Hardware-in-the-Loop Simulation of Robots Performing Contact Tasks

Farhad Aghili Erick Dupuis Jean-Claude Piedbœuf
Jean de Carufel
Space Technologies, Canadian Space Agency,
6767 route de l'Aéroport, St-Hubert, Québec, Canada
tel: (450) 926-4688, fax: (450) 926-4695 email: Jean-Claude.Piedboeuf@space.gc.ca

Abstract

The SPDM Task Verification Facility is being developed to verify all SPDM tasks on the ground prior to their execution in space. The operations requiring a contact between the end effector and the work-site will be verified using a hardware-in-the-loop simulator (HLS). Two algorithms for the control of the HLS are proposed: a position control algorithm inspired from a force reflecting master-slave control architecture and Cartesian linearisation scheme. The performance of each scheme is analysed in terms of its ability to faithfully reproduce the dynamic behaviour of SPDM using the HLS. Conditions are given on the position control scheme to ensure adequate performance and its limitations are shown. The Cartesian feedback linearisation scheme is shown to give good fidelity emulation without the strict restrictions of the first control algorithm. A linearisation error compensator is proposed for the second scheme and is shown to ensure boundedness of the error in the presence of linearisation errors. Experimental results showing the application of the two control schemes to a single link robot are given.

1 Introduction

The International Space Station (ISS) will be assemble in space by collaboration of United State, Canada, Japan, Russia and European countries. The Special Purpose Dextrous Manipulator (SPDM) will play a key role in the assembly and maintenance of ISS which involves execution of numerous tasks. The cost and risk associated with execution of robotic tasks in space require that all procedure be verified on Earth prior to their execution in space. The Canadian Space Agency is currently developing the SPDM Task Verification Facility (STVF) that will be used to verify all SPDM tasks before their occurrence in Space. The SPDM contact tasks will be verified using a hardware-in-loop simulation (HLS) [1, 2].

The main requirement for the HLS is the fidelity of the simulator dynamics with respect to that of SPDM. The pose of the robot end-effector and the contact forces generated throughout the task must accurately represent those of SPDM in free and constrained motion. Two algorithms are proposed for the control of the HLS component of STVF. The first method is based on a position control scheme whereas the second one uses a Cartesian feedback linearisation scheme. The performance and limitations of each algorithm are discussed in detail.

2 Position Control Algorithm

The first control algorithm proposed is inspired from a force-reflecting master-slave control architecture where the master is the dynamic simulation of the space robot. The operator enters velocity commands into the dynamic simulator and the ground robot is controlled to track the simulated end-effector position of the space robot. Contact forces are fed back to the dynamic simulator.

2.1 Linear Model of HLS

To analyse the closed-loop dynamics of the hardware-in-the-loop simulation, a linear model (shown in Figure 1) is developed using input-output transfer function in the Laplace domain. The robots are treated as linear systems with two inputs and a single output. The output of the space robot simulator is the end-effector position and its inputs are a velocity command and a force perturbation. The quantities and variables associated with the space robot simulator and ground robot are depicted by subscripts s and r respectively. By applying the principle of input superposition for linear system, the simulator and robot tip positions, x_s and x_r, can be calculated by

$$X_s = G(s)\dot{X}_{des} + Z_s^{-1}(s)F, \quad (1)$$
$$X_r = T(s)X_s + Z_r^{-1}(s)F. \quad (2)$$

In the above $\dot{X}_{des}$ is the real-time control command to the simulator and F is the force perturbation resulting from the contact between the robot

Proc. Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space, 1-3 June 1999 (ESA SP-440)

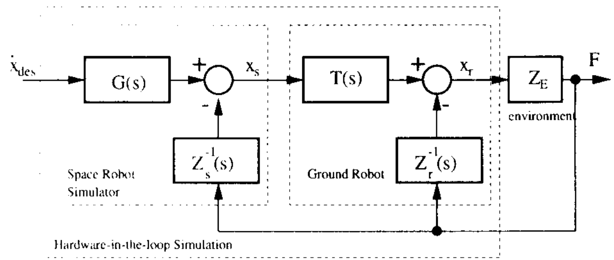

Figure 1: Linear Model of HLS with Contact

and the environment. The closed-loop dynamics of the robot position controller is represented by $T(s)$, the complementary sensitivity function of the ground robot, while $G(s)$ is the dynamics of the space robot simulator in free-space. Moreover, transfer functions $Z_s(s)$ and $Z_r(s)$ represent the impedances of the space robot simulator and of the ground robot respectively. The impedance transfer functions, $Z_s(s)$ and $Z_r(s)$, dictates the dynamical response of the robot endpoint to an external force, F.

