Haptic Rendering of the Acceleration of Particles Moving Along Pre-specified Paths

Francisco Roman Peña de la Rosa
Departmento de Ingeniería Mecatrónica
Instituto Tecnológico de Toluca
Estado de México, México
fproman393@gmail.com

Dr. Daniela Constantinescu
Department of Mechanical Engineering
University of Victoria
Victoria, BC, Canada
daniela@me.uvic.ca

Abstract

This paper describes a graphical virtual environment developed in MATLAB/Simulink and the Virtual Reality Toolbox, which represents a study of the rigid body motion in planar motion specifically a rolling cylinder “A” and a cylinder “B” which is weld in the first and is consider as a particle because of the conditions (rA≈0, mA≈0). The approach of this project is to show the user how the acceleration of a particle changes when it is moving along some path through its rectangular components (tangential and normal), and the virtual environment will enable the users to feel the acceleration directly through the forces applied. Thus, this systems is going to allow students enrolled in the Dynamic class to feel the direction and the magnitude of the acceleration under conditions defined by user. This project is based in pre-specified to a cylinder, however it can changes to a different shape using the GUI to define the parameters of the simulation.[1]

Keywords: haptic rendering, virtual environment, MATLAB/Simulink, haptic interaction, haptic device, haptic control architecture, real-time simulation, haptic interaction, computer graphics, dynamics, particle, rolling cylinder.

Project and Goals

*****To use a Novint Falcon haptic device to allow second year students enrolled in the Dynamics class to feel how the direction and the magnitude of the acceleration of a particle changes as the particle moves along some path.

*****To build a graphical virtual environment that shows a cylinder and a rolling cylinder (particle) moving along some 2D path.

*****To build a GUI which allows the user to select the path, how the particle moves along the path, and what component of the acceleration (tangential, normal or both) to feel.

*To connect the Novint Falcon to the graphical virtual environment and apply to the user a force proportional to the component of the acceleration of the particle that the user has selected.

*To use the API of the Novint Falcon and program the forces applied on the user in C/C++.

*To use a freely available software package to build the graphical virtual environment and the GUI.

Introduction

In many virtual reality applications, the user has the opportunity to feel forces and torques applied thought a haptic device and therefore, can perceive physical phenomena more realistically. In this project, the specific application of haptic rendering is focused on the motion of particles, and aims to permit users to fell how the acceleration of particle changes when it moves along pre-specified paths. Since a particle of interest is appoint on a rigid body in planar motion, each particle moves along a different path. For example we can imagine a cylinder with a center of mass G and a tangential point on the ground A. When the cylinder rolls, the acceleration of point A changes both in magnitude and in direction as the “particle” A moves along some path.

The acceleration can conge not only because of the shape of the path, but also because of the

[1] This work is supported by the Instituto Tecnológico de Toluca and COMECyT (Consejo Mexiquense de Ciencia y Tecnología). The Author is with the Department of Mechatronics Engineering, Instituto Tecnologico de Toluca, Av. Tecnológico S/N Ex Rancho la Virgen Metepec, Mex, and within the Department of Mechanical Engineering at the University of Victoria, Assistive Robotics and Mechatronics Laboratory, Victoria, BC.

increasing or decreasing speed of A. Therefore, the user can feel how the acceleration changes using a virtual environment that shows the particle moving along the path, through a GUI which allows the user to select a path and what component of the acceleration (tangential, normal or both) to fell. Finally, connecting the Novint Falcon to the graphical virtual environment and applying to the user a force proportional to the component of the acceleration of the particle that the user has selected will enable to fell the acceleration directly.

