Computation and Experimentation on the Web with Application to Internal Combustion Engines

ALLAN KIRKPATRICK
Department of Mechanical Engineering, Colorado State University

BRYAN WILLSON
Department of Mechanical Engineering, Colorado State University

ABSTRACT

The internal combustion (IC) engine is a complex engineering system with rich thermal/fluid science applications. We have developed Web-based software written in Java to introduce thermodynamics, fluid mechanics, and heat transfer applications typical of IC engines. In addition, an "on-line" engine research facility has been constructed to allow engine experimentation over the laboratory Internet using Web browsers. We include a discussion of pedagogical guidelines to be considered for effective on-line experiments. The engine hardware and architecture of the data acquisition/control/ communication systems are described.

I. INTRODUCTION

In this paper, we discuss World Wide Web (WWW) based computation and experimentation with application to internal combustion (IC) engines. There has been a great deal of work in the application of CD-ROM based multimedia technology to engineering education. Interactive graphics, video, and sound have been shown to enhance engineering students' learning.[1] The availability of Web-based teaching materials continues to increase, as detailed in reference 2. However, educational use of the Web for on-line computation and experimentation is not as well developed.

We have developed Web-based computational and experimental applications that illustrate the principles of thermodynamics, heat transfer, and fluid mechanics as applied to internal combustion engines. Devices such as internal combustion engines help students acquire comprehensive and insightful knowledge about the interrelationships and constraints in complex engineering systems, as discussed in references 3 and 4.

The objective of the work has been to enhance and complement classroom material in the thermal sciences with multimedia and laboratory applications of internal combustion engines that can be accessed by engineering students using Web browsers over the Internet. The materials are being used across the CSU Mechanical Engineering curriculum in an introductory sophomore thermal science course which integrates thermodynamics and heat transfer, a junior level heat transfer course, and a senior level engines elective class.

II. COMPUTATION ON THE WEB

In the past, teachers developing computational software would write a program in a procedural computer language such as Fortran, with text input and output, and distribute the executable code to students. The students would use the code on either university or personal computers for the solution of homework problems. When desktop software tools such as EES, MathCad and Matlab were introduced, students could solve engineering homework problems more easily with the aid of the graphical interfaces these tools provide.[5]

With the advent of the Web and associated programming languages such as Java, it is possible to think about new paradigms for computational software.[6] Computational software can be made available to anyone, at any time, using any type of computer. All that is required is that the student have a Java-supported Web browser. The student does not need a licensed copy of the commercial software. The problem of distribution of software upgrades is minimized, as updates can be posted on the Web. The software can also be included with introductory teaching materials. A compilation of Java applets written for engineering and science applications is available on the Web.[7] For example, an interactive Java applet for an introductory controls course, is given in reference 8.

Java has a number of features that are very useful for the development of Web-based engineering education modules. Most importantly, Java programs can be included as applications or "applets" in Web pages. Applets are transmitted from a server over the Internet and executed by a Web browser on a local or client desktop computer. Java programs are also machine independent, as they are compiled in a byte code that can be run by Unix, Windows, and Macintosh computers. Networking features are built into Java, so that data input and output need not necessarily be on the same machine on which the Java application is running, allowing communication and remote control of processes, such as on-line experiments. The networking features include input and output data streaming, threads, and sockets.

We have developed Web pages in thermodynamics, heat transfer, and fluid mechanics that function as an introductory tutorial to internal combustion engines.[9,10] When a computation is needed to illustrate a point, a Java applet can be accessed to perform the calculation. The Web pages have different applets embedded in the tutorial material, so that the computations are always performed in the proper context of the material being presented. The calculations range from simple algebraic equations to a Runge-Kutta differential equation solver and a linear system of equations matrix solver. The student uses the applet by entering parameter values on input boxes on the Web page. The computations are performed by the Java applet on the local client computer, and the results are plotted on the same page. The speed of computation thus depends on the client computing resources, not the network bandwidth. These Java programs downloaded and resident on a remote student's Web browser, can react to user input and change dynamically, not just run the same animation or sound over and over.

In the following sections we give more detail about the thermodynamics and heat transfer software.

A. Thermodynamics Computational Software

In standard introductory thermodynamics courses, the main system analyzed is a simple piston-cylinder process with expansion and compression. The system is modeled as a closed system undergoing adiabatic, isothermal, or isentropic changes. The students solve problems dealing with this system through application of the energy equations of thermodynamics and an equation of state, often the ideal gas equation.

In our thermodynamics Web-based module, basic models of the various thermal processes occurring in internal combustion engines are presented. The organization of the module is given in table 1. The internal combustion engine is a more complex system and as such is a suitable follow-on application to the simple piston-cylinder model. The most basic cycle, known as the Otto cycle, involves constant volume heat addition where the combustion is modeled as occurring instantaneously. The next levels of complexity are models with arbitrary heat release, where the heat addition due to combustion is given as a function of crank angle, followed by a model which incorporates heat transfer to the cylinder walls. The Web page for the Otto cycle with finite heat release is shown in figure 1. In these models, the thermodynamic state at each crank angle is expressed as a differential equation. Determination of the pressure, temperature, net work, and heat transfer requires solution of the governing energy and state equations.

The students vary typical engine parameters such the engine speed, cylinder bore and stroke, onset of combustion, and the duration of combustion, as shown in figure 2. The intent is to allow rapid determination of the effect of the major variables on engine performance and efficiency. The equation solver integrates governing equations given the initial conditions, and produces engine cycle plots and cycle statistics for several simultaneous cases, as shown in figure 3.

Engine Parameters
- Slider Crank Model
- Work and Efficiency
- Cylinder Volume, Surface Area
- Mean Effective Pressure

Engine Cycle Models
- Otto Cycle Model
- Otto Heat Release Model
- Otto Heat Release and Transfer Model
- Heat Release Fraction

Combustion
- Combustion Products
- Chemical Equilibrium
- Dissociation

Table 1. Elements of the Thermodynamics Web-based Software.

B. Heat Transfer Web-based Software

Typical systems in introductory heat transfer classes are composite walls and single component objects of various geometrical shapes. The internal combustion engine is a more complex system and a source of examples of almost every conceivable type of heat transfer, including multi-mode conduction, convection, and radiation. Temperatures and heat fluxes cover wide ranges in the various components of the internal combustion engine. In this module, of which a representative Web page is presented in figure 4, we discuss the heat transfer processes in the engine components, then consider the engine parameters and variables which affect the heat transfer processes. The various models of the heat transfer processes and example calculations of the heat transfer rates and coefficients are indicated in table 2.