By rewriting equations (1) and (2), the linear model of the hardware-in-the-loop simulation can be rewritten in matrix form.

$$\begin{bmatrix} X_s \\ X_r \end{bmatrix} = \begin{bmatrix} G & Z_s^{-1} \\ TG & Z_r^{-1} + TZ_s^{-1} \end{bmatrix} \begin{bmatrix} \dot{X}_{des} \\ F \end{bmatrix} \quad (3)$$

The equivalent impedance of the hardware-in-the-loop simulator is expressed as:

$$Z_{hls} = \left(Z_r^{-1} + TZ_s^{-1}\right)^{-1} \quad (4)$$

Equation (3) characterise the input-output dynamics of the simulator and the robot prototype with the simulator in the loop. This model will be used to address performance and robustness issues of hardware-in-the-loop simulation using a position control algorithm.

2.2 Performance Analysis

In the case where a position control algorithm inspired from the force reflecting master-slave configuration is used, the ground robot is inserted between the space robot simulator and the environment. The ability of the hardware-in-the-loop simulator to reproduce faithfully the dynamic behaviour of the space robot in terms of end-effector position and contact force is limited by the ability of the ground robot to track the simulator both in free-motion and during contact tasks.

2.2.1 Free-Motion

In the free-space, the ability of the ground robot to track the position of the space robot simulator is completely determined by the closed-loop performance of the position control system.

$$X_r(s) = T(s)X_s(s) \quad (5)$$

To ensure perfect tracking of the space robot position by the ground robot, it is sufficient to design the ground robot position controller such that $T(s) \approx 1$ within the bandwidth of interest.

2.2.2 Contact Tasks

Let us define the environment impedance as follows:

$$Z_e = \frac{F}{X} \quad (6)$$

In contact the closed-loop response of the HLS is given by

$$\frac{X_r}{\dot{X}_{des}} = \frac{TG}{1 + \left[TZ_s^{-1} + Z_r^{-1}\right] Z_e} \quad (7)$$

If the space robot were to contact the surface directly, its closed loop response would be

$$\frac{X_s}{\dot{X}_{des}} = \frac{G}{1 + Z_s^{-1} Z_e} \quad (8)$$

Therefore, an extra condition for obtaining the same response in contact is given by

$$Z_r^{-1} = (1 - T)\, Z_s^{-1} \quad (9)$$

This is equivalent to setting the condition that:

$$Z_{hls} = Z_s \quad (10)$$

Designing the force feedback law of the rigid robot to meet this condition is referred to as *impedance matching*. For linear systems, this may seem practical. For nonlinear systems such as robots, however, the design of the law requires the linearisation of the dynamics around the operating point and the online redesign of the control law.

The impedance Z_s^{-1} is a complex transfer function representing the modelled flexible robot. Therefore, the roots of $1 + \left[TZ_s^{-1}\right] Z_e$ for various gains of $Z_e(s)$ and $T(s)$ (changing the stiffness of the environment or/and the bandwidth of the position controller) may be very different than the roots of $1 + Z_s^{-1} Z_e$. In fact, figure 2.2.2 shows an example of the root locus plot for a flexible beam contacting the environment and the one of the HLS counterpart where a rigid beam is used to follow the flexible beam tip motion. In the HLS case, a fixed linear controller resulting in a fixed transmissibility $T(s)$ of the rigid beam is used. The environment stiffness is the varying parameter. The results clearly shows that HLS is conditionally stable while the real system is unconditionally stable. This represents a main limitation for the master-slave approach.

