Int. J. Engng Ed. Vol. 22, No. 4, pp. 815–828, 2006
Printed in Great Britain.

0949-149X/91 $3.00+0.00
© 2006 TEMPUS Publications.

Hardware in the Loop Robot Simulators for On-site and Remote Education in Robotics*

HAKAN TEMELTAS, METIN GOKASAN
Istanbul Technical University, Electrical and Electronics Eng. Faculty, Maslak, 80626, Istanbul, Turkey. E-mail: temeltas @elk.itu.edu.tr

SETA BOGOSYAN
Dept. of Electrical and Comp. Eng., University of Alaska Fairbanks, AK 99775, USA. E-mail: ffsob@uaf.edu

In this study, a new approach in Hardware-In-the-Loop Simulation (HILS) is introduced as an education tool in robotics, mechatronics and control. The paper discusses the development and utilization of HILS specifically in the instruction of control and design aspects of on-site and remote robotics courses. A HIL simulator differs from computer simulation as it involves actual hardware and is not limited with the software representation of the system. The HIL architecture proposed in this study is a novel contribution to robotics education and is different from previous HIL structures for robotics, in that the developed test-bed involves the actual joint actuator and can be programmed to reflect actual dynamics affecting that particular joint in the given robotic structure. Two motors driven by high performance DSP boards are used for this purpose; one representing the joint actuator and the other used for the generation of all the torque components affecting that joint for the robot system in consideration. Thus, it is possible to evaluate the overall performance of the robot and its end-effector by combining the data from each simulated joint-associated dynamics pair. The wide range of kinematic configurations thus offered by the simple structure of HILS makes them an attractive, cost-effective solution for the problem of limited repertoire in robotics labs, giving the students the opportunity to experiment with 'open architecture robots' that are essential for a thorough education in robot control, but on the other hand are quite rare and may not be available in an average lab environment. Additionally, similar to its use in the robotics lab applications presented in this study, HILS can provide a comprehensive test environment prior to the implementation of new control strategies on the actual system, thereby ensuring the safe use and longevity of high-cost robots in on-site/remote labs. The proposed HIL architecture is combined with a client/server software configuration to allow students access to the test-bed. The results obtained from the remote control of a two-degrees-of-freedom planar arm presented as a case study have motivated the utilization of the system as an efficient education and research tool for on-site and remote robotics laboratories.

INTRODUCTION

LIKE MOST FIELDS in engineering, hands-on education in control, mechatronics and robotics requires the development of laboratories that provide a variety of experiments, flexibility and ease-of-use. However, high investment and maintenance costs as well as safety issues related to those labs pose important limitations and call for serious consideration to be given to the choice of equipment and design of experiments. Resorting to off-site facilities is another way to address the above limitations, which is an approach gaining increasing attraction as an enhancement or alternative to conventional education tools.

Remote labs are a recent and rapidly growing outcome of Information Technology, providing environments to which users are given access from anywhere in the world using the Internet in order to perform experiments, watch the performance and/or collect back data for analysis. In this paper, a novel hardware-in-the-loop (HIL) simulator setup is proposed as an efficient laboratory tool in the education of robotics, mechatronics, and control. The utilization of this novel approach as an on-site and remote experimentation tool is also discussed in detail for robotics education.

Remote robotics laboratories have gained an increasing popularity in the last decade with a variety of applications ranging from circuits and microelectronics to dynamical systems and control. The existing remote robotics labs offer two types of access:

- *'Virtual' access.* In this approach, the necessary software defining a variety of robotic structures is offered to the students at the virtual lab, who are thus able to perform simulations using the software. An example of a remote lab in this category was offered in [1], based on a software package that combines Internet-based programming tools to generate three-dimensional virtual

* Accepted 16 May 2005.

models of robot manipulators from a Denavit-Hartenberger (DH) parameter table. The package is designed to help students visualize robotic structures and evaluate the effects of the variation of certain robot parameters. Another study in this category is presented in [2], in which software is developed to configure an intranet-oriented robot system by using a database engine.

- *'Physical' access.* There are a number of studies presented on internet based remote lab applications involved with control engineering, such as [3–6] discussing requirements, problems and some solutions for real-time laboratory experimentation over the Internet. Several studies specifically addressing remote robotic labs with 'physical' access, can be listed as follows: [7] presents a paper in which the underlying philosophy and different installations are given for a remote robotic lab; [8] describes the use of a remotely controllable four-axis robot by using LabVIEW software and web technology; [9] presents a project enabling students to work with a 5 degree-of-freedom robotic manipulator from a remote site. Among more recent studies, [10] describes the hardware and software environment for an on-line system performing very basic control functions with toys used as simple robotic devices and finally in [11], a virtual robotics lab is presented which combines the teleoperation of an actual arm with simulations performed simultaneously.

Laboratories in the first category, however useful they may be as an education tool and as a first step in the design and control of robots are limited with virtual use and most of the time will not reflect the actual dynamical behavior of the system due to the often neglected consideration of unmodeled dynamics, parameter uncertainties, and measurement errors etc. In this regard, labs in the second category giving access to actual experimentation are ultimately more effective for the goals of a more hands-on education and research. However, due to the employment of actual robots and equipment, labs in this category -whether on-site or remote - are expensive to set-up, maintain and provide with long hours of daily staffing. Safety and longevity issues of these costly instruments arise as other important points of consideration, particularly when open access is allowed to users of a multitude of education, background and experience levels. The variety of the kinematic structures offered to the student may also be a concern for robotics labs, calling for an increased number of robots in a variety of configurations. On the other hand, even with a good selection of robots in the lab, the low level of control capability offered to the students on robotic structures in most on-site/remote robotics labs poses an additional limitation. Most robot manipulators in the market come with an in-built control structure that generally allows users to enter controller design parameters and trajectories to be tracked by the robot end-effector or joints. However, for the utilization of a robot manipulator as an education tool in robot control or control in general, 'open architecture robots' should be used for a higher level of involvement with the control aspects—namely, by allowing the student to control the torque input for each joint actuator of the robot, with his/her own control algorithms. However, such open-architecture robotic structures are quite rare and costly, bringing along additional concerns of student and equipment safety, liability, and maintenance.

The major contribution of this study is the development of a remotely controllable hardware-in-the-loop (HIL) simulation test-bed, which is proposed as a novel addition to on-site and remote robotics labs to solve most of the above mentioned limitations. The paper gives a detailed description for the preparation and use of the HIL simulator for a pilot on-site graduate robotics course. The developed remote capability was tested in this course with several student experiments conducted from another lab environment in the same building with the use of a client computer and Internet. The steps involved in the preparation of the HILS and its remote use are given in detail for a student experiment on a two-degree-of-freedom (DOF) planar robotic configuration. However, those steps can be extended to the preparation and experimentation process on any robotics configuration.

Hardware-in-the-loop simulation (HILS) differs from computer simulation as it involves actual hardware and is not limited with a software-based representation of the system. The set-up incorporates some of the crucial hardware that takes part in the actual system, thus enabling a more realistic evaluation basis for the system in real time, thanks to the high speed microelectronics and computer technologies of our day. Due to the value it offers both in manufacturing and education, HIL systems have found a wide range of applications in aircraft and missile industries [12, 13], automotive industry [14, 15], motion control, mechatronics [16–20] and finally in the field of robotics; however, most HIL related work is, limited to the use of the DSP-based controller as the actual hardware in the generation of the classical computer simulation results. Reported applications of the HIL concept in robotics is given in [21, 22], in which the HIL approach is used with the incorporation of the actual motion controller into the simulation process of a robot arm. However, the contribution has been limited to the use of one piece of actual hardware, which is the DSP-based controller used in the generation of the computer simulation software.