Review of Heat Transfer
- Energy Flows in Engines
- Review of Conduction, Convection, and Radiation
- Resistance Modeling

Heat Transfer in the Piston and Cylinder
- Cylinder Heat Transfer Processes
- Cylinder Heat Flux and Temperature
- Piston Heat Transfer Process
- Engine Heat Transfer Correlations

Engine Heat Transfer Correlations
- Instantaneous Heat Transfer Coefficients
- Coolant Heat Transfer
- Exhaust System Heat Transfer

Table 2. Elements of the Heat Transfer Web-based Software.

The heat transfer modules have two fin performance programs which compute the temperature distribution and heat flow in fins used in air cooled engines. An example page, shown in figure 5, allows students to enter fin parameters such as length, thickness, thermal conductivity, as well as the number of nodes. The modules solve the steady state energy equation using a matrix inversion routine written in Java, and plot several cases simultaneously, as shown in figure 6. This useful tool allows students to determine experientially when a fin is "too long."

III. Experimentation on the Web

Currently, the number of well-equipped engine research laboratories at U.S. universities is declining due to the high costs of keeping a lab equipped with state-of-the-art equipment. Even at universities with well-equipped engine research facilities, access by undergraduate students is limited. At the same time, there is an increasing demand for students with a solid understanding of engine theory, engine control, engine applications engineering (gas compression, distributed utility generation, cogeneration), combustion analysis and emissions control. In addition, there is a need to increase the availability of specialized laboratory equipment for research use, and to give laboratory access to those who, for a variety of reasons such as physical disability or out-of-town location, are unable to be physically present in a laboratory.

A possible solution to such access problems is to make the laboratory equipment available on the Internet to remote users.[11] There have been a number of recent initiatives in on-line experimentation. For example, an on-line control laboratory is discussed in reference 12.

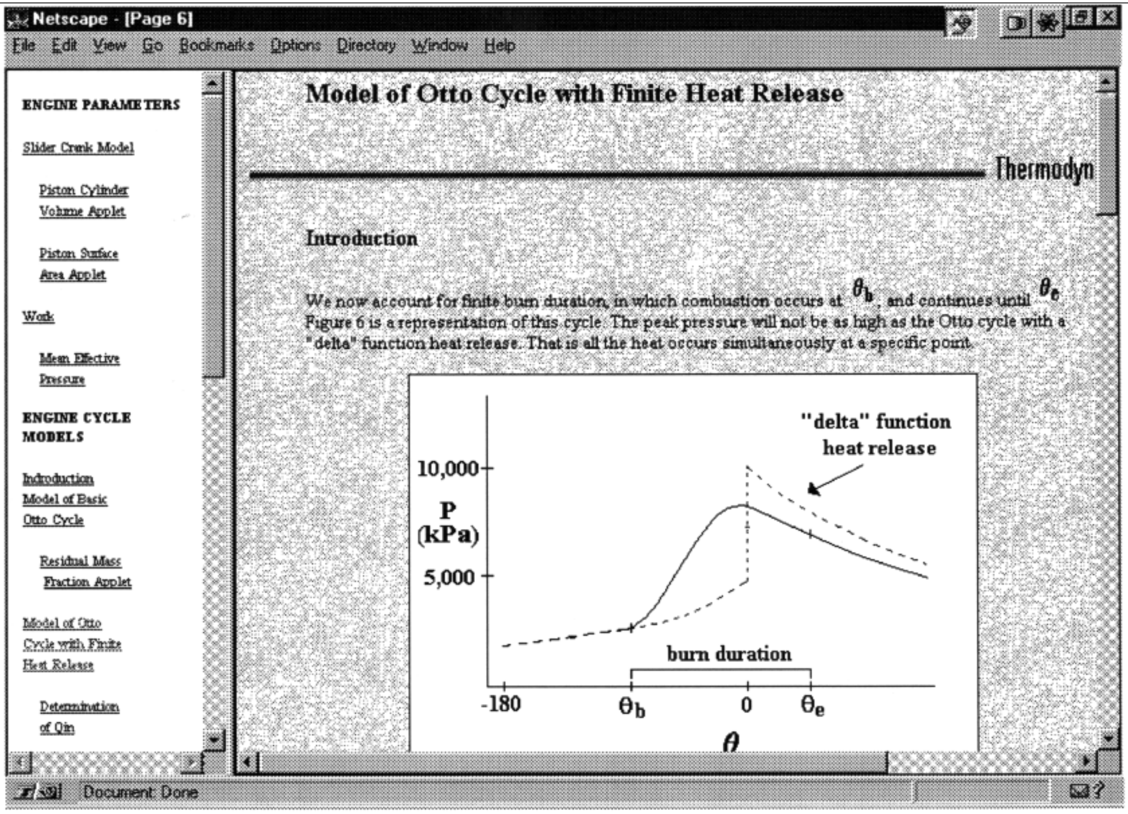

Figure 1. Representative thermodynamics Web page.