1. Acceleration of particles moving along some path

In the planar rigid-body motion all the particles of a body move along paths which are equidistant from a fixed plane. They are: a) path of rectilinear translation, b) path of curvilinear translation and c) general plane motion. For every one of these types of paths, the behavior of the particles is different. For example when we talk about translation the motion occurs when a line in the body remains parallel to its original orientation throughout the motion and it can be a rectilinear translation or a curvilinear translation, which depends of the paths of motion. Otherwise, the rotation about a fixed axis, which describes the general idea of this project. When a rigid body rotates about a fixed axis, all the particles of the body, except those which lie on the axis rotation, move along circular paths. In this case, the cylinder rotates on the “x” axis, and at the same time it represents the center of mass G. In other words the tangential point A consider as a particle describes a curvilinear behavior in its displacement while the point G just a rectilinear line.

The acceleration in each point has a different behavior, for example in the case of point G, the line of action of α (angular acceleration) is the same as that for ϖ (angular velocity), however its sense of direction depends on whether ϖ is increasing or decreasing. But in the motion of point A it travels along a circular path of a specific radius (r) with center at point G. This path is contained within the shaded plane shown in top view and the acceleration of the point A can be expressed in terms of its normal and tangential components.

However, the shape and the material are important factors in the acceleration of the particles and in the same way the path the object is moving along. Also is important to consider if the cylinder rolls without slipping or with slipping, because it means that the force of friction is going to change the velocity and as a consequence the magnitude of the acceleration. So considering a rolling cylinder without slipping, the bottom of it (the point of contact) must be momentarily at rest relative to a fixed observer in this case that “fixed observer” is the point G.

1.1 Mathematical modeling of rolling cylinder

Let us consider, for instance, a rigid, homogeneous cylinder with mass m and radius R, lying on a horizontal surface, as shown in the figure “write number”. The velocity of the contact point, Vp, and the velocity of the center of mass Vc, are related by Vp= Vc+ ω×R, where ω is the angular velocity with respect to the axis of the cylinder and R is the position vector of point P relative to the center of mass.

The Free Body Diagram of the rolling cylinder shown as follow:

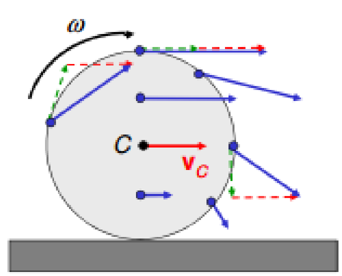


Figure 1 Different points on a rolling cylinder have different velocities (blue vectors). The velocity of each particle (blue) is the velocity of the centre (red) plus the velocity rw relative to the centre (green). The point in contact with the ground is momentarily stationary.

To study the motion described by the cylinder we must consider two phases which are clearly different. In the first a frictional force acts on the cylinder in direction opposite to the motion of the center of mass of the body.