The HIL design proposed in this study is new for both on-site and remote robotics labs and differs from the previous work done for robotics, in that the developed test stand (Fig. 1) incorporates not only the actual motion controller, but also the joint actuator which can be tested in real-time under various dynamics effects, desired trajectories and control methods. The set-up consists of two motors coupled through their shaft for the representation and real-time simulation of the performance of any one joint in a given robot configuration. One of the motors simulates the joint actuator, while the other motor is used for the generation of the dynamics acting upon that particular joint considering the given configuration. The two motors acting as 'actuator' and 'load torque simulator' are driven separately via high performance controller boards; the resulting set-up thus offers an 'open architecture' control structure, by allowing the user to intervene into the actual torque generation process with his/her control algorithm to test the response of each robot joint in turn, for a given joint reference trajectory and under the real dynamic effects applied to the joint actuator through the shaft.

The HIL simulation set-up can be configured in accordance with a given D-H table. Thus more 'customized' user requirements can be fulfilled without the need of the actual robot. Hence, in addition to offering kinematic variety and open-architecture set-ups, HIL simulation will also address some of the safety and security issues by providing a test-bed prior to the experimentation on the actual robot. In this study, a client/server communication strategy is also developed for the use of HIL for remote education. Finally, as a case study, the real time simulation and related animation of a two degree of freedom planar robot arm is performed on the HIL setup, which is controlled remotely with the proposed communication protocol. The results obtained are very motivating for the utilization of the proposed HIL robot simulators in on and off-site robotic labs.

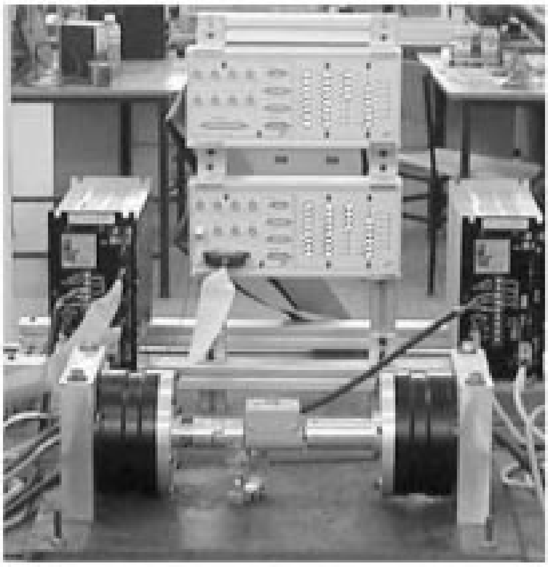


Fig. 1. The HILS test stand.

HARDWARE IN THE LOOP ROBOT SIMULATOR

The proposed HIL system in this study aims to simulate the real-time performance of a user-developed control method on a robot system of the user's choice. The simulation process is carried out for one joint at a time and unlike computer simulations, involves the physical generation of the torque components affecting that joint, as well as the physical control of the joint actuator for a certain reference trajectory while it is exposed to those actual effects.

There are two major aspects in the construction of a HIL robot simulator:

1. Design and implementation of the HIL hardware.
2. Design of the HIL software.

Design and implementation of the HIL hardware

Two motors coupled to one another through their shafts constitute the core of the proposed system given as a schematic diagram in Fig. 2 and as a functional block diagram in Fig. 3. This configuration allows the real-time simulation of any one joint in a robotic system, with one of the motors presenting the joint actuator for the robot in consideration, while the other motor is utilized as the load simulator, for the actual generation of the torque, which will affect the particular joint for the given robot system. The process thus repeated for each joint actuator-load pair throughout the whole robotic configuration, provides a very realistic tool for the understanding and evaluation of certain dynamic effects as well as the assessment of the developed control methods on the robot performance.

The HIL system could be used for the real-time simulation of geared and direct-drive robot (DDR) systems, merely by the appropriate choice of the actuator motor to match the type of the joint motors used in the actual robot to be simulated. The motor simulating the load torque on the other hand, is chosen as a direct drive (DD) motor, for the fast and accurate generation of the internal and external dynamics affecting a certain joint. Examples of such dynamics are the coupling effect between links, giving rise to Coriolis and centripetal terms, gravitational torque, friction, link flexibilities, etc. The encoders on each motor provide position information. The two motors are driven by high performance controller boards with a compact structure for control as well as the required speed and floating point capability for demanding applications of real-time simulation and control. While one high performance DSP board could be sufficient for the control and simulation, to accommodate sophisticated and computationally complex control algorithms as well as the calculation of dynamics related to the high number of DOFs, the proposed set-up uses a motion controller board for each motor.

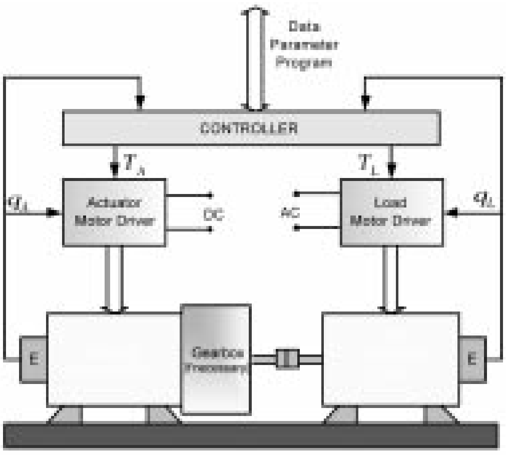


Fig. 2. Schematic diagram of the HIL simulator.

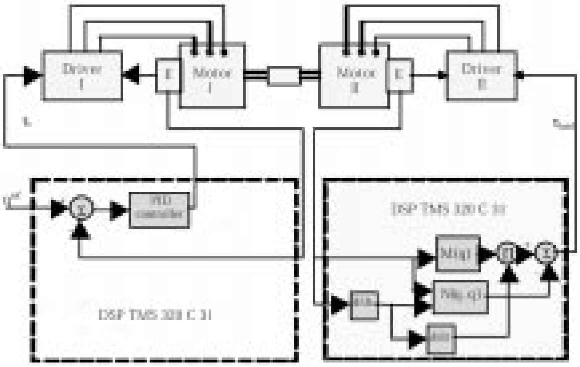


Fig. 3. Functional block diagram of the hardware structure for the HIL simulator.

The motion control unit is crucial for the realistic performance aims of the system. For the lab application presented in this study, the MATLAB/ Simulink compatibility of the controller board has been important for the choice made due to advantages gained like functionality, flexibility and ease-of-use, while also fulfilling the high performance demands of real-time simulation. For these purposes, in the proposed HIL structure, dSpace 1102 boards containing a floating point DSP-TMS320C31, a fixed point DSP-TMS320E14, analog and digital I/O, 12-bit, 4-channel ADCs and DACs, incremental encoder inputs and PWM signal outputs, are used to run the load simulator and control algorithms. These cards also have RS-232 units for communication purposes. The DSP card (DSP Board I) connected to the motor driver: Board I generates the control inputs to the joint motor for given reference trajectories, while the second card, DSP Board II, connected to the load torque simulator motor, produces the load torque corresponding to the dynamics affecting the simulated joint. Both DSP cards use the position feedback from the incremental encoders installed on the motors. The position signal for the second DSP card is required for computing the dynamic effects on the simulated joint. These equations also require joint velocity and acceleration signals, which are obtained through the first and second derivates of the position signal. To avoid the additional noise created by differentiation, special filters should be developed performing time derivation under low noise. Proper scaling of the load torque generated on the simulator is another important issue that needs to be taken into consideration in the case of an actuator motor with a different power rating than that of the actual joint motor.

Design of the HIL software

Providing access to on-site and remote students for experimentation on the HIL system requires the proper design of the software structure. The present laboratory environment, which is described in the following sections allow both user groups to experiment with robotic structures and control methods developed on the server. As mentioned previously, in the proposed structure the software for the developed algorithms and robot models is designed to run on MATLAB/ Simulink downloadable high-performance DSP boards placed in the slots of the server computer.