3 Cartesian Feedback Linearisation

In the light of the tracking problem of the position control approach, a new control algorithm is proposed. In this scheme, the dynamics of the ground

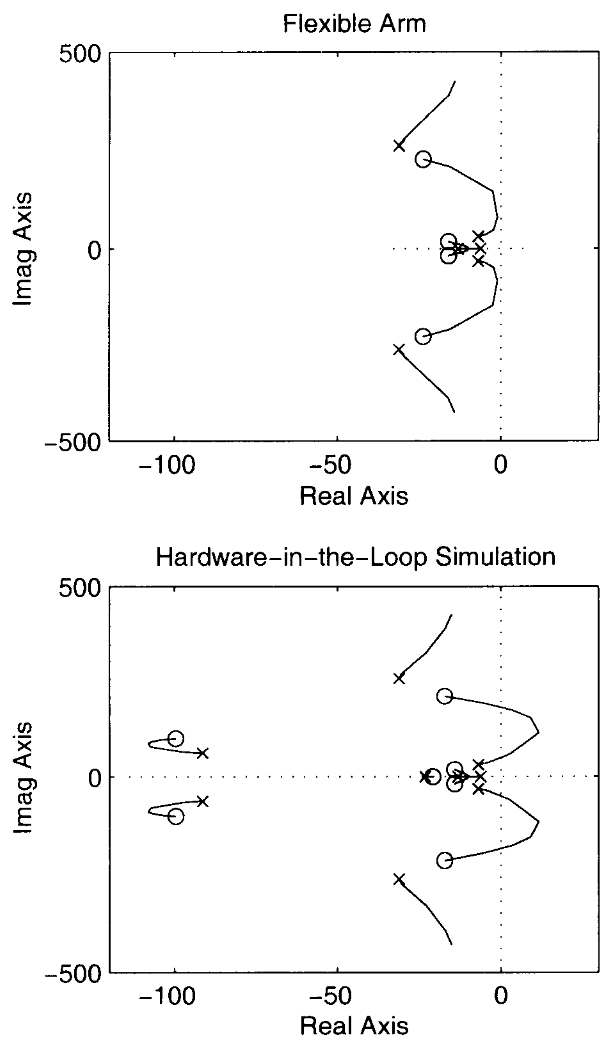

Figure 2: Effect of HLS on Stability for Master-Slave Approach

robot is linearised and the ground robot is controlled in cartesian acceleration [3]. A linear compensator is added to handle errors in the cartesian linearisation of the ground robot dynamics.

3.1 Dynamic Modelling in Cartesian Space

Let an n-DOF manipulator operates in the 6-dimensional Cartesian coordinates. Its kinematics is described by the following equations.

$$\mathbf{x} = \mathbf{\Lambda}(\mathbf{q})$$

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{11}$$

$$\ddot{\mathbf{x}} = \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{J}(\dot{\mathbf{q}})\ddot{\mathbf{q}}$$

Where $\mathbf{x} \in \Re^6$ is vector of tip position/orientation, $\mathbf{q}, \dot{\mathbf{q}} \in \Re^n$ is the generalised joint angle and velocity, $\mathbf{\Lambda}(.)$ is forward kinematics and $\mathbf{J}$ denotes the manipulator Jacobian. From (11) the joint accelerations can be expressed as

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\ddot{\mathbf{x}} - \dot{\mathbf{J}}(\mathbf{q})\ddot{\mathbf{q}} \tag{12}$$

The manipulator dynamics can be modeled by,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{u} - \mathbf{J}(\mathbf{q})^T\mathbf{f}, \tag{13}$$

where $\mathbf{M}(\mathbf{q})$ is the manipulator inertia, $\mathbf{h}(\mathbf{q})$ represents the vector on nonlinear terms including Coriolis, centrifugal, gravity, and friction torques, $\mathbf{u}$ is the vector of joint torque and $\mathbf{f}$ is the generalised force perturbation acting on the end effector. Substituting $\ddot{\mathbf{q}}$ into (13) yields

$$\bar{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{x}} + \bar{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{u} - \mathbf{J}^T\mathbf{f}, \tag{14}$$

where

$$\begin{aligned} \bar{\mathbf{M}} &\triangleq \mathbf{M}\mathbf{J}^{-1} \\ \bar{\mathbf{h}} &\triangleq \mathbf{M}\mathbf{J}^{-1}\dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{h} \end{aligned} \tag{15}$$

3.2 Linearisation Control Law

Let us define the following dynamic equations for the ground robot dynamics and the space robot dynamic simulator.

$$\bar{\mathbf{M}}_r(\mathbf{q}_r)\ddot{\mathbf{x}}_r + \bar{\mathbf{h}}_r(\mathbf{q}_r, \dot{\mathbf{q}}_r) = \mathbf{u}_r - \mathbf{J}_r^T(\mathbf{q}_r)\mathbf{f}, \tag{16}$$

$$\bar{\mathbf{M}}_s(\mathbf{q}_s)\ddot{\mathbf{x}}_s + \bar{\mathbf{h}}_s(\mathbf{q}_s, \dot{\mathbf{q}}_s) = \mathbf{u}_s - \mathbf{J}_s^T(\mathbf{q}_s)\mathbf{f}. \tag{17}$$

The following control law is applied to the ground robot,

$$\mathbf{u}_r = \boldsymbol{\alpha}\mathbf{u}_s + \boldsymbol{\beta}, \tag{18}$$

so that the ground robot and the space robot simulator robots have the same dynamic behaviour in Cartesian space, i.e. $\ddot{\mathbf{x}}_s = \ddot{\mathbf{x}}_r = \ddot{\mathbf{x}}$.

