

Design and Implementation of a Virtual Laboratory for Machine Dynamics

[doi:10.3991/ijoe.v6i2.1184](https://doi.org/10.3991/ijoe.v6i2.1184)

El-Sayed Aziz, Sven K. Esche and Constantin Chassapis
Stevens Institute of Technology, Hoboken, NJ, USA

Abstract—Laboratory experiments are considered to be a crucial component of engineering and science curricula by all stakeholders in the education process. In traditional laboratories, students develop practical skills and become effective professionals. However, the major drawbacks of traditional laboratories are their high demand on resources, significant maintenance costs and the inability to deliver the laboratory content in distance education. Virtual laboratory experiments represent a valuable option for educational laboratories, due to their advantages over traditional hands-on as well as remote experiments, including the ease of reconfiguring the experimental system, the high flexibility in the input specifications, the possibility of conducting experiments using devices otherwise infeasible and the option of re-running experiments multiple times.

In response to the need for developing laboratory resources that provide a practical experience to online engineering students, this paper describes a simulation-based virtual laboratory, which is used at Stevens Institute of Technology (SIT) in a junior-level course on mechanisms and machine dynamics as a compliment to experimental work in the traditional hands-on laboratory. This virtual laboratory system alleviates the space, time and cost constraints associated with traditional laboratories and serves as an efficient teaching aid. It conveys to the students practical issues associated with actual experiments and provides learning outcomes that are comparable to those of traditional physical laboratories. The students have the opportunity to explore a wide range of experimental configurations and parameters. In addition, the system includes a realistic rendering of the experimental setup and its components, thus providing the students with a strong feeling of immersion, as if they were performing an experiment in a traditional laboratory.

Index Terms—Virtual Experiment, Virtual Learning Environment; Simulation, Internet, Control, Machine Dynamics

I. INTRODUCTION

Recently, many engineering education institutions have been focusing on innovation in pedagogy and developing new high-quality curricula for traditional and non-traditional students, which also satisfy the ABET 2000 criteria. Some studies [1,2] reported in the education literature indicate that information technologies can be used to enhance learning by offering distance education courses and online laboratories via the Internet, where learners are given flexibility in choosing the place, time and pace of studying. Despite the past and continuing success in the development of online learning offerings, the absence of tools for providing an online laboratory

experience prevents the introduction of online undergraduate programs in certain fields [3]. In undergraduate mechanical engineering programs, for which laboratory sessions are deemed indispensable, students are expected to develop hands-on laboratory skills in order to complete the degree requirements. The acquisition and maintenance costs of laboratory equipment in conjunction with the space required to house it put a great burden on today's undergraduate engineering education institutions. In the traditional hands-on mode of laboratory operation, multiple copies of the same hardware are needed in order to accommodate all students enrolled in a class. This approach often turns out to be prohibitive due to resource limitations. Furthermore, hardware-based laboratory exercises typically offer only limited features, and because of their weight and size, most experimental devices are not suitable to be taken to the classroom. At the same time, student enrollments in engineering schools are increasing while the allocated laboratory resources do not keep up with this increase. In this environment of constrained resources, it thus becomes increasingly difficult for educational institutions to provide an adequate laboratory experiences to their undergraduate students. The research efforts described in this paper represent an attempt to find alternatives or complements to hands-on laboratories for delivering experimental experiences to on-campus as well as off-campus students.

Some of the above-mentioned problems can be solved using the concept of virtual learning environments for conducting experiments without expensive physical laboratory setups [4,5]. A virtual learning environment is a set of teaching and learning tools designed to enhance the students' learning experience by distributing educational content in a digital format (texts, images, audio, simulations, games, etc.) via the Internet. In this concept, the Internet is providing a real-time tool that eliminates space and time constraints and allows the sharing of high-quality educational software applications (online simulations), which have been demonstrated to be popular and intuitive to use [6]. Online simulations have the potential to add significant value to corporate training environments by helping the students to become familiarized with the equipment before conducting traditional hands-on laboratory exercises. Therefore, the students' activities can focus more on understanding the underlying fundamental concepts rather than on performing tedious set-up, calibration and measurement procedures, and they can thus devote a larger portion of their time to discussions of the experimental results and observations. Furthermore, due to their ability to obtain and analyze the measured data quickly and efficiently,

they are able to repeat some of the experiments. Animations, sounds, lighting effects and other aspects of the virtual laboratory world may be triggered by external events such as timers. By using virtual learning tools, the quality of undergraduate laboratories can be improved while at the same time dramatically reducing their costs of operation. Therefore, laboratory courses can be taught more efficiently and less expensively through the use of Web-based technologies.

Simulation-based virtual laboratories represent a valuable option for implementing educational laboratories due to their advantages over conventional hands-on as well as remote experiments, including the ease of reconfiguring the experimental system (e.g. change of material, dimensions, etc.), the high flexibility in the input specifications (e.g. input data that would destroy a corresponding experimental apparatus), the possibility of conducting experiments using devices otherwise infeasible (e.g. too expensive or too dangerous) and the option of rerunning experiments multiple times. Students can perform experiments on simulated systems by means of software that is provided by a server through a Web browser without downloading the source code. Recently, many software-based laboratory experiment simulations have been developed at various educational institutions around the world that allow students to define experimental procedures by specifying certain input variables and observing the resulting output [7-11]. While some of these simulations represent stand-alone software applications to be run on the individual students' computers, others reside on a server for sharing by multiple users through a Web browser. For example, a Virtual Engineering/Science Laboratory at Johns Hopkins University makes it possible to simulate engineering and science laboratory projects on a remote computer [12]. This environment mainly focuses on experiments for demonstrating theoretical concepts and runs without veritable experimental instruments. At Purdue University, a virtual laboratory called SoftLab was developed to provide an environment for both physical experiments and numerical simulations [13]. A Web-based interactive virtual system for a unit operations laboratory and a control experiment for separation processes were recently developed to give the chemical engineering students a realistic experience with industrial operations [9]. At Purdue University, a photorealistic three-dimensional (3D) computer-simulated laboratory for undergraduate instruction in microcontroller technology was developed [14], where OpenGL and C/C++ were used to program the virtual microcontroller with components to provide an authentic laboratory experience. At the University of South Australia, NetLab [15] was developed in a very realistic manner by using static images of the actual experiment with interactive features, such as buttons, knobs and a live data display.