MATLAB/Simulink-based libraries are prepared on the server to include a variety of dynamic and kinematic models of robot systems presented in the RobotLib given in Fig. 4 as well as those prepared for linear and nonlinear control methods presented in the ControlLib given in Fig. 5. Input/output functions—such as encoder, DAC, ADC, RS232, PWM, timer and interrupt functions—provided by the controller board manufacturers in DSP LIB, are also essential for user experiments.

To provide guidance and instruction for all student types, MATLAB/Simulink-based templates are prepared for the development of control software and robot dynamics to be downloaded onto

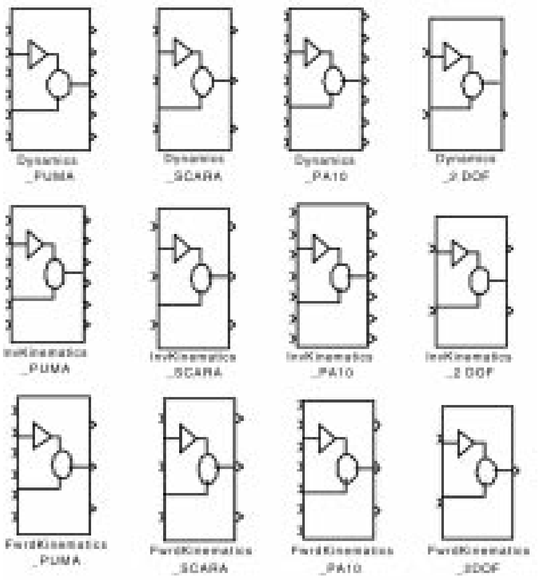


Fig. 4. RobotLib: Library for robot dynamics and kinematics models for selected robot configurations.

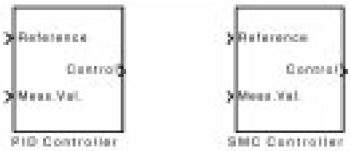


Fig. 5. ControlLib: Library for several controllers

DSP Boards I and II. Each block in the templates holds a default 'Simulink mdl' type file which can easily be replaced by the user with another 'Simulink mdl' type file of his/her choice, selected from the provided libraries.

In the current laboratory environment, on-site students have direct access to all developed libraries and templates, while remote students make use of libraries indirectly through the client software. Once again, the control and dynamic templates are offered to remote users for guidance and instruction only, through 'ReadMe.pdf' files downloaded with the client software.

In the preparation of the HIL set-up for on-site and remote use, the following steps are involved in development:

1. Development of algorithms and software for the control of robotic structures.
2. Development of models and software for robot dynamics and kinematics.
3. Development of client/server software to establish communication with remote users.

Development of algorithms and software for the control of robotic structures

The template given in Fig. 6 gives a description of the software that is downloaded onto DSP Board I driving the actuator motor.

The template allows the entry of the joint number to be simulated, related reference trajectory, position feedback from the joint encoder and the choice of controller for trajectory tracking. The reference block in Fig. 7 provides reference trajectories for each joint actuator in the robot system under consideration. The trajectories can be selected from the Simulink 'Source Lib' for each joint as given in Fig. 7. Students are also provided with the capability of entering reference trajectories for the end effector of a given robot in Cartesian space, which will be converted to joint space via the corresponding robot inverse kinematics models from the Robot Lib, also provided in the server as can be seen in Fig. 4.

The developed software on DSP Board I, allows the student to specify the joint number he/she desires to simulate, by switching the corresponding reference trajectory to the controller block as input. The controller block can run any one of the linear/nonlinear control algorithms developed in the control library to address the multilevel needs and requirements of the users. The developed software allows the user to select an algorithm from a group of control methods developed in the control library. Currently, methods such as PID and sliding mode control (SMC) suitable for independent joint control, are presented as independent control methods of which the user can change the parameters. Model-based methods such as computed torque control, adaptive linearizing control, etc., are developed and presented individually for each of the robot systems presented to on-site and remote students on the server in the RobotLib. The controller block also accepts the encoder output from Motor I as feedback, while outputting the calculated torque value for Motor I to the DAC. Figure 8 presents the controller developed for the PID controller, which is the default controller provided in the control block of Fig. 6.

The 'Enc' and 'DAC' blocks in the template are imported from the DSP library provided by dSpace. Blocks providing RS-232, ADC, DI/O, PWM timers and interrupt signals to the boards are also provided.

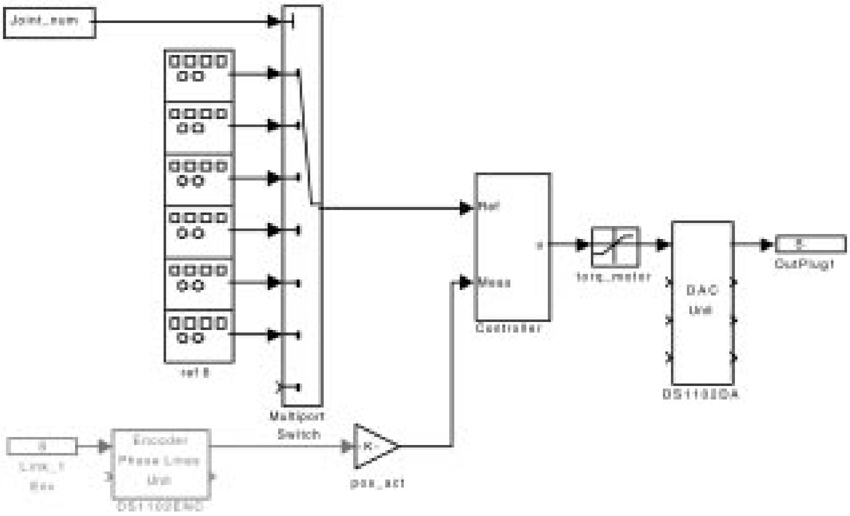


Fig. 6. Template for joint control.

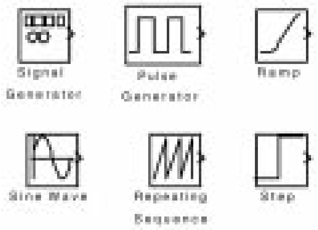


Fig. 7. Library for reference trajectories.

