H^2IL Co-simulation of Cooperative Robots based on ADAMS, MATLAB and a haptic interface

M. Bonilla-Jiménez, V. Parra-Vega, F. J. Ruiz-Sánchez
Robotics and Advanced Manufacturing Group. CINVESTAV-IPN
Ramos Arizpe, México
{jose.bonilla, vicente.parra, francisco.ruiz}@cinvestav.edu.mx
Telephone: (844)4389600 x8600

***Abstract*— MATLAB is a very popular numerical tool for simulation, however limited to what is programmed by the user. This limitation stands for a serious drawback since it is not possible to analyze a variety of physical effects. To deal with a more realistic simulation scenario it is required to analyze energy distribution, wear, flexibility, impacts, hard no linearities on remote actuation or gears and friction, to mention a few effects. All this is particularly important in robotic systems, even when running a pure academic study since simulation must comply with reality. In this paper, we address the CAE-based modeling and control of a complex cooperative robotic system guided by a human operator on line. We use ADAMS to reproduce the mechanical system to be able to handle and understand most of the phenomena acting on it, a complex controller, briefly presented and programmed in MATLAB, accounts for robustness to deal with them, then ADAMS and MATLAB are integrated in a co-simulation system. Furthermore, Hardware-In-The-Loop is explored via a haptic interface, the Falcon by Novint, to provide a kinesthetic coupling to the human user, which in practice introduces, on line, the desired trajectories generated by the human to the whole system. In this sense, it is really a Human-In-The-Loop System, that, together with the Hardware-in-the-loop techniques used in the co-simulation presented in this paper a Hardware-Human-In-The-Loop or H^2IL co-simulation, as coined in this work. Results are promising so as to provide useful parameters for the basic and detailed engineering specifications of the final real system.**

Keywords: co-simulation, Hardware-in-the loop, Human-in-the-loop, closed-loop control

I. INTRODUCTION

Modern scientific and technological paradigms are based on the use of numerical simulation as an important source of information to analyze the main properties of a system. The validity of simulation results mainly depends on the quality of the model, designed to describe the system subject to specific operation conditions according to hypothesis, assumptions and the numerical methods. In particular, differential equations are used to describe a broad class of physical phenomena in a given domain, and together with new computing facilities, simulation has became a fundamental tool in modern engineering. However, the size and the descriptive equation of these models are very sensible to the increasing complexity of reproducing real systems, becoming unpractical when custom-made programing is used.

I-A. *MATLAB-based modeling*

MATLAB has become in a popular simulation tool for engineers and scientist in development, however it is limited to what is programed by the user. In robotics and control areas, there exist mechanical systems that can not be easily modeled, some examples are humanoids, cooperative anthropomorphic robotic arms and constrained robots, due to their high degrees of freedom and techniques used to model them in the interaction with the environment or the physical phenomena like energy distribution, wear, flexibility, impacts, hard no linearities on remote actuation or gears present in them. These phenomena have to be considered when it is wanted to develop a more realistic physical simulation, for this reason it is necessary to take into account a more powerful simulation tool.

I-B. *CAE-based modeling*

Nowadays there exist many Computer-Aided Engineering (CAE) software packages that enabled the designer to draw the mechanical systems, instead of program it, and the systems is ready-to-make simulations dealing with phenomena already mentioned. Automatic Dynamic Analysis of Mechanical Systems (ADAMS) (Lianqing Yu, et al, 2008) stands as a general purpose program useful to analyze systems undergoing large non-linear displacements under the effect of non-linear force and input (Elliott,). ADAMS can be categorized as a general purpose numeric code utilizing a non-minimal set of coordinates to develop the equations of motion. It uses stiff integrators to solve these equations and sparse matrix algebra to solve the linear algebraic equations in its innermost computational loop (Zhen, et al, 2008).