The problem with the majority of computer-simulated experiments developed so far is that the representation of the virtual environments and their components is very schematic and two-dimensional and therefore fails to provide an authentic experience. If a software simulation is very detailed, it can provide educationally valuable features not available in hardware-based experiments and thus be used as an effective complement to or substitute for realistic hands-on experimentation. In order to address the shortcomings listed above, more attention has been

paid at SIT to the 3D virtual laboratory with the aim of providing the students with the same level of understanding of the experiments as a real laboratory environment.

With the flexibility of the SIT virtual laboratory that supports multi-users access, it is easy to re-run experiments, thus allowing students to learn from their mistakes. This paper will describe how to deliver a laboratory experience via the Internet. The paper is organized as follows. In the next section, the key issues and problems associated with hands-on laboratories are examined. Subsequently, the paper discusses potential solutions for these problems. Then, a case study of the conversion of one of the hands-on experiments of the mechanical engineering curriculum at SIT into a virtual experiment is presented. Specifically, a simulation-based virtual experiment for an industrial plant emulator system was designed and implemented using the new generation of interactive technologies (e.g. Java/Python, Virtual Reality Modeling Language (VRML), etc.). In this implementation, it was attempted to build the virtual emulator system such that it closely resembles the real system. Finally, several concluding remarks are outlined..

II. HANDS-ON LABORATORY EXPERIENCE

Computers equipped with data-acquisition systems have had the effect of increasing the level of hands-on experimental activity in mechanical engineering laboratories [16]. Computerized experiments tend to change the emphasis of laboratory experiments from routine, often tedious set-up, calibration and data collection activities towards data interpreting skills, critical thinking, creativity and problem solving. In addition, associated data analysis can be carried out either within the same software environment or by exporting the data to a general purpose data processing utility, such as a spreadsheet package with graph-plotting and curve-fitting features. The industrial plant emulator system shown in Figure 1 represents an experimental setup, which was designed to provide students with insights into dynamics, vibration and control system principles through hands-on demonstration and experimentation [17]. It consists of an electromechanical plant, comprised of an emulator mechanism together with actuators and sensors, and a full complement of control hardware and system interface software. This system is used at SIT in the laboratory component of a junior-level mechanical engineering course on mechanisms and machine dynamics. It can be applied to measure and analyze various inertia values, gear ratios, torques and viscous/Coulomb friction as well as to investigate the effects of proportional, integral and derivative control on the performance of a dynamic system, in particular the transient response (overshoot, speed of response) and steady-state error. The electromechanical plant consists of a drive motor (servo actuator), which is coupled via a timing belt to a drive disk. Another timing belt connects the drive disk to a speed reduction assembly, and a third belt completes the drive train to a load disk. Encoders are attached to the shafts for recording the system response. The inertia properties of the load and drive disks can be adjusted by attaching, moving or removing several brass masses of different sizes. The speed reduction ratio can be adjusted by changing the top and bottom pulleys in the speed reduction assembly. The shaft of the speed reduction

assembly is fixed using a clamp screw, which allows the belts to be slackened when exchanging the pulleys. Furthermore, a backlash mechanism is used to control the amount of backlash between the upper and bottom pulleys. Also, this plant can be configured as a torsional single-degree-of-freedom (SDOF) vibration system with forced excitation.



Figure 1. Industrial plant emulator system

The control software covers most of the above functionality in the computer-mediated hands-on laboratory. However, experimentation in a hands-on laboratory imposes temporal and physical constraints for both on-campus and off-campus students. For example, the installation of the emulator system is time-consuming, and when operating it, special attention must be paid to obeying certain safety regulations (e.g. installing the plexiglas safety cover before switching on the power to the control box, not touching any part of the mechanism while it is moving, verifying that the disks and masses are properly secured prior to powering up the control box, etc.) Moreover, every physical system can only be controlled by one student at a time. In order to provide all students opportunities for conducting laboratory exercises in a more flexible fashion, remotely accessible real-time laboratories have emerged, which reduce the equipment cost (i.e. acquisition, maintenance and replacement costs) and the overall operating and personnel costs of the laboratory. For this reason, a way of providing more flexibility for the emulator experiments was sought by taking advantage of the Internet. Compared with traditional hands-on laboratory exercises, remote laboratory experiments carried out via the Internet have various advantages, such as the feasibility of conducting experiments with long duration and sharing of expensive laboratory setups.

III. REAL-TIME REMOTE EMULATOR SYSTEM

The operational link of the students with the emulator system was established through an experiment server as shown in Figure 2. The experiment server hosts the control software, which relies on MATLAB/Simulink for operating the real physical system through the high-speed I/O control board of the setup. The Remotely Anywhere software [18], which was installed on the experiment server, allows the remote connection to the physical system via a secure link over the Internet. After a connection has been established, the students receive a message that the control software has been launched, and they can select a load-setting file to conduct an experimental procedure using the real physical emulator

system. Then, the students can see the response of the system and export the experimental data as a text file for further analysis using an external software package of their choice. The main shortcoming of the real-time emulator system is that the laboratory apparatus has to be manually preconfigured ahead of time for a specific experimental procedure, i.e. its usage by the students does not involve the configuration of the experimental setup such as adding, removing and adjusting the masses and changing the pulleys, belts and thus the gear ratios. Virtual laboratories can solve the majority of the problems associated with traditional and real-time laboratories, and they can provide students with the same level of understanding of the experiments' underlying theoretical concepts as a traditional hands-on laboratory environment [19].

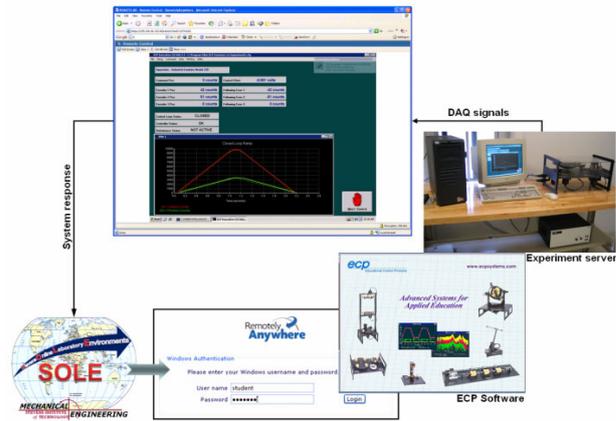


Figure 2. Real-time remote laboratory of emulator system

IV. DYNAMIC MODELS OF PLANT

A. Rigid-body Plant

Detailed analyses of the system responses for the emulator with rigid-body plant and flexible-drive dynamics follow from control theory [17, 20]. The current and subsequent sections describe the corresponding mathematical models of the emulator system that were implemented in the virtual emulator system.