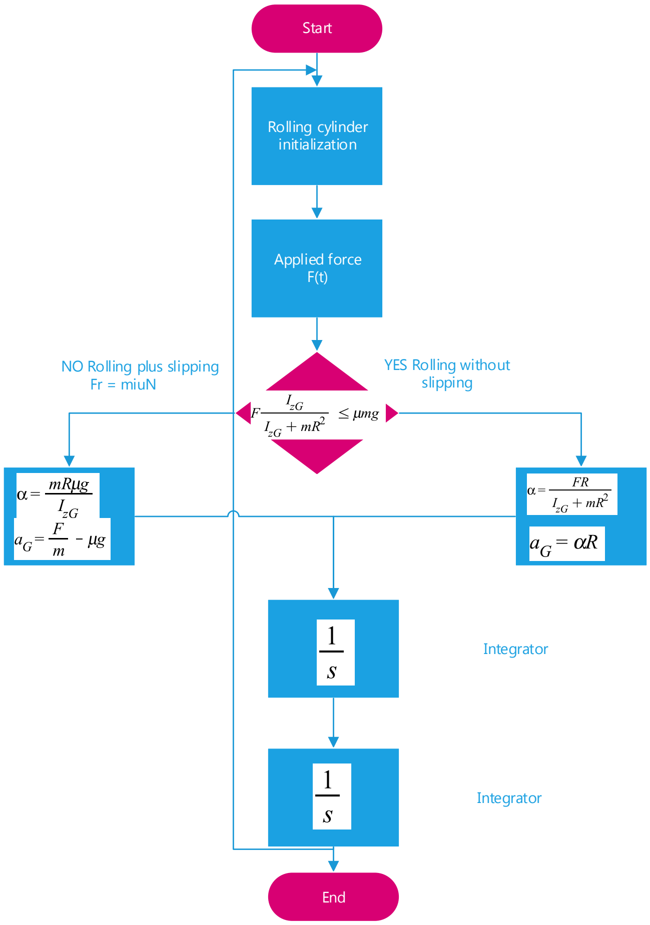


Figure 2 Flowchart for the code implementation

The center of mass is considered as the point G. And the tangential point with the ground is considered as a particle and called point A.

$$I = \begin{bmatrix} I_{xxG} & I_{xy} & I_{xz} \\ I_{yx} & I_{yyG} & I_{yz} \\ I_{zx} & I_{zy} & I_{zzG} \end{bmatrix} \tag{1}$$

Assume rolling without slipping, where ($F_R \le \mu N$, $a_G = \alpha R$) and considering the FBD shown below, the motion equations of the cylinder are as follow:

$$F - F_R = ma_G \tag{2}$$

$$N - mg = 0 \tag{3}$$

$$-FR = -I_{ZG}\alpha - ma_G R \tag{4}$$

Since $F_R \le \mu N$, so these equation are define as follow:

$$F - \mu N = ma_G \tag{5}$$

$$N - mg = 0 \tag{6}$$

$$-FR = -I_{zG}\alpha - ma_G R \tag{7}$$

Thus:

$$a_G = \frac{F - \mu mg}{m} = \frac{F}{m} - \mu g \tag{8}$$

$$\alpha = \frac{FR - mR\left(\frac{F}{m} - \mu g\right)}{I_{zG}} \tag{9}$$

$$\alpha = \frac{mR\mu g}{I_{zG}} \tag{10}$$

So the equations to define the angular acceleration (point G) and the linear acceleration (point A) under rolling without slipping conditions are as follow:

$$\alpha = \frac{FR}{I_{zG} + mR^2} \tag{11}$$

To calculate the angular acceleration of point A (consider as a particle)

$$a_G = \alpha R \tag{12}$$

To calculate the linear acceleration of point G (center of mass)

So

$$F_R = F - ma_G \tag{13}$$

$$F_R = F - mR\frac{FR}{I_{zG} + mR^2} \tag{14}$$

$$F_R = F\frac{I_{zG}}{I_{zG} + mR^2} \tag{15}$$

Thus

$$F\frac{I_{zG}}{I_{zG} + mR^2} \le \mu mg \tag{16}$$

2. System components

2.1 MATLAB/Simulink/Real time Workshop

MATLAB is used as a base for development of the haptic interface, it is an easy to use program for development, analysis and visualization of algorithms and numerical computations. Simulink is a block library that runs on MATLAB, it provides a model based programming language for simulation and analysis. The programming takes place in a graphical environment where the algorithms can be simulated and tested. Simulink is a useful tool

to avoid experimental set up and fast simulated results.
The Real Time Workshop for MATLAB, which is building code from the blocks during the simulation. Then this code can be used for real time applications in conjunction with models created by Simulink.

2.2 Virtual Reality Toolbox

The Virtual Reality Toolbox (VRT) for MATLAB is used for 3D rendering of simulated objects. It can either be implemented as a Simulink block or in low level MATLAB programming.
The Virtual Environment (VE) is built with the VRML-editor and the V-Realm Builder, where each geometrical object is defined as a node in the VRML scene graph; each node contains fields, which can be reached from MATLAB. For example in the case of the cylinder is a node that contains the field translation. To this field a signal can be sent in from MATLAB/Simulink to perform graphic rendering of real time translation of the cylinder, that is how I did it this project. The VR Toolbox is usually used to demonstrate simulated signals from Simulink 3D objects, but in this case real time signals are used instead these. A scrip in the MATLAB workspace initialize the parameters of the cylinder and the environment, and transfer the signals with the VR Toolbox to the VE. Finally for the visualization of the VE, the user can use the Visualization Toolbox or the VRML-viewer, where he analyses: rotation, translation, scale, zoom and different properties of the VE.