I-C. *Co-simulations-based experiments*

Cooperative simulation, or co-simulation for short, is a simulation methodology allowing simultaneous simulation of individual components of a system in different simulation tools exchanging information in a cooperative manner (Wei, et al, 2008). In Robotics, co-simulation is a very useful concept for designing process because it allows the use of the best simulation tools for a specific problem to recreate the main physical phenomena involved in the function of the system and provide the engineers of a virtual prototype of the entire system and the main information about its behavior under specific conditions, to determine the final specification of a new device. Moreover, co-simulation can be used to help control designers providing a virtual environment to analyze the dynamic behavior of the closed loop system submitted under normal and extreme operational conditions (Zhen, et al, 2008), (Zhang and Jin, 2008), (Zhang, et al, 2009).

I-D. Hardware-in-the-loop (HIL) and Human-in-the-loop (HIL)

The condition of a Simulation platform integrating numerical I/O data from real devices is described as Hardware in the loop, HIL, i. e., when some of the state variables of the simulated closed loop system come from physical systems by an interface through real sensor measurements by an interface. This concept was initially used in the fields of defense and aerospace engineering and now, HIL, is used by designers and testing engineers to evaluate and validate components during the development process. Extending this process, when some of the signals on the simulation systems come from a human interaction with the system, we are talking about of Human-in-the-loop, also designated by HIL but differentiated by the context of its application. This concept is used in training process and, recently it is used in human robot interaction and haptic interfaces.

I-E. The Proposal H^2IL

In this paper, we propose a co-simulation system which mix the concepts of Hardware-in-the-loop and Human-in-the-loop, designated as H^2IL co-simulation. The aim of this system is to provide of a realistic simulation of a cooperative platform of two manipulators operated by a human through haptic interfaces. This co-simulation integrates two main blocks, the first block, containing the CAE model of the collaborative robots running in ADAMS, and the second one, containing the close loop controller and the interface for the haptic device, programmed in MATLAB. We briefly present the control strategy composed of two operation modes, free and constraint motion, in both cases the controller were obtained using modern control approaches. We include some results obtained from the proposed H^2IL co-simulation of cooperative robots handling a rigid body, and we conclude with some final comments, remarking that these results will provide the data for the basic and detailed engineering specification for a final design of a real cooperative platform.

I-F. Organization

The rest of the paper is organized as follows. Section II describes the proposed H^2IL co-simulation and the blocks which is composed. Section III presents simulation results of cooperative robots done with the proposed H^2IL co-simulation. Finally in section IV some conclusions are presented.

II. GENERAL DESCRIPTION OF THE PROPOSED H^2IL CO-SIMULATION PLATFORM

The H^2IL co-simulation proposed in this paper is presented in the figure 1. It consists of two main blocks:

1. *ADAMS Block*.- is a CAE virtual model of two cooperative robot manipulators which simulate their numerical dynamic behavior based on Computer-Aided Design (CAD) model.
2. *MATLAB Block*.- is a set of MATLAB functions containing the close loop controller to command the system using a Position and Force-Position control strategy for free and constrained motion, respectively, and a MATLAB/SIMULINK s-function to add human action into the co-simulation determining the desired cartesian positions of the manipulated object through a Novint Falcon haptic interface. This block also includes an inverse kinematic solver to obtain the desired joint position of the robots from the desired Cartesian coordinates of the object.

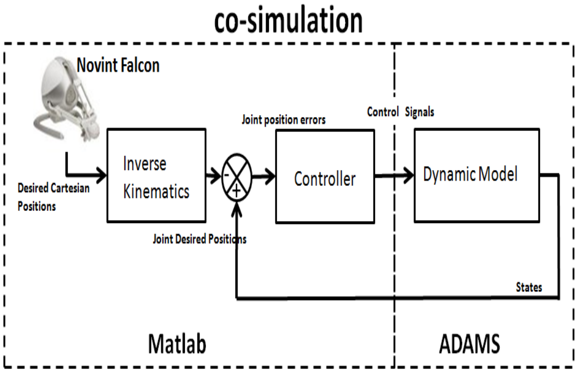

Figure 1. General overview of the H^2IL co-simulation

II-A. ADAMS-based virtual model

Before modeling the cooperative system, a 3D mechanical CAD model is introduce to ADAMS. This can be done using any CAD platform; in our case, we decided to use SolidWorks because of its facilities to do assemblies fast and, to export designs to different cad formats . Two robotics manipulators are considered which are five degrees of freedom Mitsubishi RV-M1. In the CAD model we included joint limits, constraints among links and material properties of each link according to the information contained in the user manual (Mitsubishi, 1991). In figure 2 and table I we include dimensions and mechanical properties used in the CAD model.