The rigid plant can be modeled as a simple rigid body obeying Newton's second law. The state of the system is defined by the drive shaft angle θ_1 and the load shaft angle θ_2 , respectively, as shown in Figure 3. Modeling the friction effects as being viscous, the equation of motion is:

$$J_D \ddot{\theta}_1 + c_D \dot{\theta}_1 = T_D \quad (1)$$

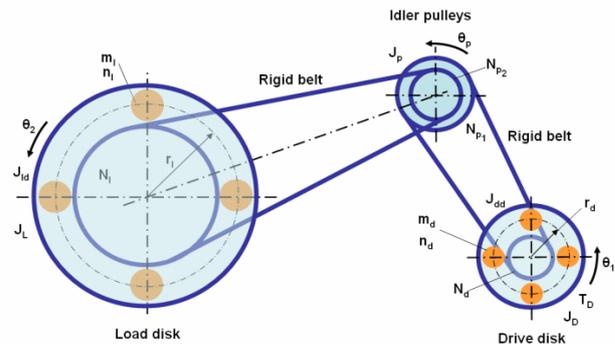


Figure 3. Model of rigid-body plant

Where: T_D is the torque produced by the proportional controller, J_D is the total moment of inertia referred to the drive disk, c_D is the friction coefficient of the drive (modeled as viscous), and $\dot{\theta}_1$ and $\ddot{\theta}_1$ are the angular velocity and angular acceleration of the drive disk, respectively.

Assuming zero initial conditions and taking the Laplace transform of Eq. (1), the response $\theta_1(s)$ to the reference input $T_D(s)$ is:

$$J_D s^2 \theta_1(s) + c_D s \theta_1(s) = T_D(s) \quad (2)$$

Then, the output/input transfer function for the control system between $\theta_1(s)$ and $T_D(s)$ can be written as:

$$\frac{\theta_1(s)}{T_D(s)} = \frac{1}{s(J_D s + c_D)} \quad (3)$$

The inertia J_p associated with the pulleys in the SR assembly (composed of the inertia of the backlash device plus the inertia of the bottom pulley J_{p1} and top pulley J_{p2}) is:

$$J_p = J_{p1} + J_{p2} + J_{backlash} \quad (4)$$

The total inertia J_D of the components of the drive part, referred to the drive shaft, is:

$$J_D = J_{drive} + \frac{J_p}{G_{r1}^2} + \frac{J_{load}}{G_r^2} \quad (5)$$

The drive inertia J_{drive} comprises the inertia of the drive disk assembly J_{dd} (i.e. the drive disk itself plus the drive motor, encoder, drive disk/drive motor belt and pulleys) and any attached weight J_{dw} :

$$J_{drive} = J_{dd} + J_{dw}$$

The moment of inertia of the weight attached to the drive disk J_{dw} is:

$$J_{dw} = n_d m_d r_d^2$$

Where: n_d is the number of masses attached to the drive disk, m_d is the corresponding mass value, and r_d is the distance from the center of the drive disk to the center of the mass. The load inertia J_{load} comprises the inertia of the load disk assembly J_{ld} (i.e., the load disk itself plus the disturbance motor, encoder, load disk/disturbance motor belt and pulleys) and the inertia of any attached weight J_{lw} :

$$J_{load} = J_{ld} + J_{lw}$$

The moment of inertia of the weight attached to the load disk J_{lw} is:

$$J_{lw} = n_l m_l r_l^2$$

Where: n_l is the number of masses attached to the load disk, m_l is the corresponding mass value, and r_l is the distance from the center of the load disk to the center of the mass.

The total moment of inertia J_L referred to the load shaft is:

$$J_L = J_{load} + G_{r2}^2 J_p + G_r^2 J_{drive} \quad (6)$$

In Eqs. (5) and (6), the gear ratios G_{r1} and G_{r2} between the idler pulley assembly and the drive and load disks, respectively, are defined as:

$$G_{r1} = \frac{N_{p1}}{N_d}, \quad G_{r2} = \frac{N_{p2}}{N_l}, \quad G_r = G_{r1} G_{r2}$$

Where: N_D is the number of teeth on the drive disk pulley (fixed number, $N_D = 12$), N_l is the number of teeth on the load disk pulley (fixed number, $N_D = 72$), N_{p1} is the number of teeth on the bottom pulley, N_{p2} is the number of teeth on the top pulley, and G_r is the total gear ratio.

Similarly, the total viscous-friction coefficients, referred to the drive shaft c_D and load shaft c_L , respectively, are:

$$c_D = c_1 + \frac{c_2}{G_r}, \quad c_L = c_2 + c_1 G_r^2$$

Figure 4 shows a block diagram of the overall system, which is comprised of the PID controller and the electromechanical plant.

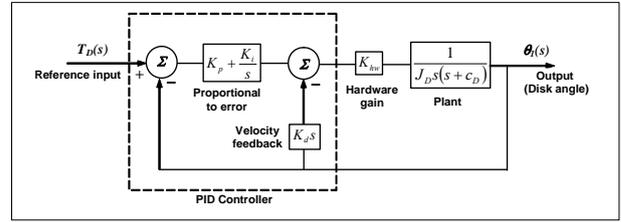


Figure 4. Block diagram for rigid plant PID control

Where: K_p is the proportional gain, K_i is the integration gain ($K_i = 0$), and K_d is the derivative gain.