Substituting the control law from (18) into equation (16) yields

$$\boldsymbol{\alpha}^{-1}\bar{\mathbf{M}}_r\ddot{\mathbf{x}}_r + \boldsymbol{\alpha}^{-1}\mathbf{h}_r = \mathbf{u}_s + \boldsymbol{\alpha}^{-1}\boldsymbol{\beta} - \boldsymbol{\alpha}^{-1}\mathbf{J}_r^T\mathbf{f} \tag{19}$$

The nonlinear feedback gain, $\boldsymbol{\alpha}(\mathbf{q}_r, \dot{\mathbf{q}}_r)$, and offset, $\boldsymbol{\beta}(\mathbf{q}_r, \dot{\mathbf{q}}_r)$, can be found by equating the terms in equations (17) and (19). Therefore,

$$\boldsymbol{\alpha} = \bar{\mathbf{M}}_r\bar{\mathbf{M}}_s^{-1}, \tag{20}$$

and,

$$\boldsymbol{\beta} = \bar{\mathbf{h}}_r - \bar{\mathbf{M}}_r\bar{\mathbf{M}}_s^{-1}\bar{\mathbf{h}}_s + \left(\mathbf{J}_r^T - \bar{\mathbf{M}}_r\bar{\mathbf{M}}_s^{-1}\mathbf{J}_s^T\right)\mathbf{f} \tag{21}$$

Figure 3 shows the realisation of the control law given by (18). By defining a new control input $\mathbf{u}_r'$, one can partition the given control law in two parts. The first part, depicted by the dashed box entitled Space Robot Simulator in Figure 3, is the calculation of the forward dynamics of the space robot simulator,

$$\mathbf{u}_r' = \ddot{\mathbf{x}}_s = \bar{\mathbf{M}}_s^{-1}\left[\mathbf{u}_s - \bar{\mathbf{h}}_s\right] - \bar{\mathbf{M}}_s^{-1}\mathbf{J}_s^T\mathbf{f}. \tag{22}$$

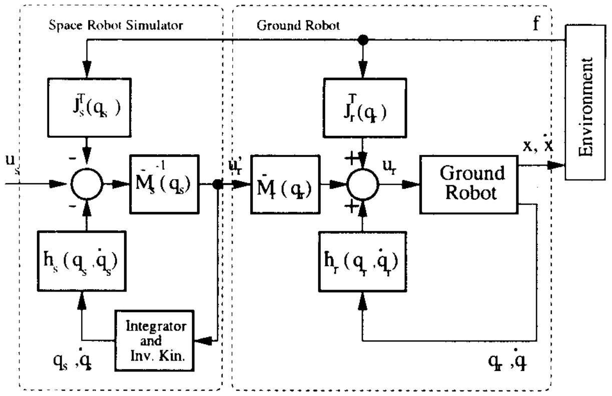

Figure 3: Schematic of the nonlinear control.

The second part of the control law is the linearisation of the ground robot dynamics,

$$\mathbf{u}_r = \bar{\mathbf{M}}_r \mathbf{u}_r' + \bar{\mathbf{h}}_r + \mathbf{J}_r^T \mathbf{f} \tag{23}$$

Combining equations (16), (17), (23) and (22), it is easily shown that the ground robot has exactly the same dynamic behaviour as the space robot simulator in the absence of linearisation errors.

$$\ddot{\mathbf{x}}_r = \ddot{\mathbf{x}}_s \tag{24}$$

3.3 Linearisation Error Compensator

To compensate for errors in the linearisation of the ground robot, a proportional-derivative compensator is added to ensure that the ground robot tracks the motion of the space robot dynamic simulator.