The CAD model developed in SolidWorks was exported into ADAMS in order to include contact force sensors, gears, ratios, joint friction, gravity forces, actuators, state variables (position, velocity and contact force) and control inputs. We also included in ADAMS an aluminum cube because the task of robots is to manipulate it. This model is presented in the figure 3. Finally, using ADAMS/Controls

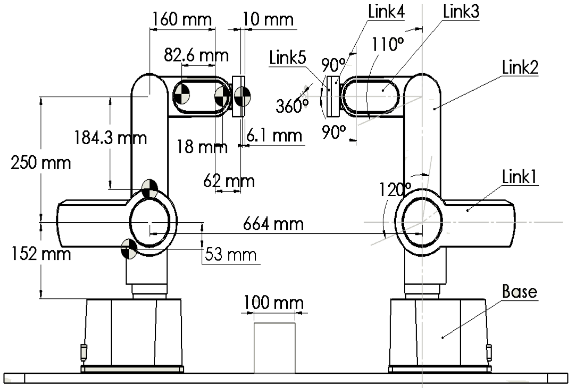

Figure 2. Cooperative robotic platform in SolidWorks

TABLA I

PRINCIPAL MOMENTS OF INERTIA

Link	Weight (kg)	Principal moments of inertia Ixx, Iyy, Izz (kgm^2)
1	4	0.032403, 0.040828, 0.061743
2	6	0.018519, 0.071117, 0.072460
3	4	0.004886, 0.017788, 0.016656
4	2	0.001292, 0.002254, 0.002662
5	1	0.000380, 0.000380, 0.000733

Plugin we generate a MATLAB/Simulink block of the model obtained with ADAMS. This block is fundamental for the co-sumulation because it interfaces ADAMS with MATLAB.

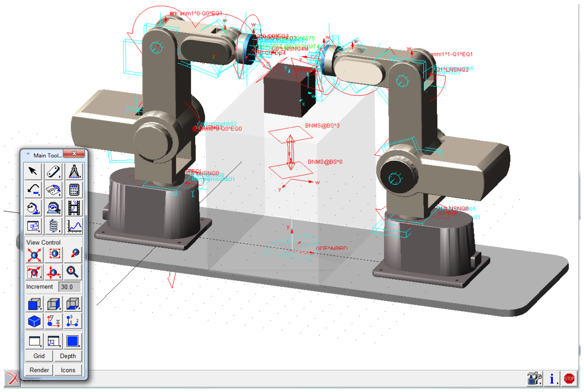

Figure 3. Cooperative robotic platform in ADAMS

II-B. The control block

The dynamic model of i rigid $n-$link serial robot manipulators in free motion, with all actuated revolute joints described in joint coordinates, is given by the equation 1. And in the same way the equation 2 describes the robot dynamics when these are contacting a surface in m points.

$$
\begin{aligned}
H_i(q_i)\ddot{q}_i + C_i(q_i,\dot{q}_i)\dot{q}_i + g_i(q_i) &= \tau_{fri} \qquad (1)\\
H_i(q_i)\ddot{q}_i + C_i(q_i,\dot{q}_i)\dot{q}_i + g_i(q_i) &= \tau_{coi} + J_{\varphi_i}^T\lambda_i \quad (2)
\end{aligned}
$$