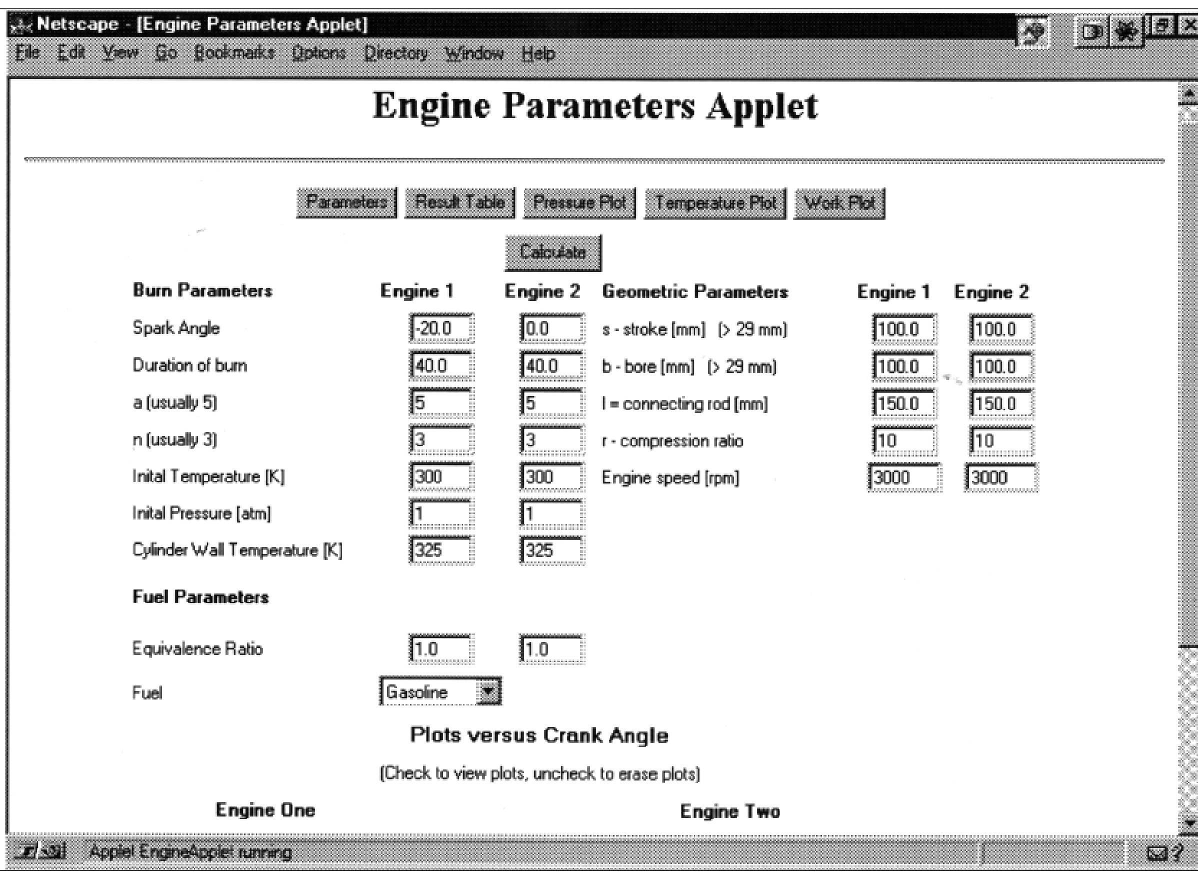

Figure 2. Sample applet for engine performance.

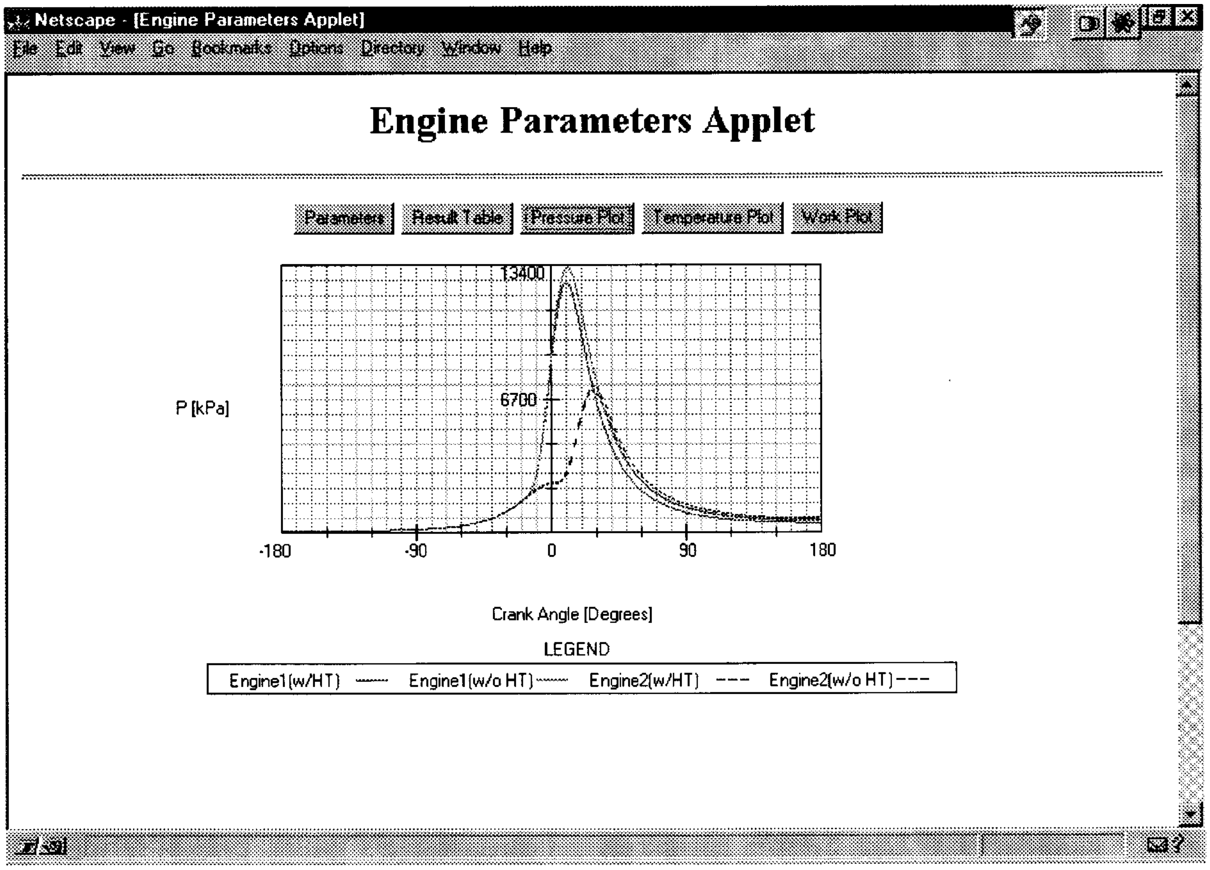

Figure 3. Sample applet for cylinder pressure.

