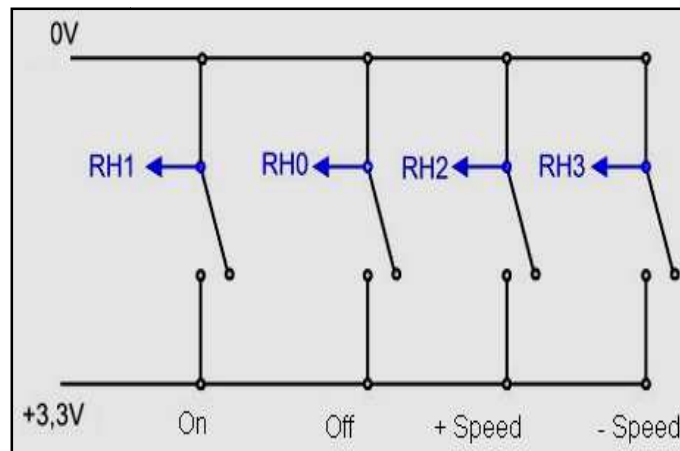


Figure 8 - Manual Control: Interface Between the Real and Virtual.

4.4 Graphical processing: interface between the real and virtual event

One of the major requirements of this project is the computer's ability to render 3D graphics, and this ability is closely linked to the video card used in the PC. It is difficult to know *a priori* what will be the required processing capacity. However, surveys of commercial card's capabilities can be easily found in the specialized literature, which should be very useful in the initial design phase. Some video cards are able to handle 40 million triangles in 1 second and in order to have a good 3D visual effect, the applications must have a refresh rate of 24 or more frames per second. This rate, however, is not stable at run time, so it is normal to consider a refresh rate above 30 frames per second. Typically, pictures with 630 thousand triangles results in good appearance. In the case of this work, the total number of triangles in all 3D models would not pass 50,000, which means that virtually any computer found nowadays is able to run this application with enough speed and good visualization. It is also possible to introduce the ability to process light, texture and effects in order to perform better images. Besides the graphical capabilities (that is, exhibit images in the monitor), there are other tasks that must be performed in order to implement a virtual ambient, as the location of the user in the room (in order to draw the correct size of the objects) and which is the mouse press state.

4.5 Communication between real and virtual world

The communication between microcontroller and PC uses TCP/IP protocol, with the computer as server and the microcontroller as client. A physical/link level Ethernet network was used and programmed with the Winsock standard library. It was a design objective to achieve a communication performance as close of the real time as possible, in order that events in the physical and virtual world occur synchronously. TCP/IP protocol also has the ability to connect more than 60 thousand equipment only with the use of HUBs or SWITCHs. The TCP/IP protocol has the capability of delivery packages with 100% of certainty by implementing a mechanism of receive confirmation and error correction (CRC). However, there is no time of deliverance guarantee, which means that the time to a package to reach its target is random and depends on conditions like the network traffic and electric interference and noise in the place where the system is. That is, the network is not *deterministic*, but practical results have shown that good results are obtained if the bandwidth occupation does not exceed 25% of the maximal capacity.

On the other hand, for more critical applications, there are industrial networks and standards, which effectively guarantee real time behavior. Some examples are: 1.2 HART protocol with Device Net in 125 Kbps, 250 Kbps or 500Kbps velocity and Profibus DP with velocities up to 12 Mbps (or Profibus PA with 31.25 Kbps).

Figure 9 shows a link diagram between the supervisory PC developed in this work and the network. Instead of a microcontroller, a PLC (programmable logical controller) could also be used as the control device, or even many different types of controller (as long as they provide TCP/IP communication capability).