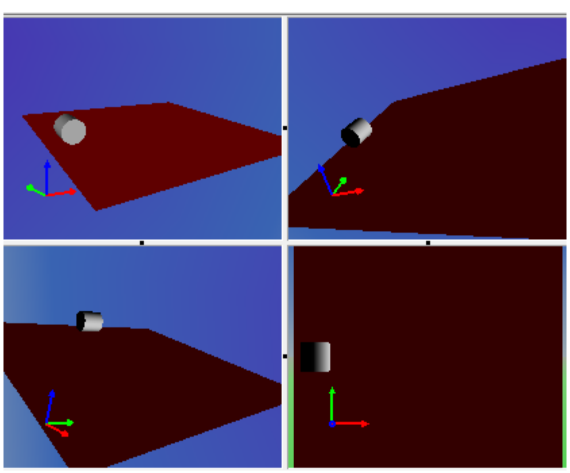


Figure 3 Virtual Environment programed in MATLAB Virtual Reality Modeling Language.

2.3 Haptic Technology and the Novint Falcon

There are in practice a number of different force feedback devices on the market, some as supporting and assistive platforms for disabled users and other as games platforms. The Nintendo Wii joystick and the Play Station controller are the most famous and commercially purchased games platforms that offer force feedback to the player. This new way, which enables users to actively interact with the virtual environment, has changed dramatically the way the former interact with the game environment and the high level of pleasure they reach during the play. Another force-feedback new generation of games consoles is the Novint Falcon introduced by Novint Corporation. The Novint Falcon is the consumer's first 3D device which allows user to "feel" what is happening in the virtual environment.
In the actual use, the user grasp the ball-shaped grip and the variety of joints on the arms allows the actual grip to move just like a mouse, though oriented vertically with an additional axis of movement of in and out. The Falcon allows users to control and interact with games in more realistic way, allowing the development of real physical skill and muscle memory.

Figure 4 The Novint Falcon Device

3. System Overview

A virtual environment is built in the MATLAB VR Toolbox. The procedure of the algorithm is as follows:

*Define a virtual environment in the MATLAB VR Toolbox.
*Develop a scrip where the user can initialize the dimensions of the cylinder: mass, length, radius of gyration and position.
*Develop an algorithm to calculate the angular acceleration of the tangential point A, which is consider as a particle and the velocity of G, which the cylinder's center of mass.
*Check the type of rolling movement on the cylinder.
*If "NO" rolling plus slipping.
*Data visualization on the Real Time Workspace
*Calculate the acceleration (alpha) and velocity (a) values and send the signals to Simulink model for data visualization.
*Graphic rendering of the cylinder moving along some defined by user and a virtual environment for interaction between the user and the application.