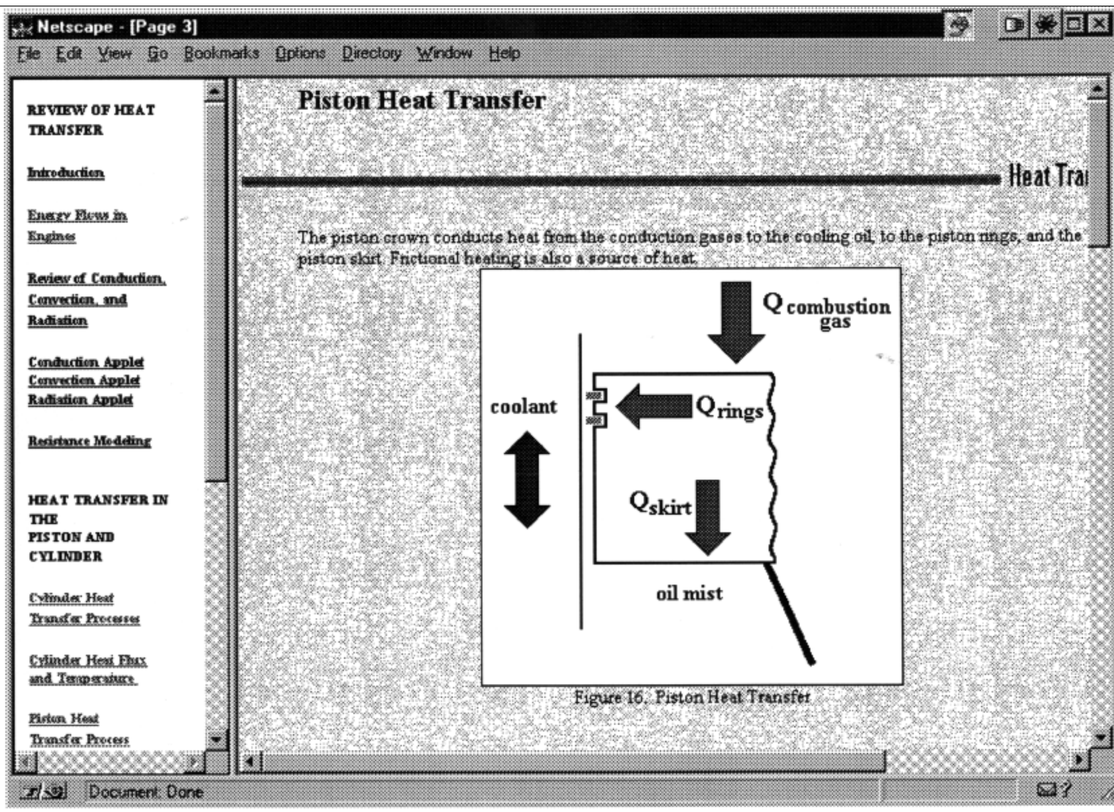

Figure 4. Representative heat transfer Web page.

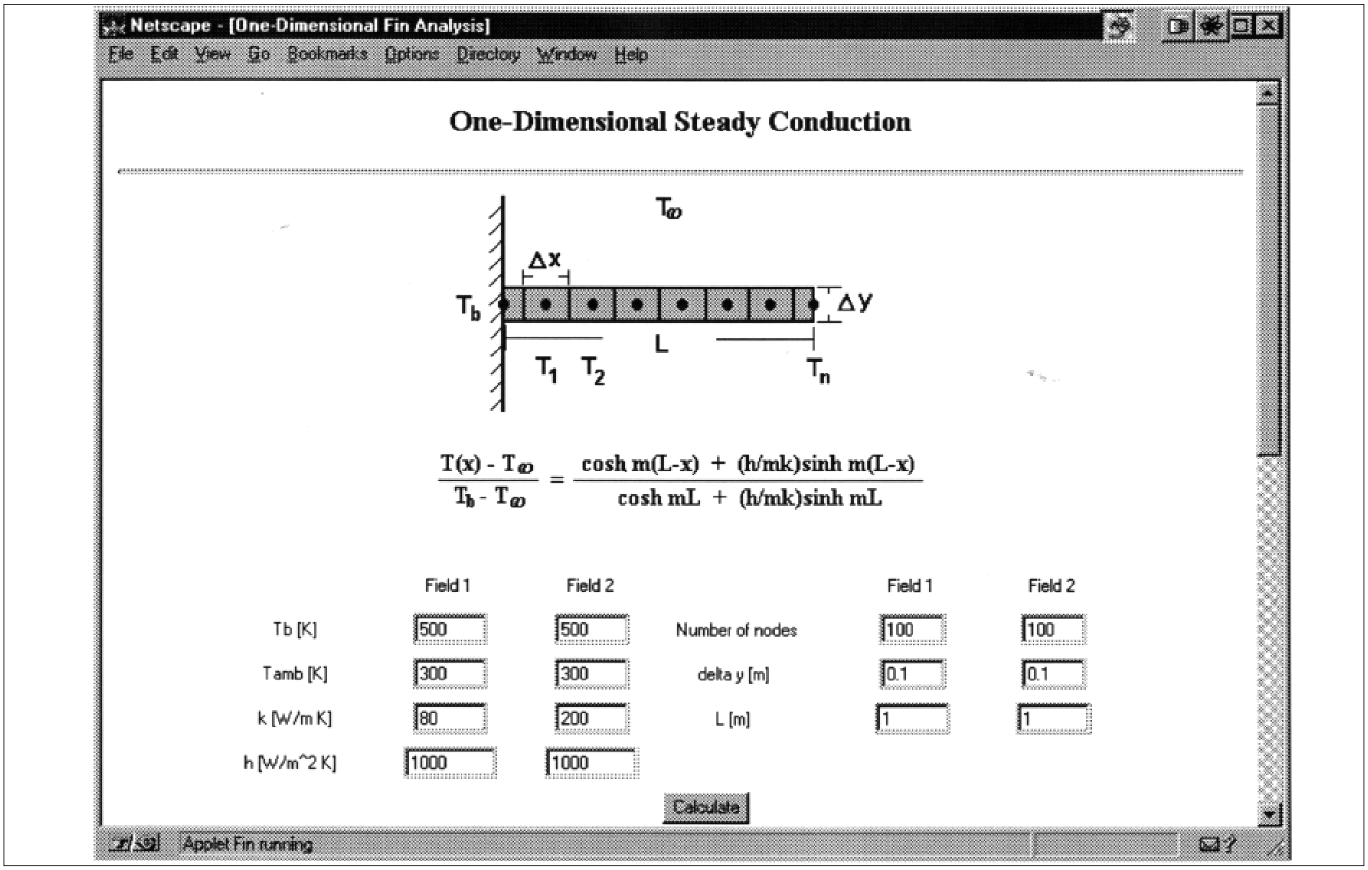

Figure 5. Sample applet for fin heat transfer.