Therefore, the output/input transfer function for the control system as shown in Figure 5 is:

$$\frac{\theta_1(s)}{T_D(s)} = \frac{\frac{K_p K_{hw}}{J_D}}{s^2 + \left(c_D + \frac{K_d K_{hw}}{J_D} \right) s + \frac{K_p K_{hw}}{J_D}} \quad (7)$$

For a unit-step input:

$$\theta_1(s) = \frac{1}{s} - \frac{\zeta \omega_n}{(s + \zeta \omega_n)^2 + \omega_d^2} - \frac{s + \zeta \omega_n}{(s + \zeta \omega_n)^2 + \omega_d^2} \quad (8)$$

Where:

$$\omega_n = \sqrt{\frac{K_p K_{hw}}{J_D}}, \quad \zeta = \frac{c_D + K_d K_{hw}}{2 \omega_n J_D}, \quad \omega_d = \omega_n \sqrt{1 - \zeta^2}$$

Performing the inverse Laplace transform results in:

$$\theta_1(t) = 1 - e^{-\zeta \omega_n t} \left[\frac{\zeta}{\sqrt{1 - \zeta^2}} \sin(\omega_d t) + \cos(\omega_d t) \right] \quad (9)$$

The hardware gain K_{hw} of the system is defined as:

$$K_{hw} = K_c K_a K_t K_e K_s \quad (10)$$

Where: K_c is the gain of the digital-to-analog converter (DAC) ($K_c = 10 \text{ V} / 32,768 \text{ DAC counts}$), K_a is the servo amplifier gain constant ($K_a = \sim 2 \text{ (A/V)}$), K_t is the servo motor torque constant ($K_t = \sim 0.1 \text{ Nm/A}$), K_e is the encoder gain ($K_e = 16,000 \text{ counts} / 2\pi \text{ rad}$), K_s is the controller software gain ($K_s = 32 \text{ controller input counts} / \text{reference input counts}$).

B. Flexible-drive Dynamics

When using a flexible belt to connect the load disk and idler pulley as shown in Figure 5, the mechanism exhibits flexible drive dynamics. The flexible belt sections are represented by springs with constant stiffness characteristics K_L . The drive system is represented by the total inertia reflected to the drive disk J_d^* , and the load disk is represented by the total inertia reflected to the load disk J_l^* . The configuration of the electromechanical plant is defined by the positions and orientations of the two pulleys and the initial belt length. The equation of motion is:

$$J_d^* \ddot{\theta}_1 + c_1 \dot{\theta}_1 + k(G_r^{-2} \theta_1 - G_r^{-1} \theta_2) = T_D \quad (11)$$

With

$$J_d^* = J_{drive} + G_{r1}^2 J_p, \quad k = 2K_L r_l^2, \quad K_L = \frac{A_b E}{L_b}$$

Where: r_l is the radius of the load pulley, A_b is the cross-sectional area of the belt, L_b is the length of the belt, and E is the modulus elasticity of the flexible belt ($E = 8 \times 10^6 \text{ N/m}^2$).

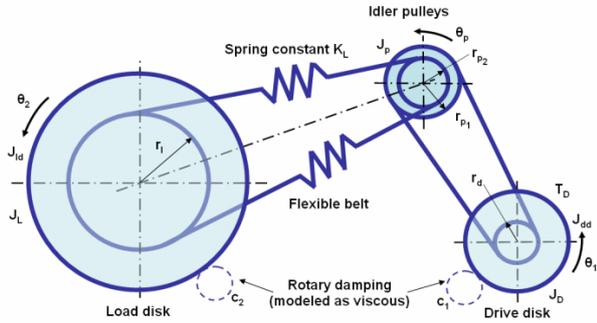


Figure 5. Model of flexible drive dynamics

Therefore, the plant transfer function between $\theta_l(s)$ and $T_D(s)$ is:

$$\frac{\theta_l(s)}{T_D(s)} = \frac{J_l s^2 + c_2 s + k}{D(s)} \quad (12)$$

Where:

$$D(s) = J_d^* J_l s^4 + (J_d^* c_2 + J_l c_1) s^3 + (J_d^* k + J_l G_r^{-2} k + c_1 c_2) s^2 + (c_1 k + c_2 G_r^{-2} k) s$$

Since the Laplace transform of the unit-step function is $1/s$, substituting $T_D(s) = 1/s$ results in the following transfer function for the PID controller:

$$G(s) = (k_p + k_d s) k_{hw} \quad (13)$$

The output/input transfer function for the control system is:

$$PID(s) = \frac{G(s)\theta_l(s)}{s[I + G(s)\theta_l(s)]} \quad (14)$$

$$PID(s) = \frac{k_{hw}(k_p + k_d s)(J_l s^2 + c_2 s + k)}{s(As^4 + Bs^3 + Cs^2 + Ds + E)} \quad (15)$$

Where:

$$A = J_d^* J_l$$

$$B = J_d^* c_2 + J_l c_1$$

$$C = J_d^* k + J_l G_r^{-2} k + c_1 c_2 + k_{hw} k_d c_2 + k_{hw} k_p J_l$$

$$D = c_1 k + c_2 G_r^{-2} k + k_{hw} k_p c_2 + k_{hw} k_d k$$

$$E = k_{hw} k_p k$$

Solving the quartic equation

$$As^4 + Bs^3 + Cs^2 + Ds + E = 0 \quad (16)$$

By expanding the powers and introducing new coefficients

$$\alpha = \frac{-3B^2}{8A^2} + \frac{C}{A}$$

$$\beta = \frac{B^3}{8A^3} - \frac{BC}{2A^2} + \frac{D}{A}$$

$$\gamma = \frac{-3B^4}{256A^4} + \frac{CB^2}{16A^3} + \frac{D}{A}$$

$$P = \frac{\alpha^2}{12} - \gamma$$

$$Q = -\frac{\alpha^3}{108} + \frac{\alpha\gamma}{3} - \frac{\beta^2}{8}$$

$$U = \sqrt[3]{\frac{Q}{2} \pm \sqrt{\frac{Q^2}{4} + \frac{P^3}{27}}}$$

$$y = -\frac{5}{6}\alpha - U + \begin{cases} 0 & U = 0 \\ \frac{P}{3U} & U \neq 0, \end{cases}$$

Then, the solutions are found to be:

$$x = -\frac{B}{4A} \pm \frac{1}{2}\sqrt{\alpha + 2y} \pm \frac{1}{2}\sqrt{-\left(3\alpha + 2y \pm \frac{2\beta}{\sqrt{\alpha + 2y}}\right)} \quad (17)$$

Then, the expansion of $PID(s)$ into partial fractions results in:

$$PID(s) = \frac{1}{s} - 2w \frac{s+c}{(s+c)^2 + \omega_1^2} + 2z \frac{\omega_1}{(s+c)^2 + \omega_1^2} - 2m \frac{s+a}{(s+a)^2 + \omega_2^2} - 2n \frac{\omega_2}{(s+a)^2 + \omega_2^2} \quad (18)$$

