

# VRML and Simulink Interface for the Development of 3-D Simulator for Mobile Robots

Daniel Domínguez

**Abstract**—The GUI for the development of real time software for the management of mobile robots is perhaps one of the most important aspects to consider for programming the interface for a virtual simulator. There are a lot of ways to develop real time software simulators for mobile robots, but also there are a lot of excellent applications that can be use for programming these simulators. This paper shows how with Matlab that is one of these applications, the cinematic, dynamic and geometric models of a mobile robot were programmed and then visualized with a real time operations on a 3-D virtual simulator. For this application were used the Matlab's toolbox of Simulink for programming the mathematical model of the mobile robot and then the VRML toolbox for visualized it. The mobile robot used for develop this simulator was the Khepera II of the K-Team Corp.

**Keywords**—Interface, Khepera II Robot, Simulink, Virtual Simulator, VRML.

## I. INTRODUCTION

THE importance of visualize the behavior of a robot on a virtual reality environment is one application that allows us not only to study its performance, but also let us to interact on a real way with its perception with the real environment.

This application can be develop with different tools, which can vary from programming code, to just animate the behavior of the robot with some 3D software.

The application of this paper is based on the real behavior of the Khepera II Robot of the K-Team Corp., where the entire cinematic, dynamic and geometric model of the robot was programmed using Matlab code and Simulink blocs, for then create a virtual model of them and interact on a virtual ambient developed on VRML. With this work, it is plan to show that Matlab's toolboxes of Simulink and VRML are a powerful tools for develop not only mathematical models, but also to create a good 3-D interface that allow to control the virtual robot with the real robot, all join on blocs.

With this single application, a complete integration of simulation, animation and real time management for the Khepera II Robot was made and let us to explore the real behavior of it in a virtual environment.

## II. MATHEMATICAL MODEL OF THE KHEPERA II ROBOT

The Khepera II is a mobile robot based on differential configuration, it means that the speed of its center of mass it's a difference between the speed of the wheels.

This configuration gives us an easy way to maintain the robot on the same direction, but it's difficult to calculate the variations of the speed of the motors which can vary highly its direction.

The parameters of the differential configuration for the robot are the followings:

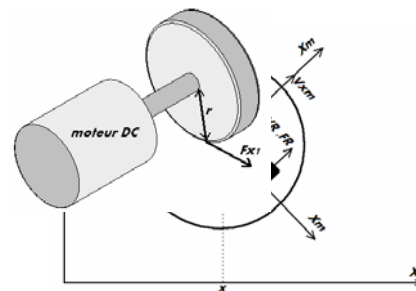


Fig. 1 Parameters of the mobile robot

Where;

- X: position in X axis on a general reference mark
- y: position in Y axis on a general reference mark
- L: longitude between the two wheels of the robot
- R: ratio of a wheel of the robot
- m: center of mass
- F: force which makes the wheel countered the ground
- $\alpha$ : angular acceleration of a wheel
- $\omega$ : angular velocity of a wheel
- J: moment of inertia
- U: tension on an engine
- F: friction force
- M: mass of the robot

Manuscript received August 15, 2007.

D. Domínguez is student of Masters Degree at the Ecole Polytechnique de Lille CO 59650, France (e-mail: Dominguez.Daniel@hotmail.com).

With these parameters, we obtain the dynamic and cinematic equations of the robot:

1.  $\dot{x} = \frac{F_L}{M} + \frac{F_R}{M}$
2.  $I_z = F_L * \frac{L}{2L_z} - F_R * \frac{L}{2L_z}$
3.  $U_L = J * \ddot{\theta}_L + f \dot{\theta}_L + F_L * r$

The inputs of the system are the two angular velocities wishes by the robot and the outputs is the desire speed. For this it's necessary to add a feedback and a PID controller that can be represent with the following block diagram:

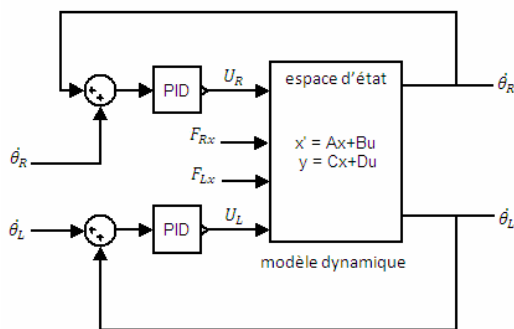


Fig. 2 State representation of the robot

With the speed that we wish, we can obtain the cinematic model of the robot, so we made the transform of coordinates which give us the following geometric representation and finally the cinematic equations:

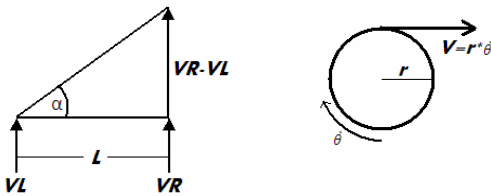


Fig. 3 Geometric model of the robot

$$\dot{x} = \frac{VR+VL}{2} * \cos \alpha = \frac{r}{2} * \dot{\theta}_R + \dot{\theta}_L * \cos \alpha$$

$$\dot{y} = \frac{VR-VL}{2} * \sin \alpha = \frac{r}{2} * \dot{\theta}_R - \dot{\theta}_L * \sin \alpha$$

$$\dot{\alpha} = \frac{VR-VL}{2} = \frac{r}{2} * \dot{\theta}_R - \dot{\theta}_L$$

Finally the following equations represent the robot's position on any time:

$$x = \int_{t_0}^{t_1} \frac{r}{2} * \dot{\theta}_R + \dot{\theta}_L * \cos \alpha * dt$$

$$y = \int_{t_0}^{t_1} \frac{r}{2} * \dot{\theta}_R - \dot{\theta}_L * \sin \alpha * dt$$

$$\alpha = \int_{t_0}^{t_1} \frac{r}{2} * \dot{\theta}_R - \dot{\theta}_L * dt$$

After find this equations, we can insert them into blocks using Matlab's Simulink Toolbox. This toolbox allows us to control and program the equations using all the math facilities of Matlab.

### III. VIRTUAL DESIGN OF THE MOBILE ROBOT

There are a lot of ways for design the virtual robot with a very realistic performance, for this work, the virtual robot was designed using Maya, that is a very powerful 3-D software and then the file was converted into \*.vrm extension for then import it into Matlab's VRML Toolbox.

The real robot and its virtual design are showing on the following picture.

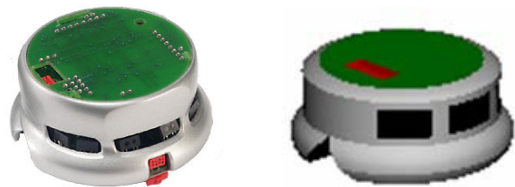


Fig. 4 Real Robot and Virtual Robot

Finally, the virtual model of the robot was integrated into a virtual space which simulates its work's environment. This was useful for made the space restriction for controlling the robot for its applications.

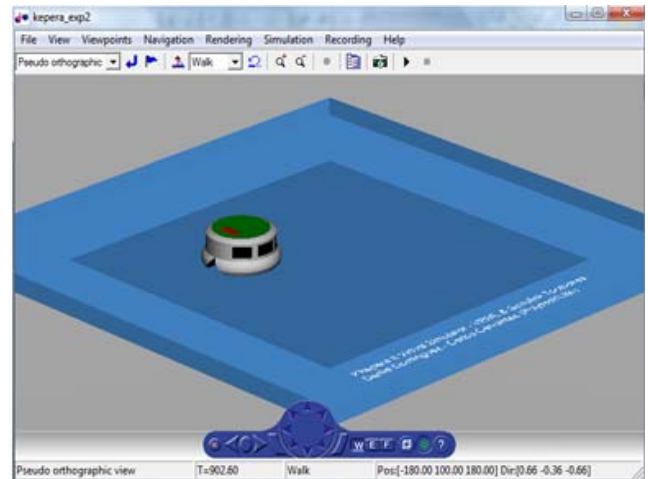


Fig. 5 Robot on its virtual environment

### IV. VIRTUAL SIMULATOR

The Virtual Simulator basically integrates the 3 important aspects for controlling the robot; the mathematical model of

the robot developed on Simulink, the virtual robot on its environment created on VRML and the communication via Bluetooth with the integrated module that has the Robot Khepera II, all join on blocks.

The Virtual Simulator allows us to control the real robot and visualize its behavior on the screen giving us an idea of its position, movement and coordinates.