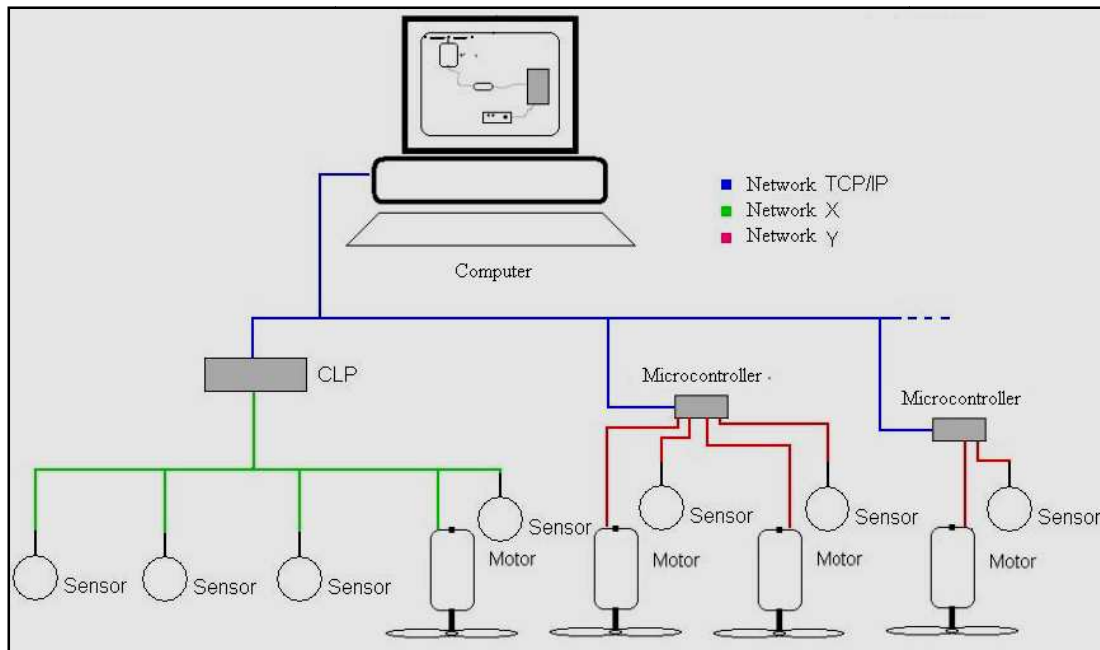


Figure 9- Network with extended capacity.

5. EXPERIMENTAL RESULTS

Tests have been carried out by navigating in the virtual environment, turning on and off the motor and changing its velocity. Figures 10 and 11 shows the user (as in a first person game) entering the virtual room. The “simplified plant” is a table and the motor is over it, as well as the switch panel. Some data as also showed in the left bottom of the screen. As the user “moves” forward, all the objects have their sizes greater than before.

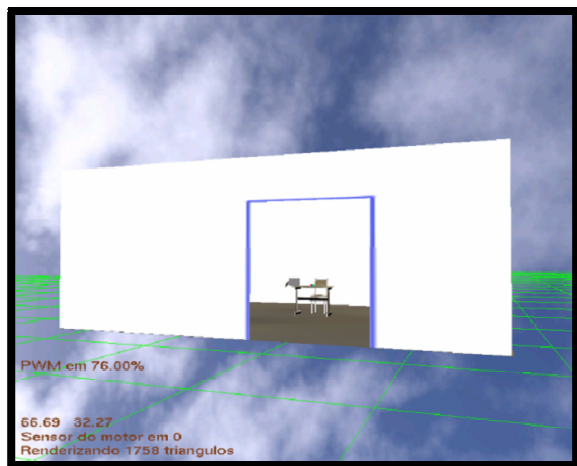


Figure 10. Operator is outside of the control room.

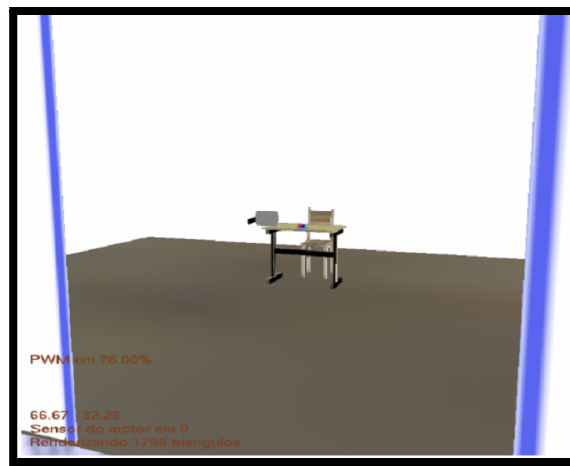


Figure 11. Virtual operator next the door.

In figure 12, the operator is next the table and the DC motor and the keys are visible. By turning on the motor (pressing the green virtual key) the real (and virtual) motor start to running. By pressing the red one, the motor stops and by pressing the blue one, the velocity changes. The same effect occurs when the physical keys are pressed, simulating in the virtual ambient all the real system’s functionalities. In figure 13, it is shown that the operator can circulate around the table. The real velocity is measured by a sensor connected to the microcontroller.