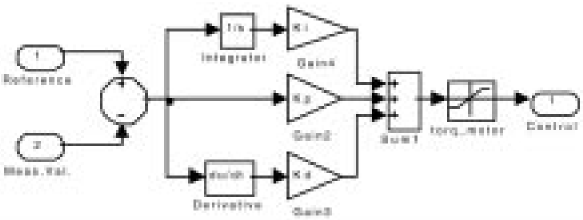


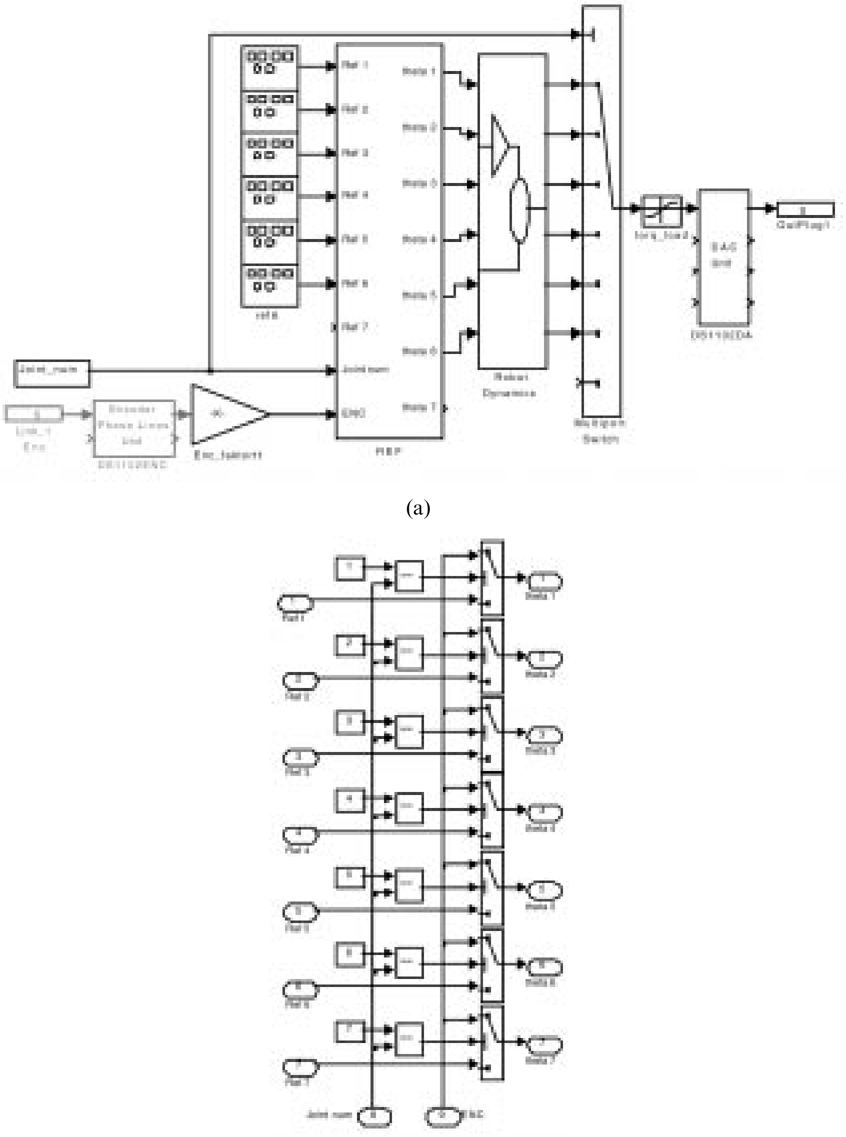
Fig. 8. Controller developed for PID controller.

Development of software for robot dynamics and kinematics modeling

The real-time simulation of the dynamics affecting the joint in consideration is performed by the second motor, which is programmed to generate a load torque equal to the amount of the torque supposed to be affecting the joint in consideration for the given robot system. For this purpose, the software related to the corresponding dynamics is downloaded onto the DSP Board II, driving the second motor. For the efficient development of the .mdl files for the robot dynamics, both user groups are provided with a template as given in Fig. 9.

The reference block in this template matches the reference block in the controller template for each application; while the angular position of the simulated joint, i, is obtained from the encoder, the angular positions of the other joints required for the calculation of the dynamics are taken equal to their reference values.

Using the joint angular position obtained from



(a)

(b)

Fig. 9. (a) Template for the simulation of robot dynamics; (b) detailed demonstration of REF block.

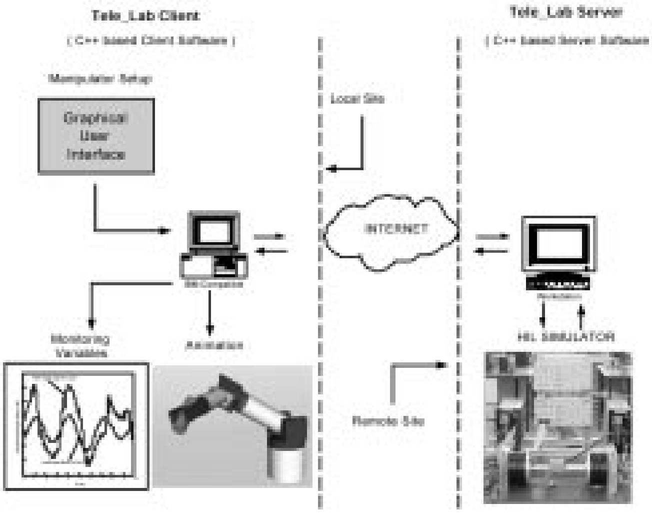


Fig. 10. Remote access for HIL simulator.

the encoder on the second motor for the simulation of the dynamics affecting that joint provides flexibility of use, in that with this approach, besides direct-drive and geared robot configurations, systems with joint flexibilities can also be accommodated and analyzed with HIL simulations.

Finally, dynamic models for a variety of robot configurations are developed using MATLAB/Simulink for the robot dynamics block, the default for which is chosen as the 2-DOF planar arm. The models in this block are provided to the student in the RobotLib. Figure 4 presents a library of robot configurations in which forward/inverse kinematics and dynamics models are provided for the proposed application.

Development of client/server communication software for remote users

Remote users are given access to the HIL system via the client/server communication principle presented in Fig. 10. The server, also called the Tele-Lab Server, is located in the remote lab and executes all tasks in order to operate the HIL simulator for the given robot configuration. The server software, which is developed under C++, has three basic functions:

- Session management, which creates communication between a client and the server by listening to the specified port.
- Receiving and setting up the user selected parameters on the HIL simulator and DSP hardware; passing back output data for user analysis.
- Resource sharing management to organize clients connecting to the server simultaneously by queuing their experiment demands and informing the client about the status before and during the user's experiments.

The graphical user interface (GUI) for the server program is demonstrated in Fig. 11a. The client software, also called TeleLab. Client, is designed to run on minimal hardware required for communication with the remote lab via the Internet. The main functions of this utility are to manage and monitor the communication between the server and client. The successful management of the communication provides the remote user with the capabilities to select the robot structure, reference

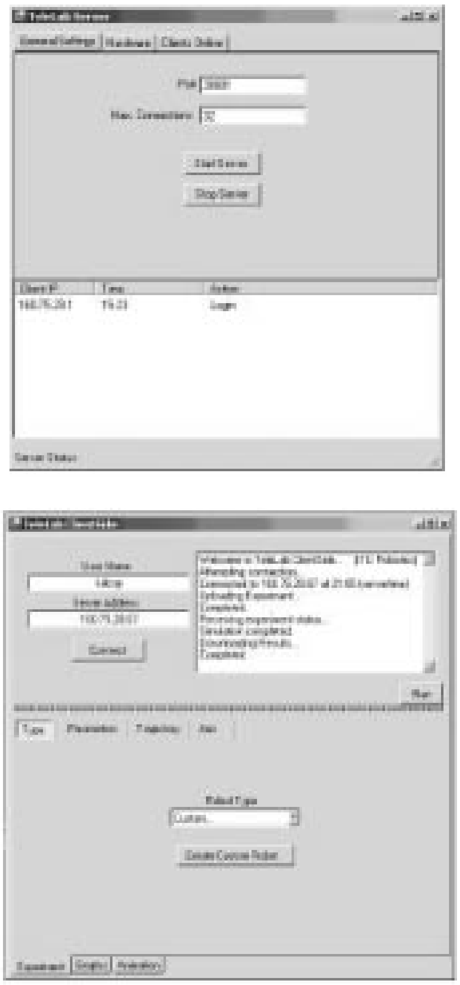


Fig. 11. (a) Server GUI; (b) client GUI.

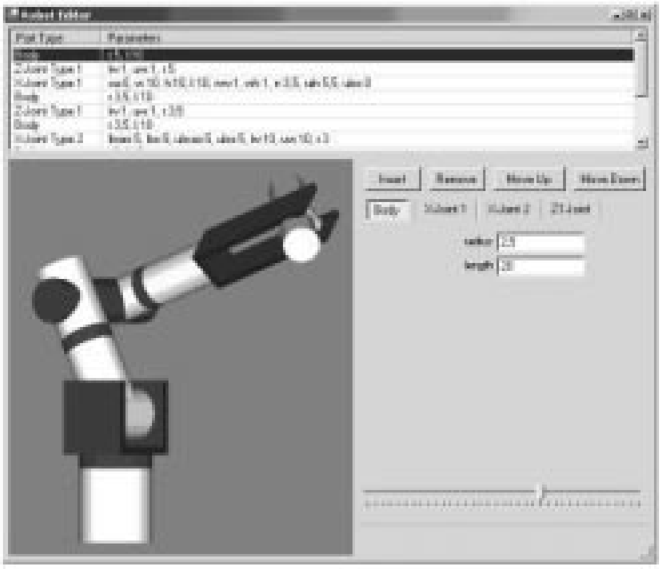


Fig. 12. Robot editor.

trajectory in Cartesian or joint space and a control method, on each of which, he/she can enter or change parameters. The required software is downloaded from the laboratory website. The illustration of client side GUI is given in Fig. 11b.