By taking the inverse Laplace transform of Eq. (18), we obtain the time solution for the flexible drive system for a step input:

$$y(t) = 1 - 2we^{-ct} \cos(\omega_1 t) + 2ze^{-ct} \sin(\omega_1 t) - 2me^{-at} \cos(\omega_2 t) - 2ne^{-at} \sin(\omega_2 t) \quad (19)$$

V. IMPLEMENTATION

A. Software Tools

These dynamic models for simulating the emulator system behavior were programmed into a Java applet. Jython [21] was used to write the Java applet. Jython represents an implementation of the Python programming language. It is written in pure Java, combines the advantages of Python and the Java virtual machine and libraries (such as Swing, Java Cryptography, Java API, etc.) and serves as a handy complement to the Java platform. For a user, the visible part of the emulator system is the graphical user interface (GUI) that was implemented as a Web page as shown in Figure 6. The GUI works in any Java enabled Web browser on any operating system. With the GUI of the emulator system, it is possible to specify the input parameters of the simulations, such as changing the plant model, adding, positioning and removing the masses on the drive and load disks, changing the numbers of teeth for the gears, adjusting the hardware gains, and controller parameters.

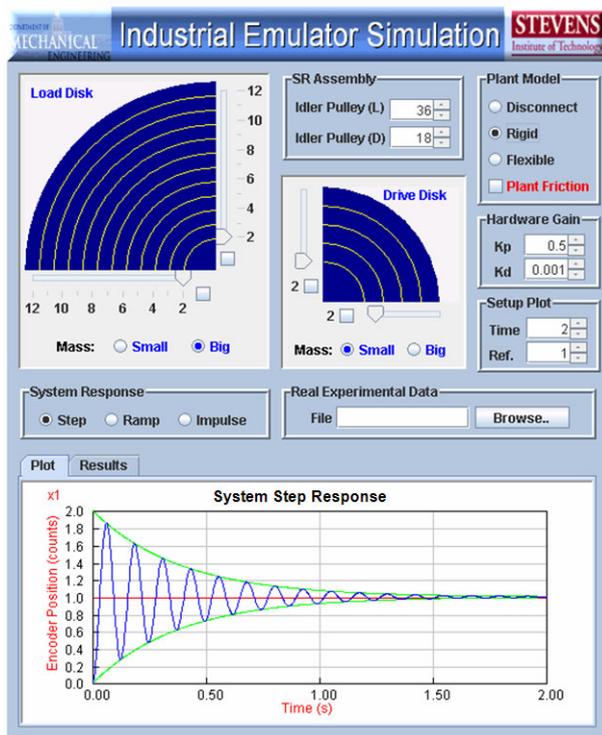


Figure 6. Java applet-based user interface of virtual emulator system

B. Virtual Emulator Features

Virtual experiments provide students the opportunity to conduct experimental procedures at any time that is convenient to them and to repeat them several times if so desired. As shown in Figure 7, the load and drive disks are depicted in the GUI as quarter circles, bordered by vertical and horizontal sliders. When any of four checkboxes located next to the disks are checked, the corresponding number of 200 g or 500 g masses is attached symmetrically to each disk. By means of sliders next to

the disks, the masses' desired radial locations measured from the disks' centers can be adjusted. The plant model is selected by clicking on the appropriate radio button for disconnected (i.e. removing the belts connecting the drive and load disks through the idler gear), rigid, or flexible drive dynamics. By checking the Plant Friction check box, friction effects can be included in the model. In order to investigate the effects of proportional, integral and derivative control on the performance of the dynamical system, one has to change the k_p and k_d values using the corresponding spinning buttons under hardware gain. The speed reduction assembly pulleys can be configured by spinning the N_{il} and N_{id} buttons to the desired values based on the real experimental setup. The Setup Plot box enables the students to adjust the desired time and the set point for the inverse Laplace transform function to generate the response curve. By selecting the Step radio button from the system response box, the simulation program automatically generates a two-dimensional plot for the step response of the system. The simulation program interactively calculates the natural frequency and damping ratio that define the dynamic system behavior (under-damped, over-damped and critically damped). Moreover, the students can also investigate other cases such as ramp and impulse responses.

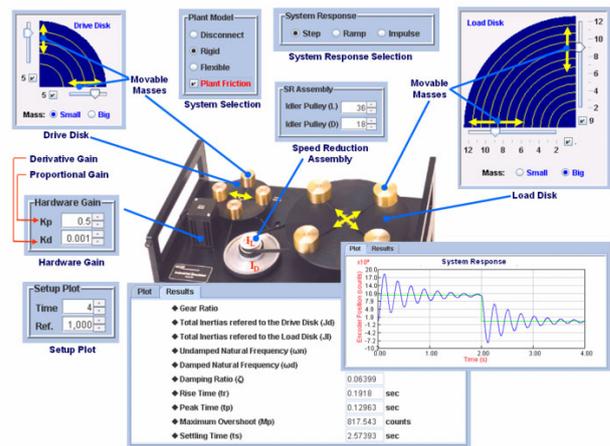


Figure 7. Java applet features

VI. VIRTUAL EXPERIMENT DESIGN

The Virtual Reality Modeling Language (VRML) [22] is used to generate and display the 3D virtual emulator experimentation system as shown in Figure 8. VRML was chosen for this environment as it represents a Web standard and free plug-ins are readily available for all major Web browsers. In addition, it does not require the user to have access to expensive hardware or software, and thus it can be easily adopted by a wide student spectrum. VRML is a text-based language and represents a powerful tool for the description of virtual worlds on the World Wide Web. It is platform-independent and includes capabilities for creating for instance 3D objects, light sources and animations. Using this tool, the students are enabled to get a fairly good feel for the 3D virtual emulator experimentation system.

Although the use of 3D objects in VRML may give the impression of being technically difficult to develop, in fact VRML is relatively straightforward to write, with simple text constructs defining the structure of the objects and the actions which can be carried out using them [23]. VRML

source files contain a definition for each object that appears in the environment, together with information about each object's location and attributes. For example, dynamic scene changes can be stimulated by scripted actions, message passing, user commands or behavior protocols, implemented using either via Java applets or complete VRML scene replacement [24]. Displaying realistic motions of objects in VRML is achieved using interpolators. The inputs and outputs of the interpolators are connected with other nodes via ROUTEs, which provide the wiring that connects the computations to the renderings. More detailed information about the VRML mechanism for animating models can be found elsewhere [25].