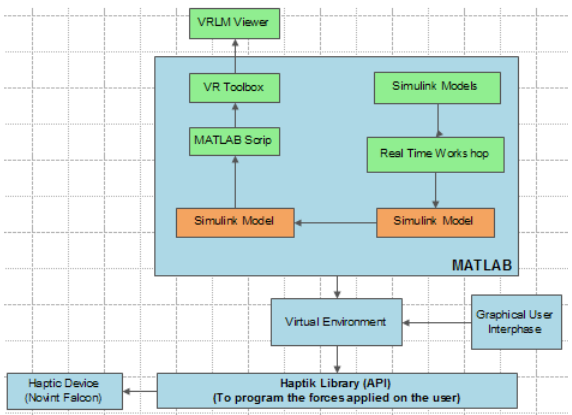


Figure 5 System Overview of the MATLAB/Simulink model

4. Haptic rendering of a solid body and graphical virtual environment in MATLAB/Simulink

To describe how haptic rendering works, we can imagine whatever shape located at the origin of a 3D virtual environment, for example a cylinder with a fixed axis "Z", in this case the user only can interact with the virtual object using a point which represents the end point of the haptic device. In the real world, this is analogous to interact with the cylinder using the hand or fingers. Using the Matlab Visualization Toolbox the user can freely explore the 3D space and through the Virtual Reality (VR) toolbox for MATLAB which is used for 3D rendering of simulated objects. It can either be implemented as a Simulink block or in low level MATLAB programming. The virtual environment is built up with the VRML editor, where each geometrical object is defined as a node in the VRML scene graph. Each node contains fields, which can be reached from MATLAB, for example in this case a cylinder is a node that contains the field translation and rotation. To these fields a signal can be sent in from MATLAB/Simulink to perform graphic rendering of real time translation or rotation that is how I did in this project. The VR toolbox is usually used to demonstrate simulated signals from Simulink as 3D objects. But in this case, I have used it to simulate haptic rendering of rigid bodies: the cylinder A and the cylinder B that is weld in the first one and is considered as a particle.

4.1 Implementation details

Figure 5 shows the graphics rendering (virtual environment) and the haptic rendering process in more detail. Note that in addition to the GUI and the haptic device, the keyboard and mouse have been added to the diagram to set de dimensions in the GUI by the user and control the through it control the interaction between the VE and the user.

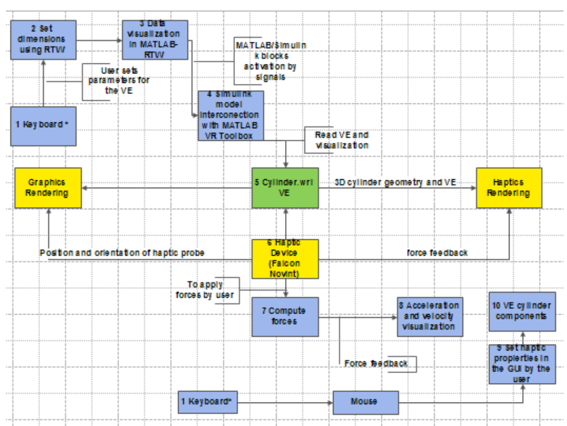


Figure 6 Implementation details of the Virtual Environment

5. Simulation (Rolling Cylinder)

I used the Virtual Reality Toolbox for MATLAB/Simulink, SimMechanics and the Real Time Workshop to simulate the rolling cylinder behavior subjected to: a) without slipping condition and b) plus slipping condition. The modeled rolling cylinder virtual environment is consisted of one plane which is consider as the ground, a cylinder which is weld with the ground in a tangential point “A” as a constrain, and moving along a rectilinear path using the plane XZ as a prismatic joint. Therefore the cylinder can moves along a path from two arbitrary points defined by user.

The cylinder has the following parameters: 500 gr mass and 10 cm radius for the first simulation, because R (radius) and m (mass) can be initialized through a script in the MATLAB/Real Time Workshop by user, which means that the user can define the conditions of the simulations before it starts.

By the help of the Script created, one can update automatically the values in the Simulink blocks, and through it all the values in the virtual environment are going to change before the next simulation starts. Thanks to Virtual Reality Toolbox/Simulink, one can design dynamic system and after the simulation, find its outputs such as linear velocity and linear acceleration for the point G (center of mass) and the angular velocity and angular acceleration for the point A (tangential point).

The rolling cylinder Simulink model as shown in Figure 7, represents the particle dynamic behavior on the virtual environment. Users can use double click to change the simulations parameters as Figure 10 shows. However though the script which controls the rolling cylinder initialization, user just set the input values as the interface requires and automatically the system updates the values in Simulink and in the virtual environment.