Where $(q_i, \dot{q}_i, \ddot{q}_i)^T \in \mathbb{R}^n$ are the generalized joint position, velocit and acceleration coordinates of the $i-th$ robot, matrix $H_i(q_i) \in \mathbb{R}^{nxn}$ is a positive-definite inertial matrix, $C_i(q_i,\dot{q}_i) = (\frac{1}{2}\dot{H}_i(q_i) + \bar{S}_i(q_i,\dot{q}_i)) \in \mathbb{R}^{nxn}$ models the coriolis forces and $\bar{S}_i(q_i,\dot{q}_i) \in \mathbb{R}^{nxn}$ is a skew-symmetric matrix, $g_i(q_i) \in \mathbb{R}^n$ is a vector of gravitational torques, $\tau_i \in \mathbb{R}^n$ is a vector of input-controlled joint torques, $\lambda_i \in \mathbb{R}^m$ is a vector of Lagrange multipliers or contact forces, $J_{\varphi_i} = \bigtriangledown\varphi_i/||\varphi_i|| \in \mathbb{R}^{mxn}$ is an orthonormal matrix, where $\bigtriangledown\varphi_i$ denotes the gradient of the object surface vector and φ_i maps a vector onto the normal plane at the tangent plane that arise at the contact point $\varphi_i(q_i) = 0$.

The controller used in the simulation of the cooperative robots is composed of two control laws, designed in a passivity-based approach for the operational space, for two different operation conditions: free motion and constrained cooperative motion. Figure 4 shows a timeline describing the moments in the simulations that each controller is used. The control applied for free motion is dynamic sliding PID controller for tracking task of robot manipulator (Parra-Vega et al, 2003), it is a low computational cost algorithm because it is a model-free control. The main equations of this controller are:

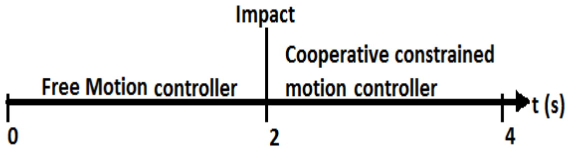

Figure 4. Timeline of control activation in H^2IL co-simulation

$$
\begin{aligned}
\tau_{fri} &= -K_d\left(S_{qi} + \gamma\int_{t_0}^{t} sgn(s_{qi})dt\right)\\
S_{qi} &= \Delta\dot{q}_i + \alpha\Delta q_i - (\Delta\dot{q}_i + \alpha\Delta q_i)(t_0)\exp^{-\beta t} \quad (3)
\end{aligned}
$$

where $\Delta q_i = q_i - q_{di}$ and $\Delta\dot{q}_i = \dot{q}_i - \dot{q}_{di}$ stand for the position and velocity tracking errors, q_{di} stands for the desired joint position vector; K_d, α and γ are the feedback gains of appropriate dimensions. This controller, according with the stability analysis in our previous work (Parra-Vega et al, 2003) guaranties exponential convergence of tracking errors. This is important because the manipulators must grasp the object before handling it, so we need to arrive to the contact point without overshooting, to avoid as much as possible, an impact phase among the robot and the object.

The control law for cooperative constrained motion, that is, the control law applied when the robots handle an object cooperatively, is a decentralized Sliding Force/Position PD Control of Cooperative Robots (Garca-Rodríguez and Parra-Vega, 2005). This control law does not require any communication among manipulators and was designed on the basis of the Joint-Space Orthogonalization Method (JSOM) which states that the exerted contact torque and the joint velocity are orthogonal to each other at the contact point, thus torques and velocities belong to two complementary subspaces (Yun-Hui, et al , 1996). This is useful because it is possible to find a mathematical description of the contact phenomena when the final effector of a robot is moving along the surface. Extending this method to analyze the robots as a cooperative system, we can decouple joint velocity and force subspaces for each constrained manipulator. Exploding this property the equations describing this controllers are:

$$\begin{aligned}
\tau_{coi} &= \tau_{qi} + \tau_{fi} \qquad (4)\\
\tau_{qi} &= -K_v \left(S_{qfi} - \gamma \int_{t_0}^{t} sgn(S_{qi})dt + J_{\varphi}^{+}(q_i) S_{pi} \right)\\
S_{qi} &= \Delta\dot{q}_i + \alpha \Delta q_i - \left(\Delta\dot{q}_i + \alpha\Delta q_i\right)(t_0)\exp^{-\beta t}\\
S_{pi} &= \Delta\dot{p}_i + \eta \int_{t_0}^{t} \Delta p_i \qquad (5)\\
\tau_{fi} &= J_{\varphi_i}^{T}(q_i)\left(-\lambda_d + \dot{S_{dfi}}\right) + \gamma_{fi}\tanh(\nu_i S_{qfi} + S_{vfi})\\
S_{vfi} &= S_{qfi} + \gamma_{fi} \int_{0}^{t} sgn(S_{qfi})dt\\
S_{qfi} &= \int_{t_0}^{t} \Delta\lambda_i dt - S_{dfi}\\
S_{dfi} &= \Delta\lambda_i(t_0)\exp^{-\beta_f (t-t_0)} \qquad (6)
\end{aligned}$$

As in the free motion controller (equation 4), the controller described above guaranties exponential convergence on tracking errors including $\Delta\lambda_i = \lambda_i - \lambda_{di}$, λ_i and λ_{di} are the contact force and desired contact force in the robot i, respectively. It is important taking into account that in this case $K_v = K_d Q_{\varphi}(q_i)$, α and γ, η, γ_{fi} and ν_i are the feedback gains of the controller, all these gains, as in the previews case, have adequate dimension.

We mentioned that using JSOM leads us to two orthogonal subspaces, one spanned by the constrained Jacobian matrix $J_{\varphi i}(q_i) = \frac{\nabla \varphi q_i}{||\nabla \varphi q_i||}$, where $\varphi(q_i)$ is the motion constraint i, referred to the contact surface, expressed in generalized coordinates q_i of the robot i. And the other, spanned for the orthogonal projection of $J_{\varphi i}(q_i)$ is $Q_{\varphi}(q_i) = I - J_{\varphi}^{+} J_{\varphi i}$ where $J_{\varphi}^{+} = J_{\varphi i}^{T}(J_{\varphi i} J_{\varphi i}^{T})^{-1}$ stands for the constrained Jacobian pseudo-inverse.

Analyzing cooperative robots using JSOM, $J_{\varphi}\dot{q}_i = p_i$ and its time integral are called constrained velocity and constrained position, respectively, and together with a well-solved planning, help us to achieve in normal direction at the contact surface and keep the contact on, and leads the errors of constrained position Δp and velocity $\Delta\dot{p}$ to 0. To be more specific on it we suggest reading (Yun-Hui, et al, 1996).

In (Parra-Vega et al, 2003) and (Garca-Rodríguez and Parra-Vega, 2005) S_{qi}, S_{qfi} are called error and force manifolds respectively, and they are the sliding surfaces induced to the system by the controllers so, it is necessary monitoring them during the simulation.

It is important to mention that in order to analyze the robustness of the position/force controller we also implemented a modified version without constrained position errors, *i.e.*, $\eta = 0$. It is necessary to be noted that this change does not affect the stability analysis, this means, the outcome is still exponentially stable. The control laws exposed in this section was selected because we have already analyzed them in previous works and we have experience applying them.

II-C. The Haptic Interface

In order to interface the ADAMS model and the operator, the use of a Novint Falcon haptic interface is proposed. It is a delta-3 parallel manipulator widely use for research on haptics because, it provides high-fidelity three-dimensional force feedback. From the haptic interface we get the desired position of the object being manipulated, this is performed using Haptik Library, a component-based open-source library which provides a hardware abstraction layer for access to haptic devices, and together with a MATLAB S-function enable the use of Hardware-in-the-loop. Figure 5 shows a description of the Hardware-in-the-loop used in the proposed H^2IL co-simulation