Figure 8. Comparison between real and virtual emulator experiments

A VRML plug-in is needed on the client side to interpret the 3D scenes. There are several good free plug-ins for most operating systems that easily self-install within the Web browser and enable the user to view VRML files (e.g. Cortona VRML viewer [26]). Once the plug-in is installed, viewing VRML files over the Internet or from the desktop is straightforward. The Cortona VRML plug-in works as a highly interactive VRML viewer for several Web browsers and office applications. It provides a motion control interface and facilitates the rendering of views via the graphics hardware. The three main navigation modes that the Cortona VRML Client offers are WALK, FLY and EXAMINE. The users can switch the navigation mode by clicking buttons on the vertical toolbar of the plug-in (e.g. clicking STUDY to enter the EXAMINE mode). The users can navigate with the mouse, the keyboard or both. Each navigation mode may have the options of PLAN, PAN, TURN and ROLL. The combination of a navigation mode and its option determines the possible camera motion and orientation. Also, the users can select a viewpoint from a list of predefined viewpoints.

All VRML objects of the emulator experiment are generated and controlled within the Java applet, and then the entire virtual model is interactively sent as text to the VRML environment on the students' computers. From the Java applet, the students can disconnect or connect the belts between the idler gears and the drive and load disks, respectively, control the location and the number of the masses on the drive and load disks, and specify the hardware gains (e.g. proportional and derivative gains) and time to animate the virtual emulator experiment. In addition, the students can walk inside the laboratory, examine the experiment from different positions, and animate the experiment to check how the motion is transferred from the drive disk to the load disk through the belt connections.

VII. STEVENS ONLINE LABORATORY ENVIRONMENTS

Over the last several years, various funded projects have been carried out at Stevens Institute of Technology (SIT) that aimed at conceiving, designing, implementing, testing and assessing innovative forms of the online laboratories for undergraduate engineering and science education [27-29]. The resulting Stevens Online Laboratory Environment (SOLE) enables both on-campus and distance-learning modes. Starting in 1998, several remotely accessible experiments were developed for usage in a sophomore-level core engineering course on dynamical systems [30]. These experiments were designed to be conducted in batch mode, implemented using LabVIEW [31] and included a one-degree-of-freedom mechanical vibration system, a liquid-level control system, a muffler system and a set of electrical systems based on operational amplifiers. Furthermore, an architecture for virtual laboratory experiments representing pure software simulations was devised [32], based on which a variety of virtual experiment simulations were implemented, including a virtual fluid mechanics laboratory [33] as well as simulations of a vibration system and a beam deflection system [34]. Before conducting an actual remote experiment, the students can view some relevant Web pages relating to the experiment to study the experimental setup and to read the available operating instructions by clicking on the links provided in the menu bar or in the left menu (see Figure 9). The system was implemented using a client-server network approach, which allows the concurrent execution of multiple experiments using separate experimental setups.

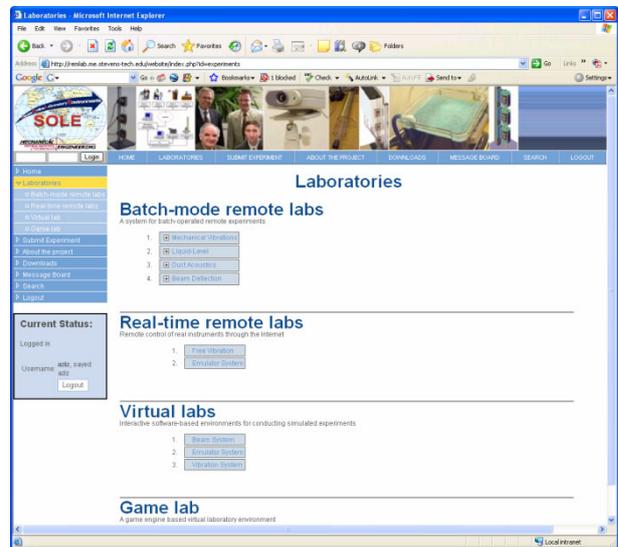


Figure 9. GUI of Stevens Online Laboratory Environment

When the students select the industrial plant emulator experiment at the simulation laboratory, the corresponding Java applet and the 3D virtual emulator experimentation system are launched on their local computers. When the applet is first loaded, a dialog window pops up in the Web browser, in which the student is asked to give full access permission to the applet for reading and writing files on the client computer. Granting these permissions at most gives to the applet the same privileges as any other executable program (stand-alone application) that one would run on the local computer.

Figure 10 shows the layout of the graphical user interface of the emulator system that the students see in their Web browser. It is composed of two main panels. The left command panel of the Java applet provides different input options for the emulator simulation, such as selecting the plant model, specifying the hardware gains (e.g. proportional and derivative gains), adjusting the location of the masses on the drive and load disks, changing the idler pulleys as well as uploading experimental data from a remotely conducted physical experiment to compare with the simulation results. During the experimentation phase, changes in parameters and variables are immediately reflected numerically and graphically as a response to the students' actions. In the right panel, the VRML rendering of the simulation model is displayed, which aims to convey a strong feeling of immersion. The students can navigate and examine the experimental setup, for example by rotating the view inside the virtual environment using the standard VRML navigation features. This allows them to move around the emulator experiment model and view it from any angle while the experiment is animated. Furthermore, the students can zoom in at any position to see the components of the experiment in more detail. In addition, the Java applet saves on the student's local computer the Excel data and VRML files for future use. The students can utilize these two files for plotting and fitting of data to determine the system response and animate the 3D virtual emulator experimentation system.