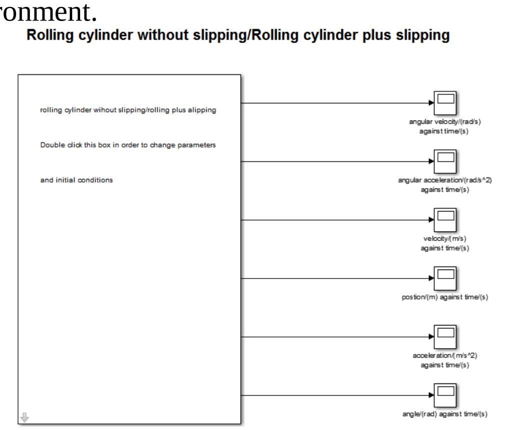


Figure 7 Simulink model for the rolling cylinder system dynamics

Figure 8 shows the Simulink/SimMechanics model which is the dynamic representation of the rolling cylinder into the virtual environment, this model describes the rigid body kinematics and how the bodies are joined.

There are different types of blocks, one is used to represent a rigid body with properties customized by user, but in this case we are using the script values as a default initial variable state. Another important type of block allows user to define the degrees of freedom on the simulation. Finally we have the sensor blocks that are used to sense the motion of a rigid body, or the effect of an applied force.

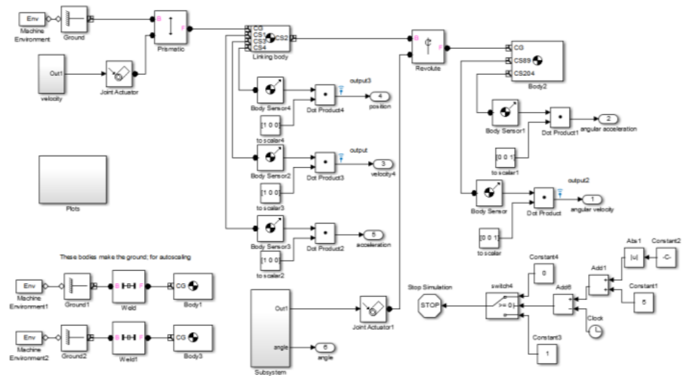


Figure 8 Simulink/SimMechanics Subsystem used to the VE conditions

Figure 9 and Figure 10 show the system parameters

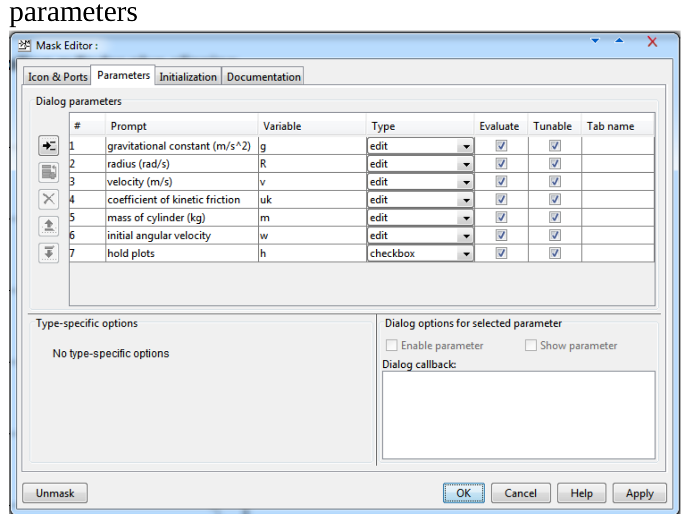


Figure 9 Dialog box to control the cylinder variables.