Presuming the presence of errors in the linearisation of the ground robot dynamics, its resulting motion is then described by:

$$\ddot{\mathbf{x}}_r = \mathbf{u}_r' + \mathbf{\Delta} = \ddot{\mathbf{x}}_s + \mathbf{\Delta} \tag{25}$$

To alleviate the linearisation errors represented by $\mathbf{\Delta}$, a compensator is added to the Cartesian feedback linearisation controller. This can be expressed as a modification to the reference signal fed to the Cartesian feedback linearisation controller as follows:

$$\mathbf{u}_r' = \ddot{\mathbf{x}}_s + \mathbf{K}_v(\dot{\mathbf{x}}_s - \dot{\mathbf{x}}_r) + \mathbf{K}_p(\mathbf{x}_s - \mathbf{x}_r) \tag{26}$$

Substituting (26) into (25), the following closed-loop behaviour is obtained for the error.

$$(\ddot{\mathbf{x}}_s - \ddot{\mathbf{x}}_r) + \mathbf{K}_v(\dot{\mathbf{x}}_s - \dot{\mathbf{x}}_r) + \mathbf{K}_p(\mathbf{x}_s - \mathbf{x}_r) = \mathbf{\Delta} \tag{27}$$

This equation shows that for bounded errors in linearisation, it is possible to design a linear compensator such that tracking errors between the space robot simulation and the ground robot are bounded.

4 Experimental Results

Both algorithms presented in this paper were verified experimentally using a one degree of freedom ssytem. The results obtained supports very well the analysis presented.

4.1 Experimental Setup

The experimental setup is schematically represented in Figure 4. A DC motor coupled to an harmonic drive is used to control the position of a rigid beam in the horizontal plane. The position of the motor is measured using a relative encoder located on the motor side and the contact force using a one dof force sensor (FMS) located on the environment side. The joint velocity commands are generated by an operator through a hand controller. The operator commands, as well as the measured contact loads, are supplied directly to a controller driving a simulated flexible manipulator. The controller of the rigib beam receives the simulated tip acceleration, velocity and position, and uses them for defining the desired motion for the rigid beam.

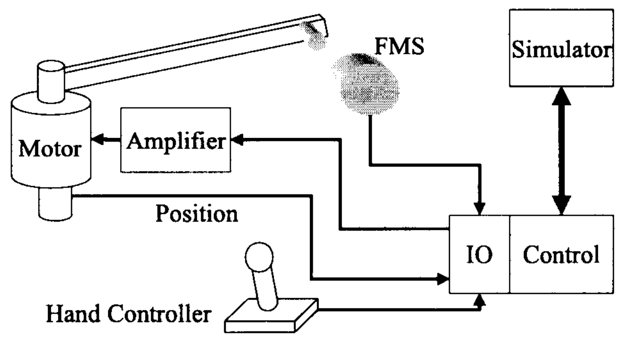

Figure 4: One DOF Prototype Setup

To track the tip motion of the simulated flexible beam, a 3 DOF hardware would be required. Since the rigid beam system has only one DOF, the tracking capabilities are limited. In the experiment presented here, the coordinate θ_{eq} defined in Figure 5 is used as the common coordinate between the simulator and the hardware. It is defined as the angle made by the line passing through the motor axis of rotation and the point on the tip that can make contact with the flat wall.

4.2 Results with Position Control

For the first control algorithm, experimental results were obtained with and without impedance matching.

4.2.1 No Impedance Matching

The results without impedance matching are shown in Figures 6 and 7. During the five seconds of non-contact motion, the tracking error between the hardware and the simulator remains small. In contact,

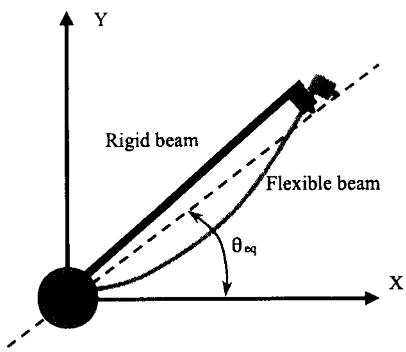

Figure 5: Definition of the tracking angle

however, the tracking error becomes large and constant. This steady-state error, predicted using equation 7 and 8, results from the difference in impedance between the HLS and the simulated system.