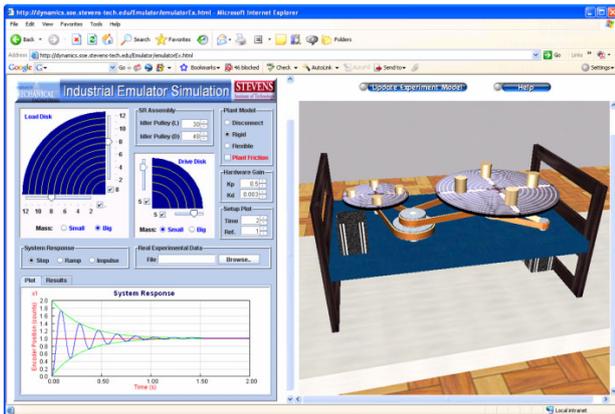


Figure 10. Components of user interface at client site for virtual emulator experiment

Currently, three experiments that use the described laboratory environment are available to measure and analyze various inertia values and the gear ratios for the rigid and flexible plants. In the hands-on laboratory employing the physical emulator setup, the students have to place a number of masses on the drive and load disks, adjust variable inertia values by moving or removing masses, adjust the speed reduction ratio by interchanging the belt pulleys in the speed reduction assembly, and replace the timing belt with a flexible belt. In order to implement all these capabilities remotely, it would be necessary to have multiple modules of the same hardware configured with different experimental parameters. This would quite be costly. One solution to this problem is to change the experimental setup manually to allow the students to run all experiments remotely based on a certain time schedule. An alternative solution is to combine the remote experiment using one particular setup with virtual

experiments, in which a wide range of variables, initial conditions, time intervals and proportional-integral-derivative (PID) controller parameters can be explored. In addition, the virtual laboratory module provides a very realistic representation of the experimental device and its components, thus providing the students with a strong feeling of immersion, as if they were performing a real-world experiment in a traditional laboratory.

VIII. EXPERIMENTAL VALIDATION

Based on the mathematical model developed earlier, a system simulation was performed to verify the performance of that mathematical model. The experimental validation was based on data collected in a real experiment. The simulation system comprises an additional feature for comparing the simulation results with data obtained from a real experiment, which can be imported from the control software. By clicking on the Browse button, the students can load the real data file, and the simulation program then combines and plots the real and simulation data, thus allowing them to compare the simulation system response with that from the real experiment as shown in Figure 11 and Figure 12. These figures show that there is quite good agreement between the experimental and simulation results for the industrial emulator system. In addition, the simulation system automatically generates a text file every time when new input data are entered. Utilizing this text file in conjunction with real data imported from the control software connected to the real system, the students can use an external analysis software (e.g. EXCEL) to plot the system response for the simulation and real experiments.



Figure 11. Comparison of real and simulation results (rigid plant)

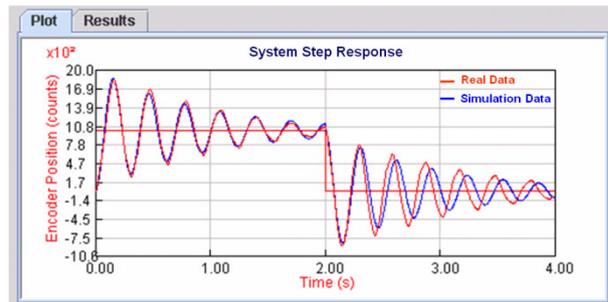


Figure 12. Comparison of real and simulation results (flexible plant)

IX. CONCLUSIONS

Simulation-based virtual experiments offer several distinct advantages over traditional hands-on and real-time remote experiments while providing the students with the same level of understanding of the underlying physical concepts. This paper describes how to deliver a laboratory experience involving an industrial plant emulator system

via the Internet. The virtual laboratory produces output responses that are similar to those of the corresponding real physical experiment setup. The VRML visualization of the real physical experiment offers a very realistic representation of the experimental device and its components, thus providing the students with a strong feeling of immersion, as if they were performing a real-world experiment in a traditional laboratory.

The experience gained through this work will benefit other mechanical engineering laboratories at SIT by making the engineering laboratories more accessible to the students and hence providing them with more opportunities for experimentation and self-paced learning.

ACKNOWLEDGMENT

This work was supported by NSF Grants No. 0326309 and 0817463. This support is gratefully acknowledged.