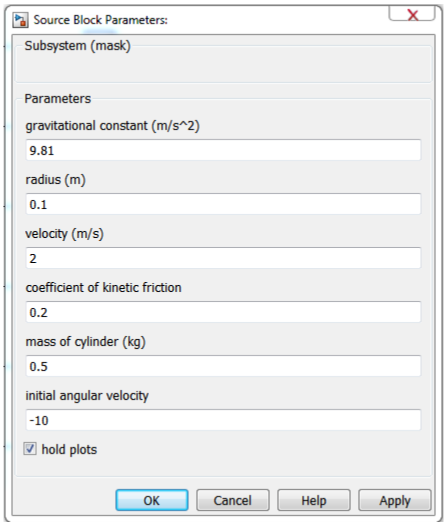
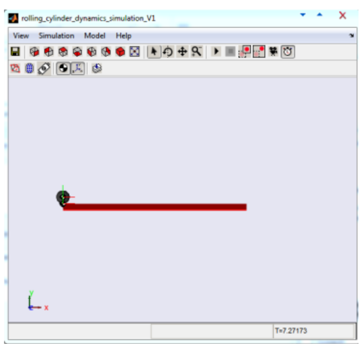
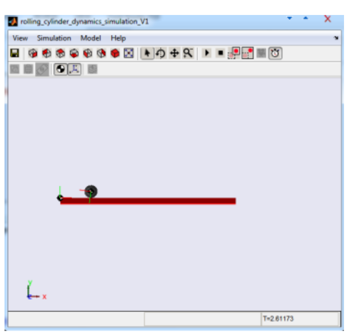


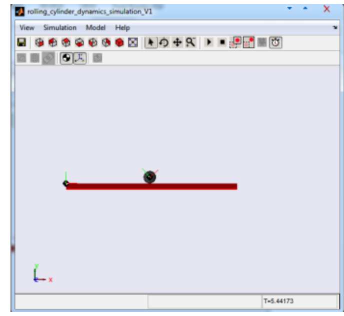
Figure 10 Block parameters subsystem setting



a) Rolling cylinder at the initial position

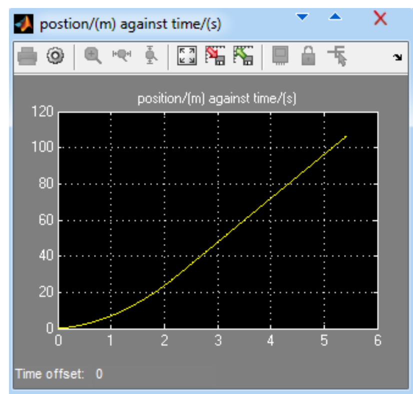


b) Rolling cylinder in movement in a fixed axis Z.

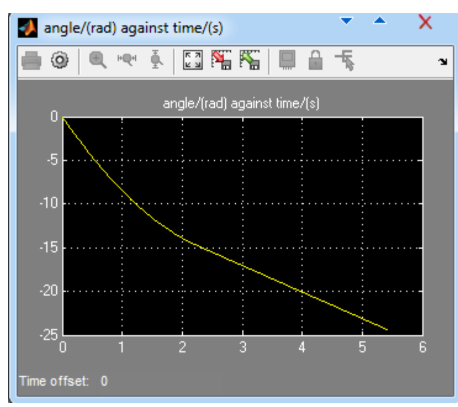


c) Rolling cylinder almost at the end of the rectilinear path

Figure 11 Simulation of the rolling cylinder moving along a rectilinear path (front view)



a) Position (m) against time (s)



b) angle (rad) against time (s)

Figure 12 Simulation of the rolling cylinder with R=0.1 m, m=0.5 Kg

6. Conclusions and Future Work

It is hoped that this brief approach of haptic rendering though the simulation of dynamics systems in MATLAB/Simulink

The Virtual Reality Simulation is better than traditional substation simulation in visualization, interaction, immersion and imagination, because allow users to feel in real time how the "avatar" in the virtual environment experiments changes on some of the following variables: position, angles, velocity, acceleration, length, weight though the haptic feedback of the haptic device. The results show that the virtual environment is going to be updated with the information customized by users on the MATLAB script, which is an algorithm that calls different functions to calculate and estimate the response of the cylinder in the virtual environment and send this information to users though the Falcon Novint.