EXPERIMENTATION PROCESS FOR ON-SITE AND REMOTE ROBOTIC EDUCATION

As mentioned previously, the HIL robot simulator lab facility developed for both on-site and remote users curently offers more versatile experiment options to on-site users due to their direct access capability. As a matter of fact, on-site users can exploit the full capacity that the original experiment developers exercise: They can simply use the dynamics and kinematics models of commercial robot systems, such as PUMA 560, SCARA, PA-10, 2–3 DOF planar robot arms etc. provided in the RobotLib on the server, or they can develop custom robots or create robotic systems of their choice by assembling the structures provided for use in the 'Robot Editor'. The latter is a WYSIWYG-GUI, given in Fig. 12, for constructing custom robots from predefined primitives, such as bodies and joints. Similarly, on-site students are given full capability to develop and implement control algorithms of their choice, hence exploiting the 'open architecture control' capacity of the HIL simulator. However, to ensure a safe operation, they are required to perform first a computer simulation of the developed control method on the robot system in consideration, using the templates provided on the server. Only after satisfactory results, can they download the developed .mdl files onto DSP Board I and II, by performing the necessary modifications.

A second option for students is to perform experiments using the robot systems and control algorithms provided on the server with the developed libraries. For this purpose, the student refers to each block presented in the templates and in order to experiment with models or algorithms other than the defaults given in the blocks, he/she copies the model or algorithm from the corresponding library onto the given block. Experimentation at this level involves selecting the appropriate blocks and changing parameters inside these blocks, which can easily be performed via the GUIs provided both on the client and server side.

The existing environment offers the remote users only the second experimentation option via the GUI on the client side. After entering the lab website with the proper *user id* and *password*, the remote user downloads the client software, which gives the user access to the GUI to express his/her experiment demands, namely the robot type, reference trajectories, joint number, control method and related parameters as well as the ReadMe file.

In addition to plots provided for the performance analysis of each joint and the end effector, the user may also download the animation software on the server which is provided for both user types. This option provides users the capability to watch repeatedly the performance the actual robot would be demonstrating with the reference trajectory and the control algorithm in consideration.

IMPLEMANTATION PROCESS OF HILS FOR A ROBOTIC COURSE

The HIL robot simulator described in the previous sections is used for a lab assignments in a pilot on-site graduate robotics class. The students are first introduced to the HILS concepts as well as the related libraries and templates prepared prior to the course. Although not a requirement for HIL applications, due to the specific approach taken in this study, the students are required to have a working knowledge of MATLAB/Simulink.

The following is an outline of how the HIL set-up is integrated into the robotics modeling and control syllabus as an education tool:

- As each concept in the syllabus is covered, such as forward/inverse kinematics, forward/inverse dynamics and linear/nonlinear methods commonly used for robot control, the student is required to do manual and MATLAB/Simulink exercises related to the subject.
- The instruction in the areas of dynamics, modeling and control are also combined with HIL-based exercises. For this purpose, currently two robotics configurations, SCARA and 2-DOF planar arm, are taken into consideration, for which libraries have already been prepared on the server computer in the lab prior to the course start date.
- After each class on dynamics modeling and control of robots, the students are assigned to perform experiments on the HIL set-up considering all three robotic configurations using the existing software library and templates prepared for HIL experimentation. In those weekly assignments, the students are also required to compare the MATLAB/Simulink-based computer simulation results with those obtained through HIL simulation, commenting on the discrepancies between simulation and experiment results.
- After the completion of experiments on all three configurations, the students are assigned to implement their own software on the HIL set-up for the dynamics and control of a robotics configuration of their choice. The students are required to comply with the format used for the existing HIL experiments. This approach is taken to facilitate the addition of student-developed new robotics configurations and control methods onto the existing repertoire on the server computer.
- Finally, as a pilot lab experiment on the HIL set-up with remote access, the students are assigned to conduct one experiment each remotely using the client computer in another lab.

CASE STUDY: HIL SIMULATION OF A 2-DOF PLANAR ARM

As an example for both on-site and remote application of the HIL simulator in the above mentioned robotic course, the real-time simulation of a 2-DOF planar arm is taken into consideration. To provide a practical example for all aspects of the HIL simulation, each step of experimentation, from the development of robot modeling and control prior to the course, to the actual use of the experiment by the remote users, is covered in the following section.

Development of dynamical model for the 2-DOF planar arm

As a guide for the development of the libraries for robot dynamics as given in Fig. 4, the dynamic equations for a 2-DOF planar robot arm are derived, using Euler-Lagrange formulation as follows Fig. 13 refers.

$$\tau_m = M(q)\ddot{q} + C(q,\dot{q}) + G(q) + F\dot{q} \qquad (1)$$

where:

$M(q)$: Mass and inertia matrix;
l_1, l_2: length of link 1 and 2, respectively;
m_1, m_2: mass of link 1 and 2, respectively;
I_1, I_2: inertia of link 1 and 2, respectively;
I_{m1}, l_{m2}: inertia of motor 1 and 2, respectively;
m_{m1}, m_{m2}, m_p: mass of motor 1, 2 and en-effector, respectively;
$C(q,\dot{q})$: Coriolis and centrifugal vector;
$G(q)$: Gravity vector;
F: Friction matrix;
$q = [q_1\ q_2]^T$: generalized link coordinates.

The matrix expressions in (1) can be given explicitly, also including the motor inertia and friction effects as below:

$$M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{12} & M_{22} \end{bmatrix}$$

$$\begin{aligned} M_{11} &= m_1 l_1^2/4 + m_2(l_1^2 + l_2^2/4 + l_1 l_2 c_2) + m_{m2} l_1^2 \\ &\quad + I_1 + I_2 + I_{m1} + I_{m2} + m_p(l_1^2 + l_2^2 \\ &\quad + 2 l_1 l_2 c_2) \\ M_{12} &= m_2(l_2^2/4 + l_2 l_1 c_2/2) + I_2 + I_{m2} \\ &\quad + m_p(l_2^2 + l_1 l_2 c_2) M_{22} = m_2 l_2^2/4 + I_2 \\ &\quad + I_{m2} + m_p l_2^2 \end{aligned}$$

$$C(q,\dot{q}) = \begin{bmatrix} -(m_2 + 2m_p) l_1 l_2 s_2 \dot{q}_2 (\dot{q}_1 + \dot{q}_2/2 \\ (m_2 + 2m_p) l_1 l_2 s_2 \dot{q}_1 (\dot{q}_1/2) \end{bmatrix}$$

$$F = \begin{bmatrix} B_{ml} & 0 \\ 0 & B_{m2} \end{bmatrix}$$

$$G(q) = \left[\frac{\left(\frac{m_1 l_1}{2} + m_2 l_1 + m_{m2} l_1 + m_p l_1\right) + \left(\frac{m_2 l_2}{2} + m_p l_2\right) g c_{12}}{(m_2 l_2/2 + m_p l_2) g c_{12}} \right]$$

where, $c_1 = \cos q_1 \;\; c_2 = \cos q_2 \;\; c_{12} = \cos(q_1 + q_2)$;

$$s_1 = \sin q_1 \;\; s_2 = \sin q_2 \;\; s_{12} \sin(q_1 + q_2);$$

$$\tau_{y1} = \tau_{m1} - I_{m1}\ddot{q}_1 - B_{m1}\dot{q}_1 - I_{ms}\ddot{q}_1 - B_{ms}\dot{q}_1;$$

$$\tau_{y2} = \tau_{m2} - I_{m2}\ddot{q}_2 - B_{m2}\dot{q}_2 - I_{ms}\ddot{q}_2 - B_{ms}\dot{q}_2; \qquad (2)$$

Here, I_{ms} and B_{ms} are inertia and viscous friction coefficients of the motor used for load torque simulation, respectively;
τ_{y1} and τ_{y2} are the load torque incurring at joint 1 and 2, respectively.