REFERENCES

- [1] Yuen, S. & Naidu, S., (2007), "Using multimedia to close the gap between theory and practice in engineering education", *International Journal of Engineering Education*, Vol. 23, No. 3, pp. 536-544.
- [2] Tzafestas, C. S., Palaiologou, N. & Alifragis, M., (2006), "Virtual and remote robotic laboratory: comparative experimental evaluation", *IEEE Transactions on Education*, Vol. 49, No. 3, pp. 360-369. doi:10.1109/TE.2006.879255
- [3] Javidi, G. & Sheybani, E., (2006), "Virtual engineering lab", *Proceedings of the 36th ASEE/IEEE Frontiers in Education Conference*, October 28-31, San Diego, CA, USA.
- [4] Alam, F., Tang, H. & Tu, J., (2004), "The development of an integrated experimental and computational teaching and learning tool for thermal fluid science", *World Transactions on Engineering and Technology Education*, Vol. 3, No. 2, pp. 249-252.
- [5] Palanki, S. & Kolavennu, S., (2003), "Simulation of control of a CSTR process", *International Journal of Engineering Education*, Vol. 19, No. 3, pp. 398-402.
- [6] Chi, X., & Spedding, T. A., (2006), "A Web-based intelligent virtual learning environment for industrial continuous improvement", *Proceedings of the IEEE International Conference on Industrial Informatics*, Singapore, pp. 1102-1107.
- [7] Erdman, A. G. & Riley, D. R., (1981), "Computer-aided linkage design using the LINCAGES package", *ASME Paper*, No. 91, DET 121.
- [8] Slater, K. & Gramoll, K., (1995), "Vibration visualization using longitudinal vibration simulator (LVS)", *Proceedings of the ASEE Annual Conference and Exposition*, June 25-28, Anaheim, California, USA.
- [9] Shin, D., Yoon, E. S., Lee, K. Y. & Lee, E. S., (2002), "A Web-based, interactive virtual laboratory system for unit operations and process systems engineering education: issues, design and implementation", *Computers and Chemical Engineering Journal*, Vol. 26, pp. 319-330. doi:10.1016/S0098-1354(01)00749-9
- [10] Gillet, D., Geoffroy, F., Zeramdini, K., Nguyen, A. V., Rekik, Y. & Piguet, Y., (2003), "The Cockpit: an effective metaphor for Web-based experimentation in engineering education", *International Journal of Engineering Education*, Vol. 19, No. 3, pp. 389-397.
- [11] Hoyer, H., Gerke, M. Masar, I. Ivanov, I. Röhrig, C. & Bischoff, A., (2003), "Virtual laboratory for real-time control of inverted pendulum/gantry crane", *Proceedings of the 11th IEEE Mediterranean Conference on Control and Automation*, June 18-20, Rhodes, Greece.
- [12] Virtual Engineering/Science Laboratory: <http://www.jhu.edu/~virtlab/virtlab.html>.
- [13] Catlin, A. C., Gaitatzes, M. G., Houstis, E. N., Ma, Z., Wang, N.-H. & Weerawarana, S., "The SoftLab experience: building virtual laboratories for computational science", Online document available at: http://www.cs.purdue.edu/research/cse/softlab/softlab-vlabs/softlab-framework/softlab_report/report.html.
- [14] Richardson, J., Adamo-Villani, N., Carpenter E. & Moore, G., (2006), "Designing and implementing a virtual 3D microcontroller laboratory environment", *Proceedings of the 36th ASEE/IEEE Frontiers in Education Conference*, October 28-31, San Diego, CA, USA.
- [15] Machotka, J. & Nedic, Z., (2006), "The remote laboratory NetLab for teaching engineering courses", *Global Journal of Engineering Education*, Vol. 10, No. 2, pp. 205-212.
- [16] Logofatu, B., Munteanu, M. & Logofatu, M., (2006), "Studies of waves and oscillations with data acquisition systems", *Romanian Journal of Physics*, Vol. 51, No. 1/2, pp. 13-20.
- [17] Industrial Plant Emulator System, Model 220, by Educational Control Products (ECP): <http://www.ecpsystems.com/>.
- [18] Remoteyanywhere homepage at: <http://www.remoteyanywhere.com/>
- [19] Nickerson, J. V., Corter, J. E., Esche, S. K. & Chassapis, C., (2006), "A model for evaluating the effectiveness of remote engineering laboratories and simulations in education", *Computers & Education - An International Journal*, Vol. 49, No. 3, pp. 708-725.
- [20] Ogata, K., (2001), "Modern Control Engineering", Fourth Edition, Prentice Hall.
- [21] Jython homepage at: <http://www.jython.org/>
- [22] Virtual Reality Modeling Language (VRML): <http://en.wikipedia.org/wiki/VRML>.
- [23] Allen, R., (1998), "The Web: interactive and multimedia education", *Journal of the Computer Networks and ISDN Systems*, Vol. 30, pp. 1717-1727. doi:10.1016/S0169-7552(98)00200-1
- [24] Brutzman, D., (1998), "The virtual reality modeling language and Java", *Communications of the ACM Archive*, Vol. 41, No. 6, pp. 57-64. doi:10.1145/276609.276620
- [25] Aziz, E.-S., Esche, S. K. & Chassapis, C., (2007), "Content-rich interactive online laboratory systems", Accepted to publish into the *Computer Applications in Engineering Education Journal*.
- [26] Cortona VRML client at: <http://www.parallelgraphics.com/products/cortona/>.
- [27] Stevens Online Laboratory Environments (SOLE) at: <http://remmlab.me.stevens-tech.edu/website/>.
- [28] Esche, S. K., Prasad, M. G. & Tsatsanis, M., (1998), "Development of a remotely accessible dynamical systems laboratory for undergraduate teaching", NSF-ILI Award No. 9851039.
- [29] Esche, S. K., Corter, J. E., Chassapis, C. & Nickerson, J. V., (2003), "ITR: An infrastructure for designing and conducting remote laboratories", NSF-ITR Grant No. 0326309.
- [30] Esche, S. K., Prasad, M. G. & Chassapis, C., (2000), "Remotely accessible laboratory approach for undergraduate education", *Proceedings of the ASEE Annual Conference and Exposition*, June 18-21, St. Louis, Missouri, USA.
- [31] LabVIEW by National Instruments Corporation, <http://www.ni.com/labview/>.
- [32] Aziz, E.-S., Esche, S. K. & Chassapis, C., (2007), "Enhancing the learning experience using simulation and experimentation to teach mechanical vibrations", *Proceedings of the 2007 ASEE Annual Conference and Exposition*, June 24-27, Honolulu, Hawaii, USA.
- [33] Jia, R., Xu, S., Gao, S., Aziz, E.-S., Esche, S. K. & Chassapis, C., (2006), "A virtual laboratory on fluid mechanics", *Proceedings of the 2006 ASEE Annual Conference and Exposition*, June 18-21, Chicago, Illinois, USA.
- [34] Aziz, E.-S., Esche, S. K. & Chassapis, C., (2006), "A scalable platform for remote and virtual laboratories", *World Transactions on Engineering and Technology Education*, Vol. 5, No. 3, pp. 445-448.

AUTHORS

Dr. El-Sayed Aziz holds a faculty position in the Mechanical Engineering Department at Mansoura University, Egypt. Currently, he is working as research scientist at Stevens Institute of Technology in Hoboken,

New Jersey, USA. He received B.S. and M.S. degrees in Mechanical Engineering from Mansoura University, Egypt, in 1991 and a Ph.D. in Mechanical Engineering from Stevens Institute of Technology in 2003. His research interests include knowledge-based engineering systems; computer-integrated design and manufacturing; Finite Element Analysis; software development and applications; as well as remote and virtual laboratories.

Dr. Sven K. Esche is currently holding a position as Associate Professor of Mechanical Engineering at Stevens Institute of Technology in Hoboken, New Jersey, USA. In 1989, he received an undergraduate degree in Applied Mechanics from Chemnitz University of Technology (Germany). After working for three years at Mercedes Benz AG in Stuttgart (Germany), he obtained M.S. and Ph.D. degrees in Mechanical Engineering from The Ohio State University in Columbus, Ohio, USA in 1994 and 1997, respectively. His current research interests include multi-scale modeling of thermo-mechanical processing of metals, integrated product and process design under

conditions of uncertainty and risk as well as remote sensing and control of distributed devices with special focus on remote laboratories.

Dr. Constantin Chassapis is a Professor, the Director of the Mechanical Engineering Department and the Deputy Dean of the School of Engineering and Sciences at Stevens Institute of Technology. His research interests are in knowledge-based engineering systems; computer-aided design and manufacturing; structure-property modeling and characterization of polymers and polymer composites as well as in remotely controlled distributed systems. He has been an active member in ASME and SPE, and he has received a best paper award from SPE's Injection Molding Division, the distinguished Assistant Professor Award at Stevens Institute of Technology, an Honorary Master's Degree from Stevens Institute of Technology, and the Tau Beta Pi Academic Excellence Award.

Manuscript received January 7th, 2010. Published as resubmitted by the authors 25 April 2010.