Future work could include some of the following suggestions:

* To change the cylinder as a particle for a tangential point using the Virtual Reality Toolbox and the V-Realm Builder.

* Development of the GUI to allows the user to select the path, how the particle moves along it and what component of the acceleration to feel.

* To use the Haptic Immersion software in combination with Android to develop an application which allow users to feel the haptic feedback through a gadget as a mobile application.

* To improve the MATLAB software using an application to control the function of the haptic device.

* To finish the project in a full way and the achievement of all the goals defined.

7. Acknowledgements

For my dear supervisor Dr. Daniela Constantinescu who leads me to the way of academic research. Thank you for all the unconditional support, guidance and encouragement that you have always during the course of this work.

Thank you for give me the opportunity to spend time in your Lab working on this project, I really appreciate to trust on me and for your help en every single moment of the project.

I also acknowledge the support of my University Instituto Tecnológico de Toluca and COMECyT, and the people who supported this project as well: M.C. Gloria Irene Carmona Chit, M.C. Mauro Sánchez Sánchez, Dr. Fidel Alejandro Camarena Vudoyra and Ing. Margarito Luna Mejia. All my colleges in ITTol, for their support and good humor and to my family for everything thanks a lot. Finally I would like to thank my family for among many things, thanks for your love, patience, and support you are constantly given me

8. References

[1] Virtual Reality Toolbox for Use with MATLAB and Simulink, http://www.mathworks.com/help/releases/R13sp2/pdf_doc/vr/vr.pdf

[2] MATLAB Simulink Getting Started Guide R2013, http://www.mathworks.com/help/pdf_doc/simulink/sl_gs.pdf

[3] Kenneth Salisbury and Francois Conti and Federico Barbagli, "Haptic Rendering: Introductory Concepts" pp. 24-30.

[4] K. Sal&bury: D. Brocki T. Massiet, N. Swarupf and C. Zillest, "Haptic Rendering: Programming Touch Interaction with Virtual Objects", *Artificial Intelligence Laboratory*, Massachusetts Institute of Technology pp. 123-130.

[5] M. Bergamasco. *Artificial Life and Virtual Reality*, N. Magnenat Thalmann and D. Thalmann editors, John Wiley, chapter *Manipulation and Exploration of Virtual Objects*, pages 149–160. 1994.

[6] Yukiko Inoue, *"Concepts, Applications, and Research of Virtual Reality Learning Environments"*, International Journal of Human and Social Sciences, pp. 1-6.

[7] B. Winn, *"Learning through virtual reality,"* Retrieved March 8, 2007, from http://www.newhorizons.org/strategies/technology/winn.htm.

[8] Magnus G. Eriksson and Jan Wikander *"A Haptic Interface Using MATLAB/Simulink"*,

Mechatronics Lab, Department of Machine Design, pp. 1-16.
[9] Open-Haptics, http://www.sensable.com/products-openhaptics-toolkit.htm.
[10] The Virtual Reality Toolbox, http://www.mathworks.com/products/virtualreality/.
[11] V-Realm Builder, http://www.ligos.com/.
[12] Real Time Workshop, http://www.mathworks.com/products/rtw/.
[13] MATLAB/Simulink, http://www.mathworks.com/.
[14] M. A. Srinivasan. *Virtual Reality: Scientific and Technical Challenges*. Report of the Committee on Virtual Reality Research and Development, chapter Haptic Interfaces. National Academy Press, National Research Council, 1995. http://touchlab.mit.edu/publications/1995006.pdf.
[15] MATLAB/SimMechanics, http://mecanismos2mm7.files.wordpress.com/2011/09/tutorial-sim-mechanics.pdf