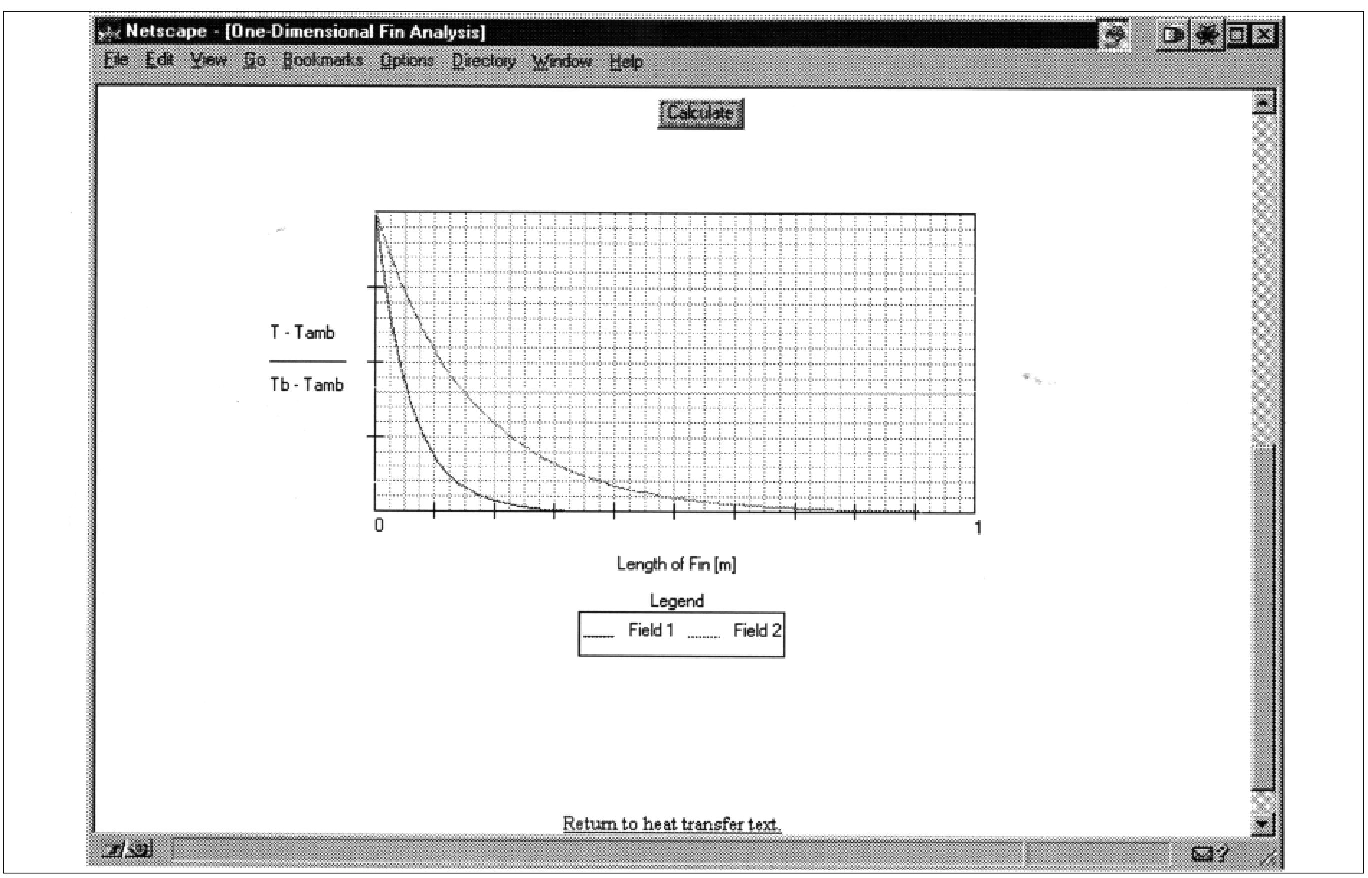

Figure 6. Sample applet for fin temperature distribution.

A. The CSU Global Engines Laboratory

We are developing a "Global Engines Laboratory" (GEL) at the Colorado State University Engines and Energy Conversion Laboratory (EECL) which can be accessed by students via the laboratory Internet. For security reasons, the laboratory is not currently available to outside users. Colorado State University students are able to "log-in" through a Web browser, and access control software, which displays a virtual control panel on their computer. The GEL presently consists of a modern gasoline engine, a DC dynamometer, an industrial data acquisition/control system, and extensive instrumentation. The engine, shown in figure 7, is a three cylinder, 25 hp, gasoline Briggs & Stratton/Daihatsu engine. The engine features overhead valves, modern combustion chamber shape, cast iron block, aluminum head, and electronic ignition. The engine itself is fitted with numerous sensors, as shown in figure 7. All data acquisition and control is performed with a robust industrial computer manufactured by Opto22. The engine is loaded with a DC dynamometer, which was developed at the EECL specifically for this application. The first test cell utilizes six thermocouples, three pressure transducers and four fluid flow meters. Currently, 24 input / output channels are used.

From an Internet Web browser, students in the laboratory can control engine load, speed, and throttle, and run a wide variety of engine tests. Basic tests will allow users to characterize torque, power, and efficiency with variations in throttle, timing, A/F ratio, etc. The engine data acquisition and control parameters are shown in figure 8, and a representative control panel is shown in figure 9. More advanced tests will allow exploration of combustion pressure, indicated work, rate of heat release, and emissions formation. Ultimately, there will be three test cells: one for gasoline, one for diesel fuel, and one for natural gas. The engines for all three test cells are built using the same engine block. The geometric similarity of the three engines facilitates direct comparison of performance on the different fuels. Capabilities will be added to control ignition timing and air/fuel ratio. Additional measurement capabilities will be added to allow measurement of engine emissions and combustion pressures.

B. Communications Model & Software

The GEL has been designed so that, in the near future, off-site users with a standard Web browser and access to the Internet can access and run experiments on the GEL test cells. As shown in figure 10, a remote user first navigates to the Web site of the GEL, which supplies necessary information on how to access the test cells, set up experimental schedules, etc. Interactive Java applets are downloaded and executed on the user's Web browser, which provides control capabilities over the test cell. The user can now send commands to the test cell using the control panel embedded in the Web page.

The Web-based communications architecture of the GEL is novel, so we will explain its operation in some detail. As shown in figure 11, the GEL software consists of three main components:

- *GELServer*: Custom server on the host laboratory computer
- *GELClient*: Java applet on the remote student computer
- *GELControl*: Control program on engine test cell controller

Data are continuously acquired from the test cell and sent to the controller. Local control loops are set up between the controller and the test cell. Desired channels of data are transferred from the controller to GELServer. Data requested by the user are

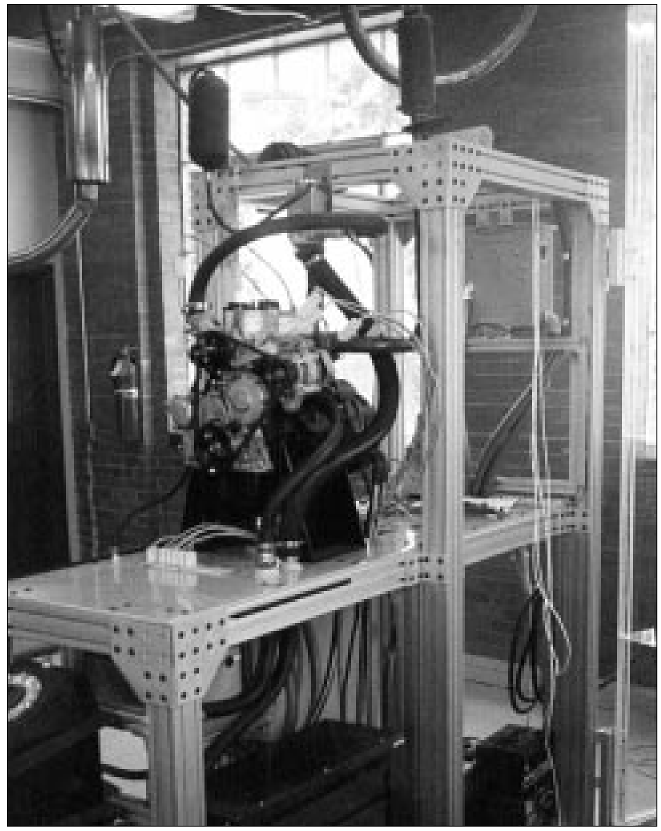

Figure 7. Photograph of on-line engine.

sent over the connection to the user, where they are displayed in graphical form. This process continues as long as the experiment is in progress, after which all connections to the user are shut down and GELServer awaits new connections. Specific implementation details are described in the following sections.