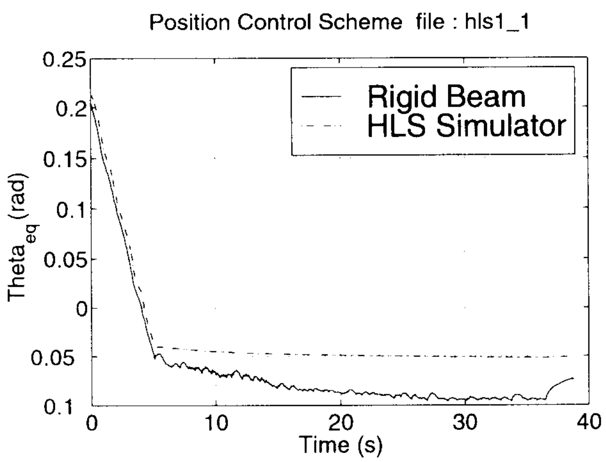

Figure 6: No impedance Matching - position vs time

4.2.2 With Impedance Matching

For the matched impedance case, a force feedback filter was designed and added to the rigid beam control scheme to shape the close-loop impedance and match the flexible beam simulator impedance. This design task requires the complete knowledge of both the master (simulated flexible beam with its control laws) and the slave (rigid beam with its position controller) systems. The results are shown in Figure 8 and 9. Tracking performance in free motion is not affected, but it is improved substantially in contact.

The overall results basically demonstrate that the control algorithm based on the force-reflecting master-slave approach requires impedance matching. Since the matching condition requires a-priori knowledge of the simulated system, the approach is limited in applications.

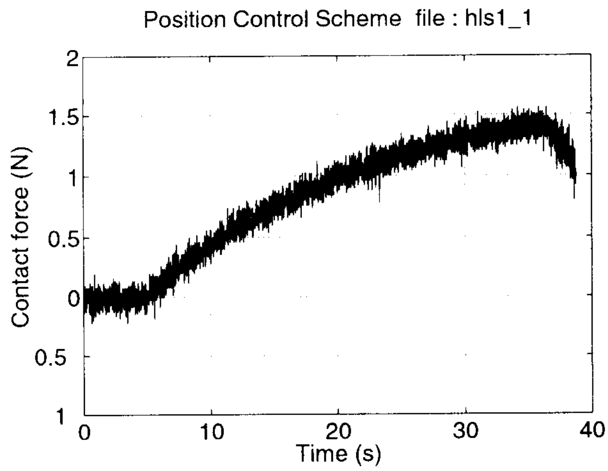

Figure 7: No impedance Matching - contact force vs time

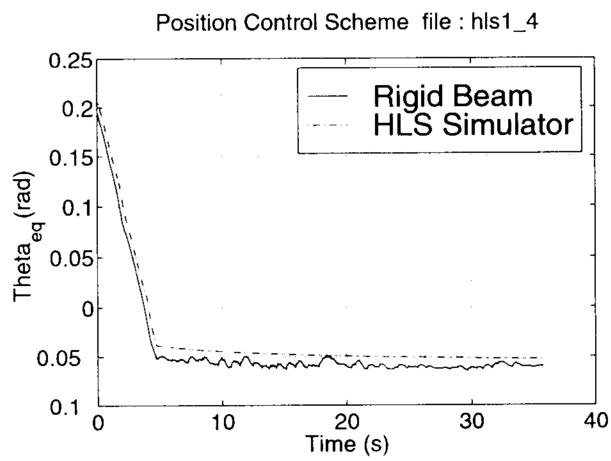

Figure 8: With impedance Matching - position vs time

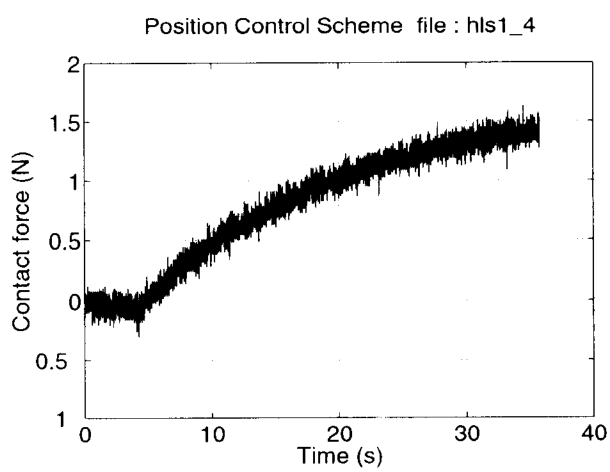

Figure 9: With impedance Matching - contact force vs time

4.3 Feedback Linearisation

The second approach was also demonstrated using the same experimental setup. In addition, pure simulation results were obtained replacing the hardware by a simulation model of the contact dynamics. The results are shown in Figure 10 and 11. The tracking results are good and the control scheme for the rigid beam does not imply any a-priori knowledge of the flexible beam simulator. The experimental contact force also compares very well with the simulated one. The error in tracking is primarely the result of the errors in compensating friction in the motor/drive system. The same observation explains the difference between the contact times in the hardware and in the simulation. These preliminary results suggest that the second scheme is more appropriate for controlling the hardware for a hardware-in-the-loop simulator, and that the performance is basically linked to the knowledge of the hardware to be linearised.