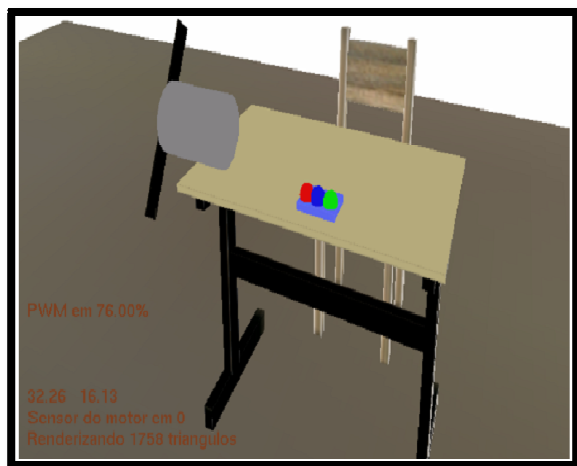


Figure 12. Virtual operator next the table.

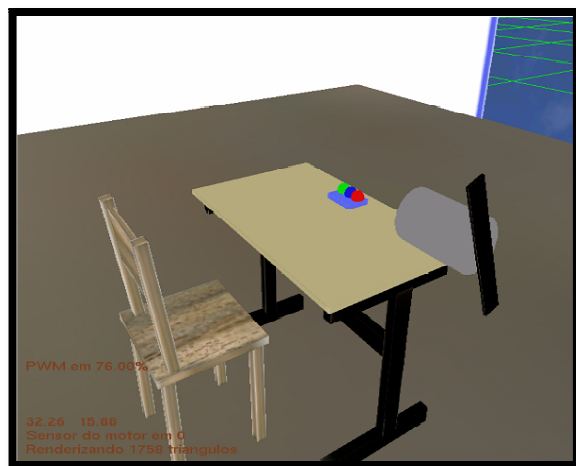


Figure 13. Virtual operator next the DC motor.

6. CONCLUSION

This project consisted in developing a 4-D supervisory and control system based in a home PC. A microcontroller acts directly in the plant (a DC motor) and communicates with a PC by means of an Ethernet network with TCP/IP protocol. The PC program was developed in C/C++ language, and the visual part (panel) has been implemented using the standard OpenGL library. Despite of the fact that it is a nondeterministic network, several tests were carried out, resulting in good performance in this simple case. For more critical application, the supervisory software can be easily modified in order to communicate with industrial network standards (deterministic networks). The graphical performance has shown to be very sensitive to traffic in the network, which has been improved when the communication routine has been executed in an independent thread. The physical world events were very well reproduced in the virtual environment, showing that the communications were sufficiently fast for not causing delays.

The 4-D modeling, supervisory and control systems have shown to be viable to be used in simple industrial plants, and the implementation is simple, if one uses the appropriate software tools. All the components used in the design were low cost and easily found. The developer's programming skills, on the other hand, had to be high level, as the programming were not trivial. As a next step in this research line, this supervisory system will be extended to be applied in a laboratory at UNESP Sorocaba, in order to implement a virtual laboratory in which the experiments can be done remotely (for example, using the Internet).

7. REFERENCES

- Aguiar, F. C., Silva, J. C., 2007, "3-Ds Max 9: Prático e Ilustrado", Ed. Érica, São Paulo.
- Pereira, F., 2008. "Microcontroladores PIC: Programação em C". 7 ed, Ed. Érica, São Paulo.
- Schildt, H., 1997. "C Completo e Total. Tradução de Roberto Carlos Mayer", Pearson Makron Books, São Paulo.
- Schuytema, P., 2008. "Design de Games: Uma Abordagem Prática". Tradução de Cláudia Mello Belhassof. C engage learning, São Paulo.
- Shreiner, D., Woo, M., Neider, J., Davis, T., 2007. "OpenGL Programming Guide. The official guide to learning OpenGL", Version 2.1., Addison Wesley, Boston.
- Siscoutto, R., Costa, R., 2008, "Realidade Virtual e Aumentada", X Symposium on Virtual and Augmented Reality, Ed. SBC, João Pessoa.
- Torres, G.I., 2001. "Redes de computadores Curso completo". Axcel Books, Bonsucesso, Rio de Janeiro.