GELServer: This is a C program that runs on the host PC and serves primarily as a bridge between the remote user and the test cell. It creates a TCP/IP stream socket and listens for connection requests from remote users on a specific port. It opens communications to the Opto22 controller and then initializes and starts the control program (GELControl). The server goes into a wait mode until a connection request is received, which is then processed to authenticate the user. The program now enters a relay mode, in which it relays commands received from the user to the test cell controller, and data and status information from the test cell back to the user. The GELServer also contains a user interface that allows local monitoring of the test cell. Commands from the remote user may be overridden, or emergency shutdown procedures initiated through this interface, in case of unexpected or hazardous situations (say, loss of connection to user).

GELClient: GELClient is a Java applet that provides the remote user with the interface to the test cell. The applet is downloaded to the user's computer from the GEL Web site after authentication. It opens a TCP/IP connection to GELServer that remains open until the end of the session. The applet provides all necessary controls for the user to start, stop, and run various experiments on the test cell. Experimental data can be viewed in real time on graphs provided with the applet. At the end of the session the applet closes the remote connection. The same program runs on any remote computer irrespective of hardware platform or

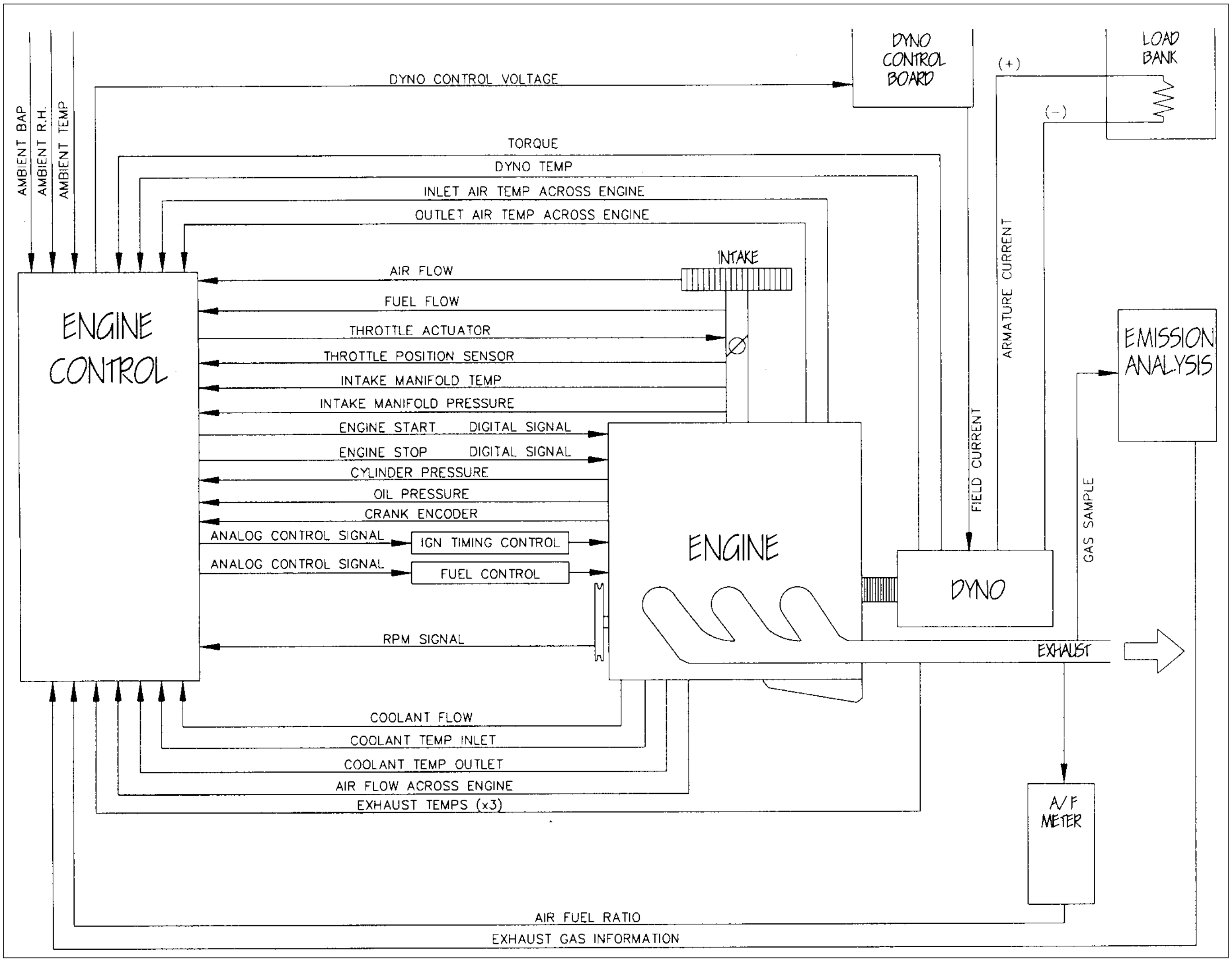

Figure 8. Engine data acquisition and control parameters.

operating system.

GELControl: This is the actual engine control program that resides on the Opto22 data acquisition/industrial control computer attached to the test cell. This program has been developed and implemented in Opto's proprietary control environment, OptoControl. At startup, the program goes through an initialization procedure and a diagnostic check to ensure that all test cell subsystems are powered up and functioning correctly. Any error is reported to GELServer, which alerts the lab personnel. When a start command is received, the control system takes the engine to an idling condition, after which it responds to user set points of speed, load, or throttle setting. Data from all available channels are streamed to the GELServer, which in turn streams them to the remote user. Algorithms built into the program ensure that at no time does the engine exceed safe operating limits.

C. Guidelines for Web-based Experimentation

From our experience, we have developed a set of guidelines which we believe are necessary for a high quality educational experience using a remote apparatus. We have found that the need to operate an experimental apparatus without on-site human input requires an extremely high degree of automation, requiring much higher level of development effort than would be the case for on-site operation.