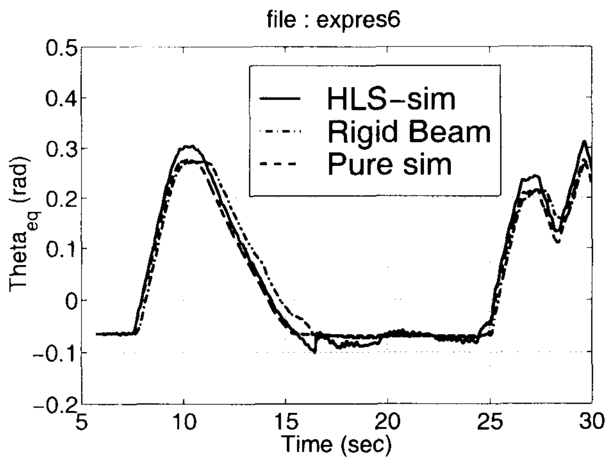

Figure 10: Cartesian Feedback Linearisation - position vs time

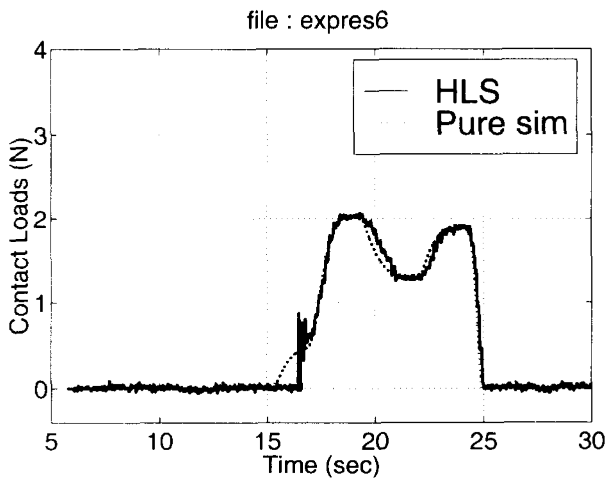

Figure 11: Cartesian Feedback Linearisation - contact force vs time

5 Conclusion

Hardware-in-the-loop simulation includes a wide variety of applications. In the case of the SPDM Task Verification Facility, a non-representative hardware is used to emulate the behaviour of a simulated system. The control system design problem generated by this application is conceptually very similar to the typical master-slave control problem. A main difference, however, lies in the objectives of the design. While for the master-slave system, the objective is to obtain stable close-loop response meeting a given performance criteria, the objective for the hardware-in-the-loop simulation is the complete transparency of the hardware. This paper has demonstrated that in the standard force-reflecting master-slave control, the complete transparency is achieved by matching the impedances between the HLS and the simulated system. This impedance matching condition is realised by properly shaping the force feedback loop. The concept was demonstrated successfully through analysis and experiments.

To alleviate the problems associated with shaping force feedback loop, this paper contributed another approach to obtain complete transparency. Using feedback linearisation, the dynamics of the hardware is linearised and decoupled in Cartesian space, providing complete transparency for cartesian motion assuming perfect linearisation. This approach has the advantage that transparency is achieved independently from the simulated system. It was also shown that for bounded linearisation errors, the tracking error is bounded as well. Experiments results desmonstrated the potential of the method.

References

[1] CSA. Special purpose dextrous manipulator (spdm) task verification facility (stvf) concept document. Technical Report CSA-SS-CD-0016, Canadian Space Agency, 1998.

[2] E. Dupuis, M. Doyon, F. Aghili, and J. Piedboeuf. Control system prototyping of the stvf manipulator test-bed. In *1st IFAC workshop on Space Robotics*, Montreal, Canada, October 1998.

[3] N. Hogan. Stable execution of contact tasks using impedance control. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1047 – 1054, Raleigh, N.C., 1987.