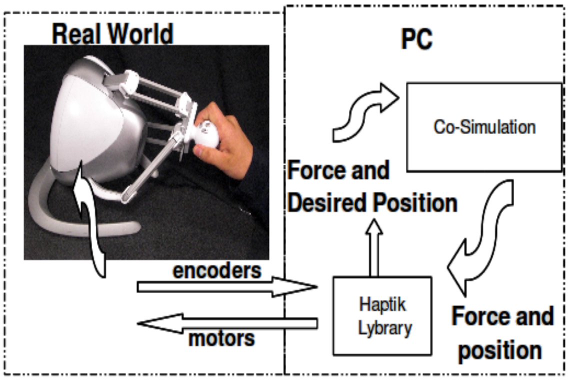

Figure 5. Diagram of Hardware-in-the-loop using Falcon

The MATLAB-block also includes a MATLAB function to calculate the inverse kinematics, that is since desired position of the object coming from the haptic interface is expressed in Cartesian coordinates, this positions must be expressed in therms of joint coordinates in order to be used for the controller. Observe that this function is not necessary if a serial robot, like PHANTOM®, is used instead of Falcon.

III. RESULTS

In order to validate the H^2IL co-simulation system proposed in this paper, we present two co-simulations of cooperative robots handling a rigid body guided by a human operator through a mechanical interface. These co-simulations were designed to grasp a rigid object and then to manipulate it cooperatively. The controllers presented in section III were tuned, for both manipulators in free motion mode, using the feedback gains: $Kd = diag\{20; 50; 30; 10; 1\}$, $\alpha = diag\{10; 30; 10; 5; 1\}$, $\gamma = diag\{,3; ,3; ,3; ,3; ,1\}$ and $\beta = 5$. This values were selected to avoid saturations of

the control signal according with the specifications of the real robots. Initial conditions of both manipulators were the same, and the object to be manipulated was located at the same distance of each other to guaranty that the manipulators arrives to its surface at the same time.
When the manipulators are in contact with the object, we switched to the position/force controller, presented in equations (4), (5) and (6) using the following feedback gains: $Kd = diag\{5; 10; 7; 5; 1\}$, $\alpha = diag\{8; 28; 20; 16; 4\}$, $\gamma = diag\{,5; 1; 1; 1; 0,1\}$, $\beta = 5$, $\beta_f = 3$, $\gamma_f = 5$ and $\eta = 10$. When the controller is switched, the desired positions of the manipulated object are taken form the haptic interface. For this simulations we considered infinite friction in the contact point to avoid that the object slides off.

Figure 6 shows the articular position errors on each joint and the contact force errors, where is clear that fast and accurate performance is achieved, we can check that during the manipulation the end effector of each manipulator remain in contact.

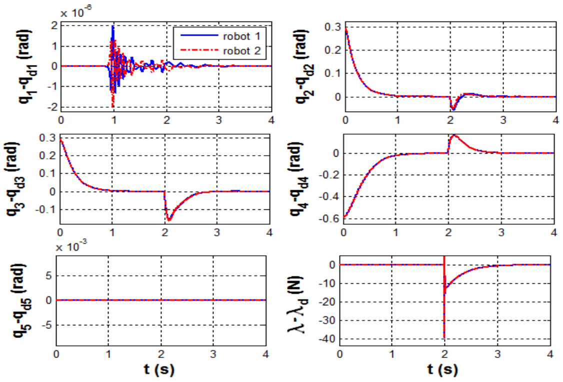

Figure 6. Articular position and force errors

Figure 7 presents the sliding surfaces S_q and S_{qf} for both manipulators, wherein we can see fast and accurate performance on the two controlled subspaces.