The next step is the transfer of load torques, τ_{y1}

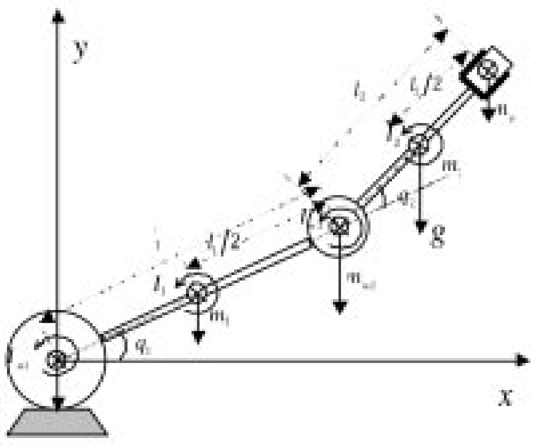


Fig. 13. Two-link planar robotic arm.

and τ_{y2} to the template prepared for the dynamic configuration prior to downloading onto DSP Board II for load simulation.

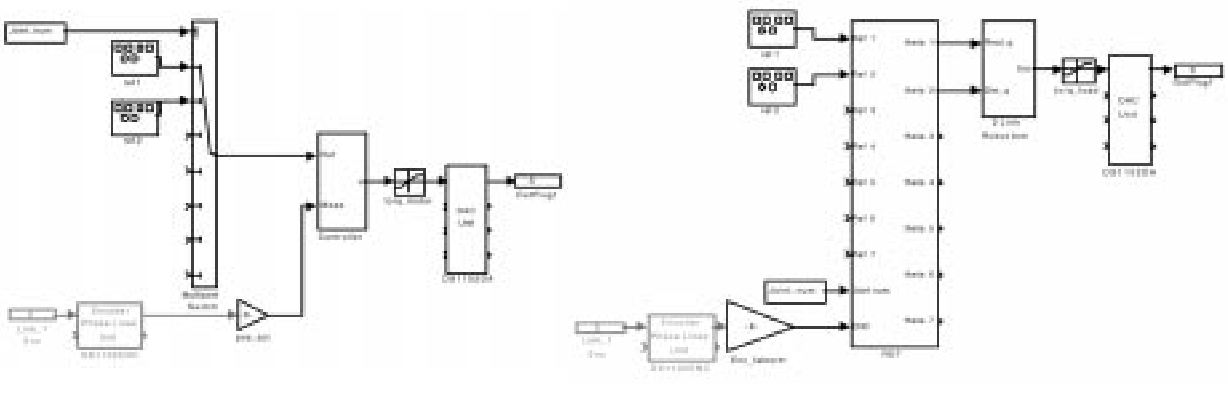
Development of the control methods

As a guide for the development of libraries for the robot control algorithms as given in Fig. 5, PID control will be taken as an example and its integration with the rest of the control template will be presented in this section. Figure 14(a) demonstrates the MATLAB/Simulink-based template for the control of the 2-DOF robot arm prepared for downloading on DSP Board I. As can be seen in Figure 14(b), the joint reference trajectories match the trajectories used in the template of robot dynamics. These references must be switched appropriately to the inputs of PID control for the simulated joint. Proper switching is also essential to perform the appropriate coupling between the joint in consideration and its corresponding load torque. As mentioned previously, for the load simulation of joint *i*, the .mdl file for which is given in Fig. 15(a), the angular position, q_i is obtained from the encoder on Motor II, while the angular positions of the other joints—joint 2 in this case—are considered to be equal to the reference trajectories for these joints. The velocity and acceleration of joint 1 is obtained with the use of special filters performing time derivation with low noise, to avoid the undesirable noise created by differentiation as demonstrated in Fig. 15(b).

Remote experimentation steps and results for the HILS of the 2-DOF arm

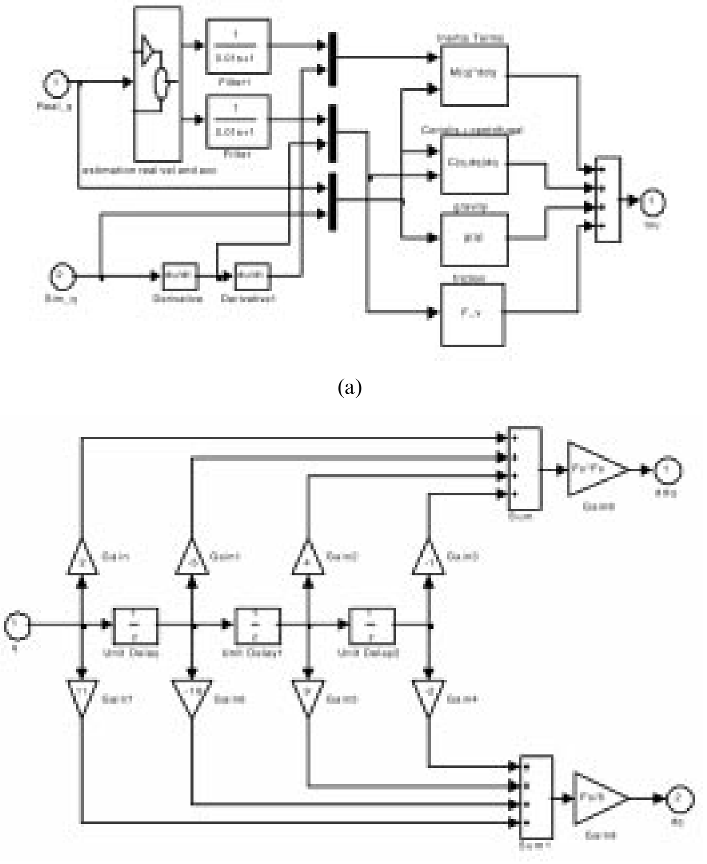
As a guide for the remote use of the HIL simulator, the steps involved for the remote experimentation assignment in the robotics course are discussed in this section. The procedure which is given for the remote implementation of the dynamics and PID-based control of a 2-DOF planar arm involves the following steps:

1. First, the client software is downloaded and the client GUI is reached.
2. From the list of robot configurations, the 2-DOF arm is selected and necessary parameters are entered into the template, such as link lengths, inertia, link masses, etc., for joint 1 first for the specific geometry in consideration. The 'ReadMe.pdf' file could also be reached at this stage, where the template for robot dynamics could be found to provide guidance on the model and corresponding parameters of the selected robot. Thus, the user could have a good understanding of the software being downloaded on DSP Board II, to simulate the load torque for joint 1.
3. Once again, using the control template in the 'ReadMe.pdf' file for guidance, the necessary selections are made for the control of joint 1 and its reference trajectory.
4. With PID control chosen for joint control, K_P (proportional fain), K_I (integral gain), and K_D (derivative gain) parameters are entered via the GUI.
5. The reference trajectory waveforms for both joints are selected as sine waves, for which the desired amplitude and frequency values are entered. Obviously, this selection is made for one joint at a time.
6. Upon the completion of data entry for joint 1, the data package is sent to the server computer



(a) (b)

Fig. 14. (a) Template for the 2 DOF arm controller; (b) Template for the dynamics of 2 DOF planar arm.



(a)

(b)

Fig. 15. (a) Simulink file for 2 DOF planar arm dynamics; (b) Detailed representation of velocity and accelerator.

by clicking the ‘Send’ button on the client GUI. This command puts the user’s request in a queue.