Experimental Design:

- Remote experimentation appears to require more data channels than on-site experimentation; sensor data must be substituted for "sensory data" that the on-site user would normally have.
- The apparatus should be designed as an on-line experiment from the ground up.
- The apparatus should have enough flexibility to allow users to design their own experiments.
- The equipment must be more robust than is the norm for experiments; the remote user cannot "jiggle connections" as an on-site user can.
- In order to ensure high availability, the basic equipment should ideally be dedicated to remote use; expensive specialty equipment for advanced users may have to be shared.
- The equipment must provide a suitable amount of information for the medium, i.e. the "appropriate bandwidth."
- The "appropriate bandwidth" is expected to change frequently as the throughput of the Internet increases.

User Interaction

- Considerable effort must be addressed to the man / machine interface to facilitate efficient interaction.
- User training may be required before allowing access.
- A mechanism for scheduling / access control must be implemented.

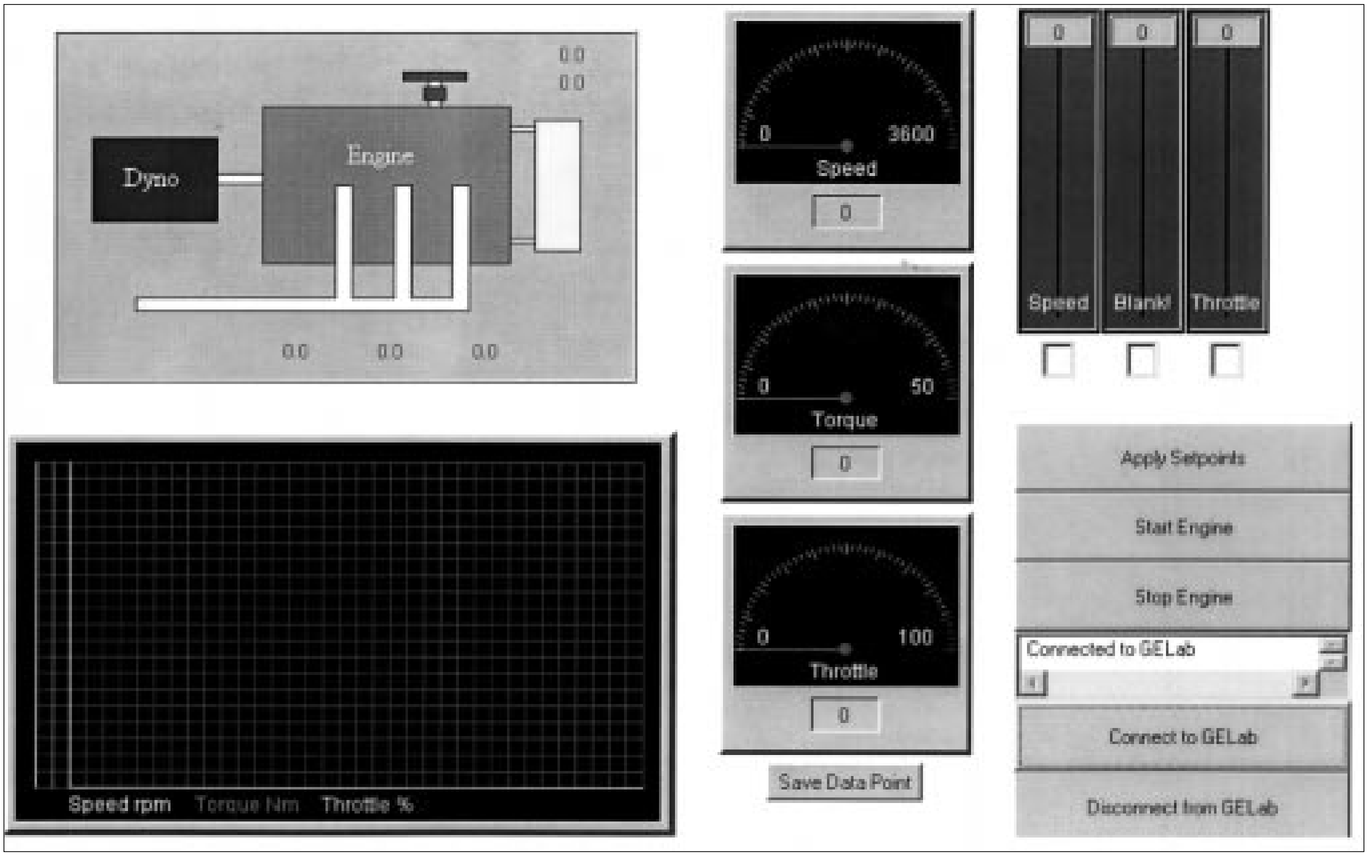

Figure 9. Sample control panel.