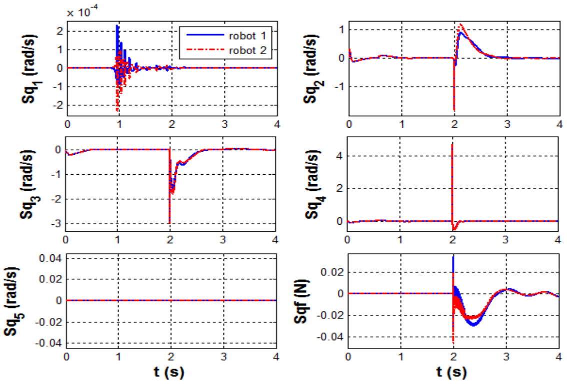

Figure 7. force and position sliding surfaces

Control signals of each manipulators are shown in the figure 8 in which we can see that there exist a peak in $t = 2(s)$, it is due to the impact of the robots with the object at this moment the contact force increase and the controllers try to lead contact force error to 0.

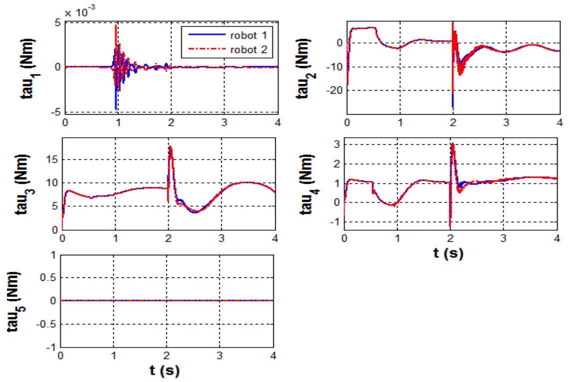

Figure 8. Control signals

As we explained in section III we also implement a modified version of the cooperative constrained control using the same feedback gains but $\eta = 0$ to evaluate the robustness of the closed loop controller face to rolling contact effects. With the controller the position, velocity and force errors do not converge at all, this is due the controller do not compensate the disturbances of the changes on the orientation as showed in figure 9, in fact the contact among the object and the robots was lost before $t = 3(s)$ so the object falls down. Figure 9 shows the object orientation at different times.

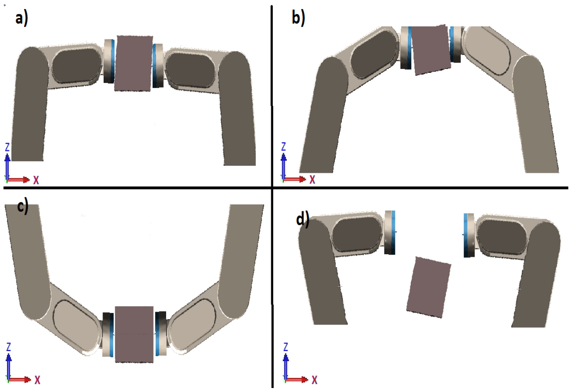

Figure 9. a) t= 2s, b) t=2.3s, c) t=2.6s d) t=3s

It should be noted that we programmed the same the simulation using MATLAB only, however, in this case we can not observe the difference among the controllers because we didn't program physical phenomena that causes that the object change its desired orientation because it is complicated, these phenomena are rolling contact and contact areas.

Control laws presented in section III shows similar performance, in fact studying stability analysis of the closed loop systems tracking errors convergence gives same results in both cases, however in the co-simulation results presented in this paper we showed that the difference among controllers, presented in section for cooperative constrained motion, is the robustness to undesired position and orientation disturbances of the object being manipulated, at least in the cased presented in this paper. Others simulation can be performed in order to evaluate robustness to more physical phenomena using the same controller or other ones. For example we can change the end effector to have contact areas instead of a point and, to evaluate if the controller used before still works for the purpose or we have to redesign it. In other hand including CAE-based modeling in co-simulations help us to do easily changes in the model being analyzed. In co-simulation realized in this paper we noted that computational cost is less susceptible to changes in model that using MATLAB only.