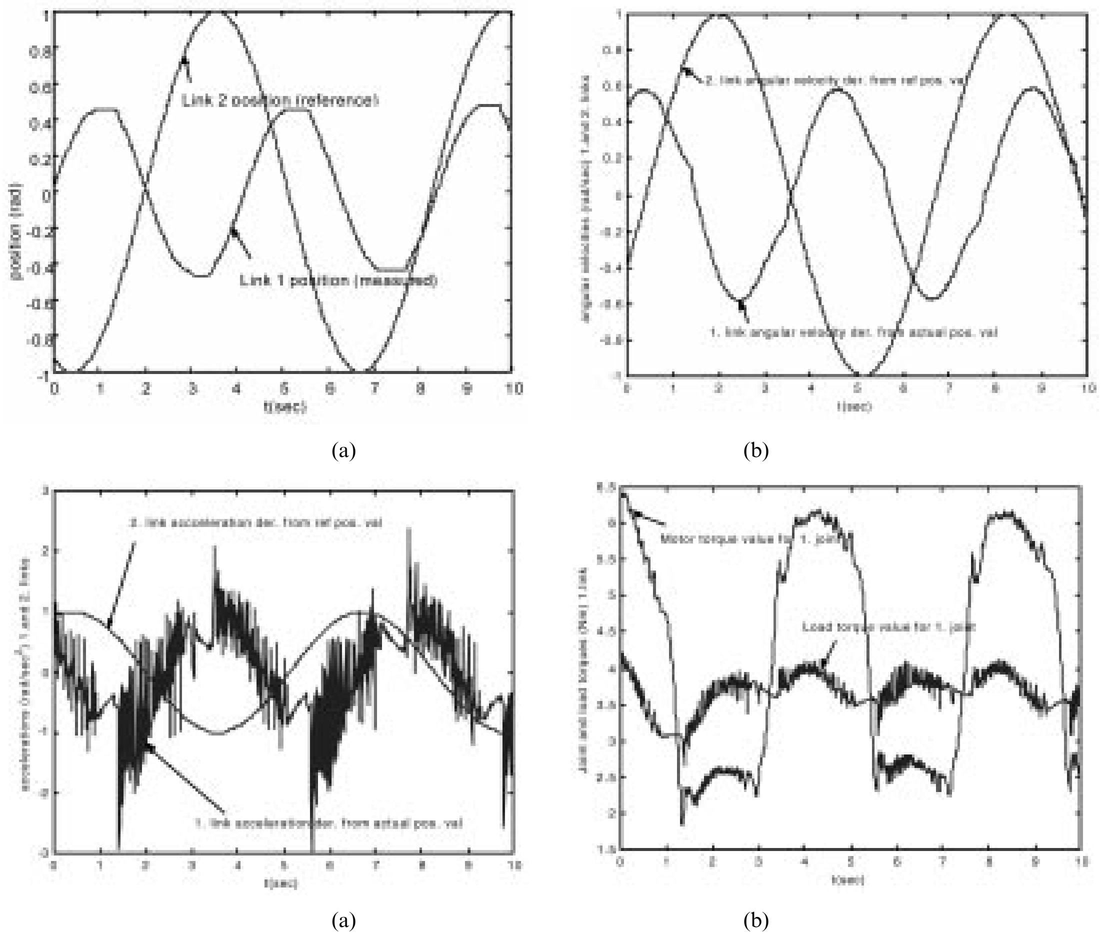
7. Upon receipt of the notification when the experiment is completed, the user downloads the ‘Result’ files which he/she can use to plot the data related to joint 1.
8. The user repeats the same procedure for joint 2 and collects back the data for that joint.
9. Finally, he/she downloads the animation software to watch the overall performance of the 2-DOF arm.

The following results are obtained from the above described remote experimentation process: Fig. 16(a) depicts the variation of q_1 and q_2, while Fig. 16(b) and (c) demonstrate the variations of the joint velocities $\dot{q}, \dot{q}_2$ and accelerations $\ddot{q}_1, \ddot{q}_2$ respectively, obtained by time derivation. The histories of the resulting torque developed on joint actuator 1 and its load dynamics generated on Motor II are also given in Fig. 16(d). The differences between these two torque values cause the motion of the joint. Finally, Fig. 17 shows a still frame of the animation performed with the results obtained for each link, with the choice of the animation option.

CONCLUSIONS AND FUTURE DIRECTIONS

This aim of this paper is to present the HIL robot simulator test-bed, which is a novel and efficient education tool for use in on-site and remote robotic laboratories. The proposed HIL set-up is obtained by incorporating not merely the actual motion controller unit as in most previous HIL applications for robotics, but also the joint actuator and the real-time load torque that is physically effective at each link. Thus, the HIL robot simulator fulfils the following educational objectives for both on-site and remote education in robotics:

- Due to the involved theory in kinematics, dynamics and control of robots, a thorough education in robotics requires lab sessions. However, there are some limitations involved with most robotics equipment in teaching the control aspects of the system, i.e. most robotic structures allow the students very little involvement in the actual development of the control strategy, limiting the student’s access to merely varying design/control parameters on already established control methods. Robotic systems



(a) (b)

(a) (b)

Fig. 16. (a) HIL simulation for link 1: position; (b) HIL simulation for link 1: velocity; (c) HIL simulation for link 1: acceleration; (d) HIL simulation for link 1: torque.

allowing a higher level of involvement, on the other hand, are rare and high priced and hence, not quite affordable for an average robotics lab. The HIL robot simulator presented in the study provides a cost-effective solution to the requirement of 'open-architecture control robots' by allowing the student the opportunity to not only change control parameters, but also design the whole control strategy, starting at the joint actuator level.

- HIL robot simulators also provide a cost-effective solution to limited repertoire in both on-site and remote labs, as the student can configure a system of his/her choice on the HIL set-up and experiment with it without the need for the actual existence of that robotic system in the lab. This gives the instructor the capability to demonstrate inertial, gravitational and load effects that will vary from configuration to configuration and would be difficult to illustrate with limited repertoire. This is an important aspect in the education of robot dynamics and control and calls for experimental support to instruction.
- The HIL set-up also provides the student with a better understanding of the arm and actuator dynamics, in comparison to mere experimentation with the actual robot. The dynamics behind the motion is no more a 'mystery', but a group of mathematical terms that he/she can add or subtract from the load simulator to identify the effects of each physical phenomenon.

In addition to the above educational objectives, another benefit of the proposed HIL robot simulator is in terms of safety and longevity concerns related to robotics equipment. This issue becomes particularly important with remote robotics labs, with the free access granted to students of a multitude of background levels to experiment on costly robotics equipment. To address this problem, the HIL set-up could effectively be used as a test-bed by the remote student to evaluate his/her newly developed strategies prior to experimentation on the actual robot. Additionally, it could provide a means for the student's initial understanding of the expected dynamics behavior of the actual robotics arm. In this respect, the ideal utilization approach for both remote and on-site labs would be providing a combination of HIL simulators and actual robots in the lab to fulfill the above mentioned objectives; that is, by

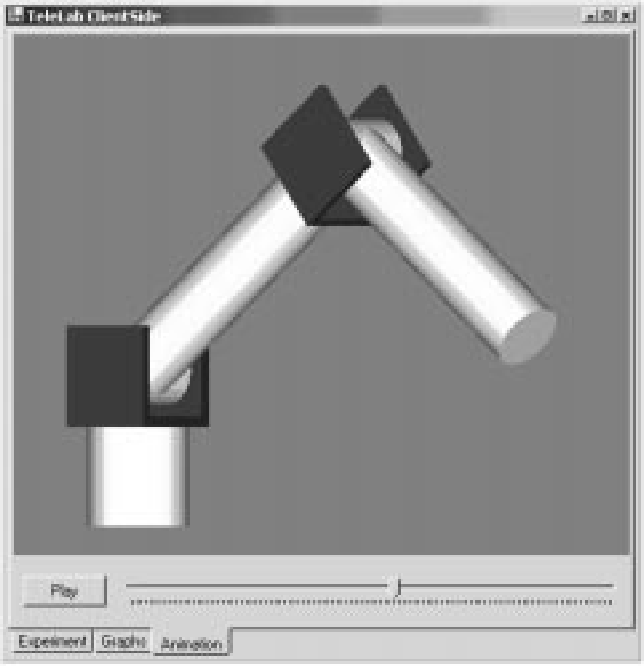


Fig. 17. Animation of the two-link planar robotic arm.

meeting the repertoire shortages of robotic labs, while also ensuring the safety and longevity of existing robotic systems.