The complete scheme of the Simulink blocs is the following:

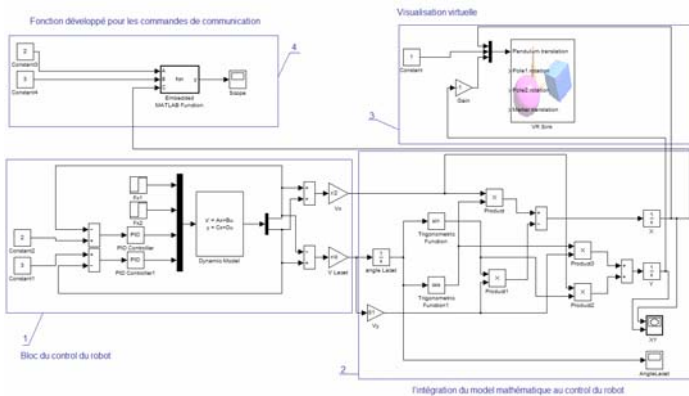


Fig. 6 Integration of the Simulink blocs of the Virtual Simulator

## V. ANALYZES AND RESULTS

This is a first approach of the application of Matlab's Toolboxes of Simulink and VRML. The main objective with this project was to give an excellent performance of the virtual robot based on the dynamic, cinematic and geometric model of the real robot. It is important to notice that applying VRML, is possible to recreate very realistic robots and with the joins of the blocs on Simulink, it is possible to show a real behavior of what is happening on the real world.

This kind of applications are open to analyze on virtual environments real characteristics of many objects, in this case, the analyze of the behavior of the Robot Khepera II based on all its mathematical model, for then controlling with a good performance on the virtual simulator.

Finally with VRML it is possible to develop a very nice GUI and it also has the advantage of be easy to integrate with other programming platforms.

## VI. CONCLUSION

This work showed one way for develops Virtual Simulators integrating various platforms of programming. Perhaps the most important advantage with the integration of VRML language and Simulink is the easy way to interact on the virtual environment.

Also with VRML is possible to develop Virtual Simulators and interface them via internet with the real robots, giving us a very extend field of applications like entertainment, medicine, exploration and others, due to the good performance on the development of the Graphics User Interface for controlling real robots with Virtual Simulators and vice versa and applying all the mathematical model of the robots on Simulink for having a very realistic interaction.

## ACKNOWLEDGMENT

D. Domínguez thanks Mr. Rochdi Merzouki, professor of robotics of the Polytec'Lille for the contribution on the mathematical analysis for the work.

Also the author would like to thank Mr. Carlos Cervantes, engineer on mechatronics for his help on the simulink connection development.

Finally the author would like to thanks Mss. Lorena Domínguez, engineer on multimedia for the development of the virtual robot.

## REFERENCES

- [1] K-Team corp. [www.k-team.com](http://www.k-team.com)
- [2] Khepera II Robot User Manual. Available: <http://www.k-team.com>
- [3] The MathWorks Matlab. Available: <http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml>
- [4] Virtual Reality Toolbox. Available: <http://www.mathworks.com/access/helpdesk/help/toolbox/vr/>
- [5] Installation instructions of Matlab scientific software package. Available: <http://shum.huji.ac.il/cc/mat/vrml.html>
- [6] Virtual Reality Toolbox Release Notes. Available: <http://www.isds.duke.edu/comp/internal/matlab/base/relnotes/virtual.htm>
- [7] Creating Standalone Virtual Environments. Available: <http://www.mathworks.fr/company/newsletters/digest/2007.html>
- [8] Cortona VRML Client. Available: <http://www.parallelgraphics.com/products/cortona/>

**Daniel Domínguez** was born in Bogotá, Colombia on September 26 of 1982. Bachelor on Mechatronics Engineering from the "Universidad Militar Nueva Granada" at Bogotá, Colombia (2005). Actually is studying Masters Degree on Automation at "Ecole Polytechnique de Lille" in Lille, France. His major field of study is Robotics.

Daniel worked as an International I.T Senior Engineer at Hardsoft in Shanghai, China, as a researcher for RFID technologies. He's involved with the applications of I.T and robotics solutions.

Mr. Domínguez is also member of the ASME (American Society Mechanical Engineers).