IV. CONCLUSIONS

In this papers we propose a H^2IL co-simulation to have a realistic simulation of the a human-in-the-loop cooperative robotic platform and to try to deal with most of the physical phenomena which we will find ourselves out in the development of a real haptic cooperative robotic platform. H^2IL co-simulation was sectioned in two main blocks:ADAMS and MATLAB, softwares used to perform the co-simulation. In ADAMS block, the cooperative robots was modeled using CAE. Block MATLAB was used to control the CAE-based model and to interface it with the human using Hardware-in-the-loop Falcon device.

The H^2IL co-simulation proposed in this paper was useful to determine essential elements of a real platform for collaborative robots operated by human through haptic interfaces. This elements are robots's end effector and its characteristics like shape, material and dimensions; the resolution needed in the haptic interfaces, sensors and actuators, and force sensors; and the last but not the least, the controller to be implemented in the platform.

To have realistic simulation of a system it is necessary to include most of the phenomena acting on it, however there are phenomena that can not easily modeled, in fact including most of the phenomena result in a complex simulation which can not be easily interpreted and tuned, in the case of controlling the system. For this reason co-simulation become in a useful tool simulating complex systems including those with human interaction, as in this paper, using Hardware-in-the-loop techniques.

In this paper we considered the robots at the same distance of the object and using the same controller we assure that they contact the object at the same time, as future work we can apply an other controller to have the robots in different initial conditions and distance to the object but arriving to the object at the same time. We can also consider a cooperative system but using more that two robots or a more complex systems like a humanoid (Wei, et al, 2008) or humanoids cooperatively handling an object.

V. ACKNOWLEDGEMENTS

Authors acknowledge CONACYT for the scholarship 266745 and the project 548

REFERENCIAS

Lianqing Yu; Yingying Xue; Shunqi Mei (2008). The Application of ADAMS in the Mechanical System Simulation Course. *2008 International Seminar on Business and Information Management*

Zhen Zhu; Naing, M.P.; Al-Mamun, A. (2009). 3-D Simulator using ADAMS for Design of an Autonomous Gyroscopically Stabilized Single Wheel Robot. *2009 IEEE International Conference on Systems, Man, and Cybernetics*

Zhang Qiang, Jin Xiaoxiong (2008). The study of fuel cell car powertrain mounting systems design using MATLAB and ADAMS. *2008 International Conference on Computer Science and Software Engineering*

Jingjun Zhang, Lihong Shi, Ruizhen Gao and Chaoyang Lian (2009). Method for obtaining direct and inverse pose solutions to Delta parallel robot based on ADAMS. *2009 International Conference on Mechatronics and Automation*

Parra-Vega, V.; Arimoto, S.; Yun-Hui Liu; Hirzinger, G.; Akella, P.; (2003). Dynamic sliding PID control for tracking of robot manipulators: theory and experiments. *IEEE Transactions on Robotics and Automation* **19**, 967–976

Garca-Rodríguez, R. and Parra-Vega, V. (2005). Decentralized Sliding Force/Position PD Control of Cooperative Robots in Operational Space under Jacobian Uncertainty. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*

Yun-Hui Liu; Parra-Vega, V.; Arimoto, S.(1996). Decentralized cooperation control: joint-space approaches for holonomic cooperation. *1996 IEEE International Conference on Robotics and Automation*

Navarro-Alarcon, D.; Parra-Vega, V.; Olguín-Diaz, E. (2008). Minimum set of feedback sensors for high performance decentralized cooperative force control of redundant manipulators. *2008 International Workshop on Robotic and Sensors Environments*

Mitsubishi Electric Coorporation. Industrial Micro-Robot System Manual Model RV-M1 Move Master EX.

Flowmaster (1980-2010). *http://www.flowmaster.com/*.

Andrew S. Elliott, Mechanical Dynamics, Inc. A Highly Efficient, General-Purpose Approach for Co-Simulation with ADAMS

Wei, Hangxin; Wu, Wei; Liu, Mingzhi; , Simulation of the Humanoid Running Robot Based on ADAMS, *Tenth International Conference on Computer Modeling and Simulation, 2008. UKSIM 2008.* , vol., no., pp.726-731, 1-3