Finally, the obvious lack of appeal of a HIL setup when compared to actual robot systems can also be partially met with the inclusion of visually engaging and realistic animation programs included in each experiment assignment. This approach has proved to be very effective in the pilot course described in this paper.

The outcomes of the pilot course have been very valuable in our evaluation of the HIL simulator for on-site and remote robotics. Besides creating increased enthusiasm and synergy in comparison to the course conducted previously with no lab sessions, the HIL simulator has helped maintain the continued interest of several graduate students, who are currently using the HIL approach to obtain the experimental data for their research in robot control. The results of their research will be used to further enhance the existing HIL experiments repertoire, while increasing the level of access for remote experimentation.

Acknowledgements—This work was supported in part by the Scientific and Technical Research Council of Turkey (TUBITAK) under grant number EEEAG201, an NSF-CISE Grant number 0423739, and partially by NWACC (Northwest Academic Computing Consortium).

REFERENCES

1. M. F. Robinette and R. Mansuer, ROBOT-DRAW, an Internet-based visualization tool for robotics education, *IEEE Trans. Education,* **44**, 2001, pp. 29–34.
2. H. Imada, Y. Araki and N. Aoki, Development of web robot using the full-text retrieval software JTOPIC, *NEC Research & Development,* **39**, 1998, pp. 55–60.
3. C. H. Salsman, H. A. Latchman, D. Gillet and O. D. Crisalle, Requirements for real-time laboratory experimentation over the Internet, *Int. Conf. Engineering Education, ICEE'98,* 1998, Rio de Janeiro, Brazil, August.
4. J. Overstreet and A. Tzes, An Internet-based real-time control engineering laboratory, *IEEE Control System Mag.,* October, 1999, pp. 19–34.
5. C. A. Bohus, L.A. Crowl, B. Aktan and M. H. Shor, Development Internet-based control experiment, *Proc. on 40th IEEE Conf., Decision and Control,* **3**, Dec. 4–7, 2001, pp. 2809–2814.
6. N. Swamy, O. Kuljaca and F. L. Lewis, Internet-based educational control systems lab using Netmeeting, *IEEE Trans. Education*, **45**, 2002, pp. 145–151.
7. W. E. Dixon, B. T. Costic and M. S. de Queiroz, A MATLAB-based control systems laboratory experience for undergraduate students: toward standardization and shared resources, *IEEE Trans. Education,* **45**(3), 2002, pp. 218–226.
8. E. Cooney and A. Shiver, Remote control of a robot using LabVIEW and the World Wide Web, *Proc. 2001 American Society for Engineering Education Annual Conf.,* American Society for Engineering Education..
9. A. Bicchi, A. Copelli, F. Quatro, L. Rizzo, F. Turchi and A. Balestrino, Breaking the lab's walls: tele-laboratories at the University of Pisa**,** *Proc. IEEE Int. Conf. Robotics and Automation,* Seoul, Korea, May 21–26, 2001, Vol. 2, pp. 1903–1908.

10. G. T. McKee, An online robot system for projects in robot intelligence, *Int. J. Eng. Educ.,* **19**, 2003, pp. 356–362.
11. F. A. Candelas, S. T. Puente, F. Torres, F. G. Ortiz, P. Gil and J. Porames, A virtual laboratory for teaching robotics, *Int. J. Eng. Educ.,* **19**, 2003, pp. 363–370.
12. P. E. Pace, M. D. Nash, D. P. Zulaica, A. A. Di Matessa and A. Hoshmer, Relative targeting architectures for captive-carry HIL missile simulator experiments, *IEEE, Trans. Aerospace and Electronic Systems,* **37**, 2001, pp. 810–823.
13. K. Georgiew, Airbus flight control system under test, *dSPACE News,* **4**(2), Dec. 1995. www.dspace.de
14. Y. Xie, V. Vilathgamuwa, K. J. Tseng and N. Nagarajan, Modeling and robust adaptive control of 3-axis motion simulator, *IEEE Industry Application Conf. 2001*, Chicago, IL, USA, Vol. 1, pp. 553–560.
15. H. Hanselmann, Hardware-in-the-loop simulation testing and its integration into a CACSD toolset, *IEEE Int. Sym. Computer Aided Control System Design,* Sep. 7, 1996, USA, pp. 15–18.
16. J. Limroth, A. Rahbar and B. Faribozi, Integration of control designs and simulation models into a modular real-time platform, *SAE 2004 World Congress and Exhibition,* Detroit, MI, USA, Technical paper 2004–01–1239, March, 2004.
17. H. Hanselmann, Hardware-in-the-loop simulation as a standard approach for development, customization and production test, *SAE Technical Paper 930207*, (1993).
18. D. Ramaswamy, W. Stuart, R. McGee, et al., A case study in hardware-in-the-loop testing: development of an Ecu for a hybrid electrical vehicle, *SAE 2004 World Congress and Exhibition,* Detroit, MI, USA, Technical paper 2004–01–0303, March, 2004.
19. R. Isermann, Mechatronic systems: concepts and applications, *Trans. Institute of Measurement and Control,* **22**(1), 2000, pp. 29–55.
20. M. Linjama, T. Virvalo, J. Gustafsson, et al., Hardware-in-the-loop environment for servo system controller design, tuning and testing, *Microproc. and Microsy.,* **24**(1), Mar 29, 2000, pp. 13–21.
21. R. Siegward, C. Wannaz, P. Garcia and R. Blank, Guiding mobile robots through the web, *Proc. Int. Conf. Intelligent Robots and Systems (IROS): Workshop on Web Robots,* 1998.
22. R. Simmons Xavier, An autonomous mobile robot on the web, *Proc. Int. Conf. Intelligent Robots and Systems (IROS): Workshop on Web Robots,* 1998.

Hakan Temeltas received the M.Sc. degree in Control and Computer Engineering from Istanbul Technical University in 1987. He joined The University of Nottingham, UK in 1988 and finished his Ph.D. in robotics 1993 at the same University. Currently he is an Associate Professor in Istanbul Technical University. His major work has been in the field of sensor data fusion for navigation and control of mobile robots in several projects. His research interests, in general are planning, modeling, control and design of robotics and mechatronics systems. He is also interested by systems and set-ups required in education of robotics and mechatronics systems.

Seta Bogosyan received her B.Sc., M.Sc. and Ph.D. in electrical and control engineering from Istanbul Technical University in 1981, 1983, and 1992, respectively. Between 1987 and 1991, she conducted her PhD studies at the University of California Santa Barbara where she worked as a visiting researcher and lecturer. In 1993, she joined Istanbul Technical University as an associate professor in electrical engineering. Currently, she is with the University of Alaska Fairbanks where she is working as an associate professor in electrical engineering. Her research interests are in nonlinear control, motion control, precise control of direct drive and underactuated robotic systems and control of electrical motors. She has authored several publications as well as a book on the control of induction motors.

Metin Gokasan received his B.Sc., M.Sc., and Ph.D. degrees in Electrical and Electronics Engineering from Istanbul Technical University in 1980, 1982, and 1989, respectively. He is with the Istanbul Technical University, Faculty of Electrical and Electronics Engineering as an associate professor since 1990. His research areas are control systems, mechatronics, power electronics and electrical drives. Presently, he is at the University of Alaska Fairbanks, where he is working as a visiting professor on sensor fusion and hybrid electrical vehicles. Dr. Gokasan is the author of several publications and a book, entitled 'Asynchronous Machines and their Control' (in Turkish).