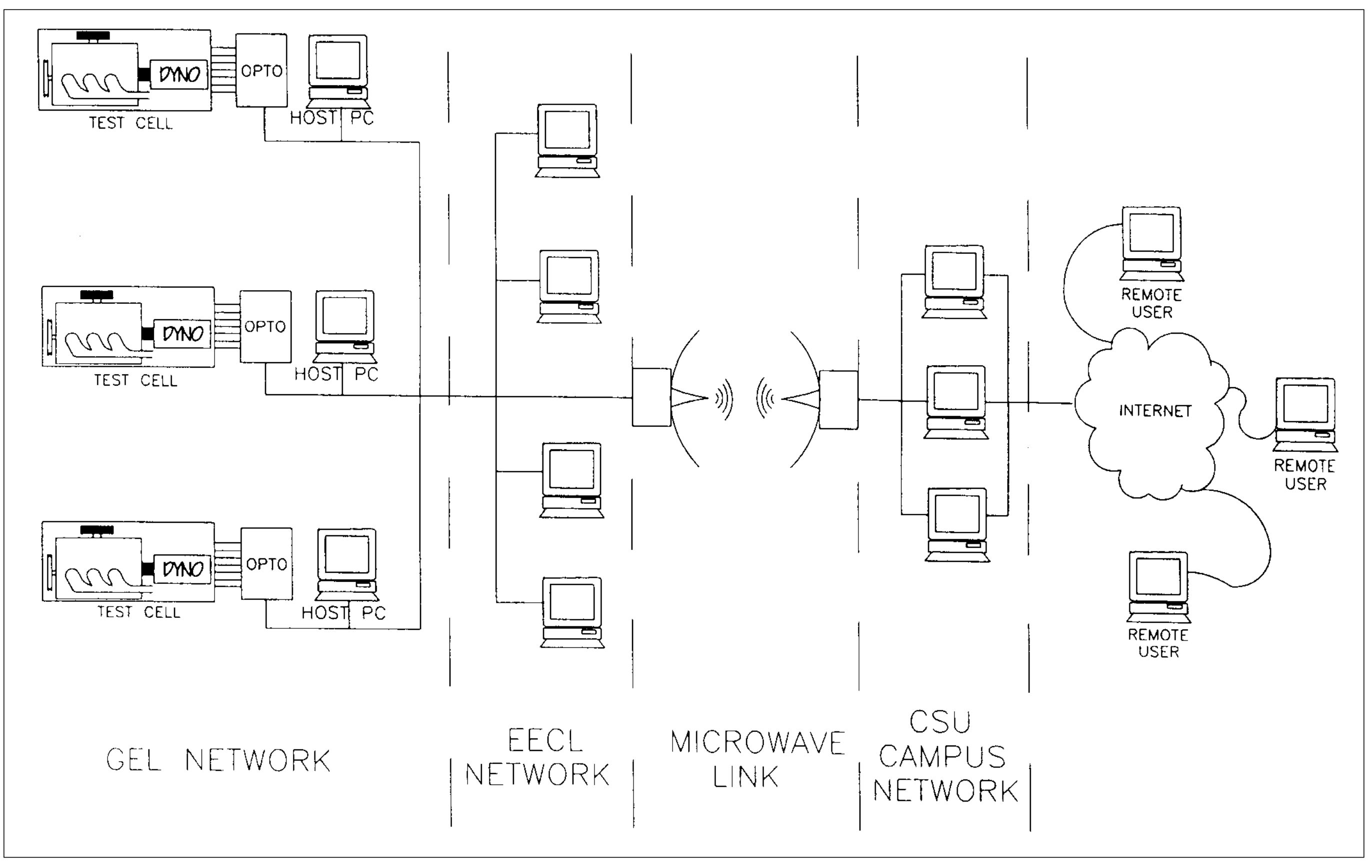

Figure 10. Global engine lab ethernet network.

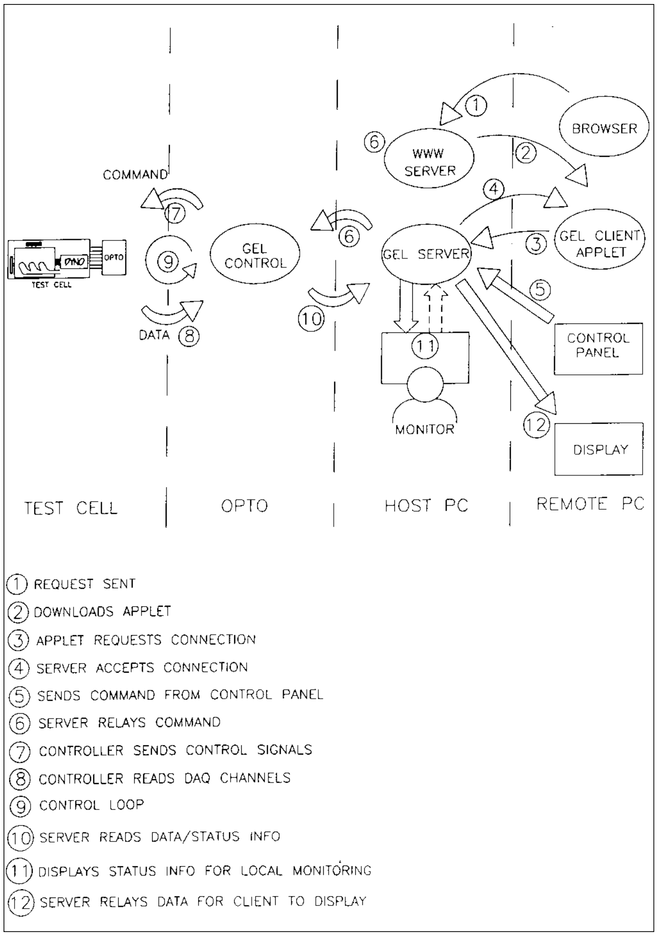

Figure 11. Web-based communication model.

- The system must have extensive safety interlocks; even a user who has been trained has little intuitive feel for "reasonableness" of results and danger conditions.

IV. Summary

In this paper, we discussed Web-based computation and experimentation with application to internal combustion (IC) engines. The work is currently under progress, with the evaluation and assessment components not complete. We have developed Web-based software applications written in Java to introduce thermodynamics and heat transfer applications of internal combustion engines. An "on-line" engine research facility has been constructed to allow engine experimentation over the Internet.

Acknowledgments

Mr. Sushil Cherian, Mr. Chris Hagen, and Ms. Amy Lee assisted with the Java programming and experiment development. This material is based on work supported by the Combined Research-Curriculum Development Program as National Science Foundation Award EEC-9527526. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Reagan, M., and S. Sheppard, "Interactive Multimedia Courseware and the Hands-on Learning Experience: An Assessment Study," *Journal of Engineering Education*, vol. 85, no. 2, 1996, pp. 123-131.

2. Wallace, D., and S. Weiner, "How Might Classroom Time Be Used Given WWW-Based Lectures?," *Journal of Engineering Education*, vol. 87, no. 3, 1998, pp. 237-248.

3. Vincenti, W., *What Engineers Know and How They Know It*, John Hopkins Press, Baltimore, MD, p. 112.

4. Woods, W.A., and M. Tindal, "Aspects of Teaching and Learning with Particular Reference to Internal Combustion Engines and Heat Engines," *International Journal of Engineering Education*, vol. 10, no. 4, 1994, pp. 355-360.

5. Harger, R. "Teaching in a Computer Classroom with a Hyperlinked, Interactive Book" *IEEE Transactions on Education*, vol. 39, no.3, 1996, pp. 327-335.

6. James, W., "Web-Based Organization of Relevant Knowledge for Web-Oriented Instruction and Training," *Proceedings, 1997Frontiers in Education Conference*, IEEE, 1997.

7. Gamelan: Java Applets for Engineering and Science, http://www.gamelan.com/

8. Tardif, P., and A. Zaccarin, "An Interactive Java Applet for Teaching Discrete Time System Theory," *Proceedings, 1997Frontiers in Education Conference*, IEEE, 1997.

9. Kirkpatrick, A., A. Lee, and B. Willson, "Development of Thermal/Fluids Web-Based Applications," *Proceedings, 1997 Frontiers in Education Conference*, IEEE, 1997.

10. The CSU Engines Modules.URL address: http://www.engr.colostate.edu/~allan/engines.html

11. Bourne, J., A. Brodersen, J. Campbell, M. Dawant, and R. Shiavi, "A Model for On-Line Learning Networks in Engineering Education," *Journal of Engineering Education*, vol. 85, no. 3, 1996, pp. 253-262

12. Aktan, B., C. Bohus, L. Crowl, and M. Shor, "Distance Learning Applied to Control Engineering Laboratories," *IEEE Transactions on Education*, vol. 39, no.3, 1996, pp. 320-326.