A HARDWARE-IN-THE-LOOP PC-BASED SYSTEM FOR DEVELOPMENT & TUNING CONTROL ALGORITHMS AND TRAINING

by

Sadashiva S. Godbole and Mike N. Vickers

Presented at the ISA/93 International Conference, Chicago, Illinois

September 20-23, 1993

A HARDWARE-IN-THE-LOOP PC-BASED SYSTEM FOR DEVELOPING & TUNING CONTROL ALGORITHMS AND TRAINING

Sadashiva S. Godbole
Advisory Engineer
B & W Nuclear Technologies
P.O. Box 10935
Lynchburg, VA, 24506-0935

and

Mike N. Vickers
Engineer
B & W Nuclear Technologies
P.O. Box 10935
Lynchburg, VA, 24506-0935

KEYWORDS

Control Algorithm, Controller Tuning, Hardware-in-the-Loop, I&C Training, PC-Based, Power Plant, Simulation, Ziegler-Nichol

ABSTRACT

The paper describes a hardware-in-the-loop system for developing and tuning control systems, training instrumentation and control technicians or automatic control students, and staging field changes before implementing in the field. The system includes a digital controller, a personal computer (PC)-based simulation of the process to be controlled, a communication link between the two based upon analog and digital input/output (I/O) hardware or serial/parallel data highway, and a graphical user interface for monitoring the process and interacting with it. The simulation PC and communication link can be configured in several ways to suit a particular application, e.g., stand-alone PC with I/O boards mounted in expansion slots, or a notebook PC with a docking station mounted in the controller rack, or a notebook PC communicating with a data acquisition system mounted in the controller rack over an RS-485 serial interface, or a dedicated PC mounted in the controller rack. The simulation system includes linear analysis capability and interface to MATLAB (a popular control analysis/synthesis software) for formal control algorithm design. The controller can be configured graphically with function blocks or directly in BASIC. The system is extremely cost effective in creating new control applications as well as maintaining existing ones. It is ideal for risk-free training and education.

INTRODUCTION

Traditionally, new control systems are debugged on-line during initial startup after installation. Subsequently, they are modified and fine tuned on-line as the process characteristics change with aging. The degree of risk associated with this method depends on the experience of the applications engineer who designs the control logic and the field service engineer who tunes the system. With the pervasion of PCs in real-time simulations and control applications, and introduction of inexpensive controllers in the market, there is little justification for taking the risk with on-line configuration and tuning.

Although control systems and personal computers are marketed individually, there are very few hardware-in-the-loop (HIL) systems on the market that synergistically integrate the control system with a non-trivial first-principles simulation of the process to be controlled. Examples of such HIL systems include compact simulators of fossil power plants and a system for developing a programmable logic controller application for factory floor[1]. However, compact simulators tend to be too costly and elaborate for the development and training application discussed in this paper.

The HIL system described in this paper provides an extremely economical and risk-free way for developing new control applications and maintaining existing control systems in the process industry. The system includes actual control hardware/software, a first-principles PC-based dynamic simulation of the process to be controlled with all its nonlinearities, and a communication link between the two based on actual analog and digital I/O, or serial or parallel data transfer interface. Affordable, expandable and flexible, the HIL system readily lends itself to developing control algorithms and configurations, tuning, training and finalizing field changes on the control hardware of choice in conjunction with a real-time model of the process to be controlled and a graphical user interface (GUI) for interaction.

HARDWARE-IN-THE-LOOP SYSTEM

HIL systems for control development and training can include analog or digital control systems such as Bailey 820, Westinghouse 7300, Bailey Infi90, or Foxboro I/A. The specific system described here and shown in **Figure 1** includes a digital controller consisting of three BWNT STAR Series 7000 modules, a Modular Modeling System (MMS)[2]-based simulation of a pressurized water reactor (PWR) on a PC, and a communication link between the two having analog and digital I/O at either end. The control algorithm configured in the STAR modules encompasses a variety of control characteristics found in typical control applications. For example, it includes loops with fast and slow dynamics, proportional-integral-derivative (PID) controllers, controller gain scheduling, feedforward, three-element level control, bumpless transfer between hand and auto modes, anti-reset windup, and low-pass filtering (i.e., first-order time lags). An alternative configuration utilizing STAR 721 or STAR 820 control modules mounted in Bailey 721 or 820 rack is also possible. The components of the HIL system are described next.

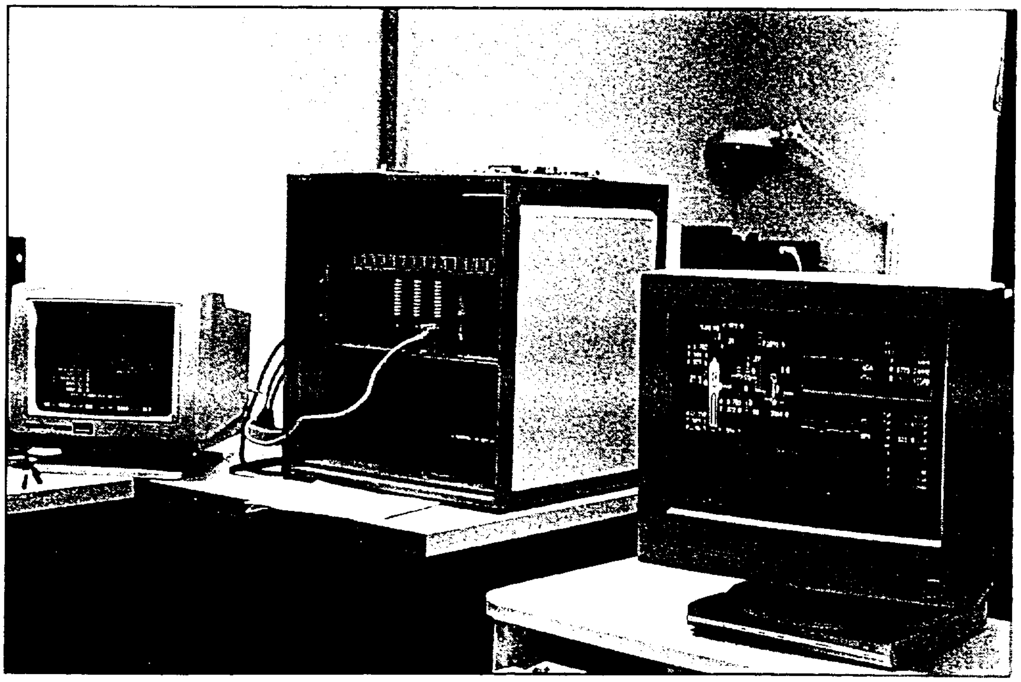


Figure 1 HIL system for developing control system and training

Controller: The STAR Series 7000 module is designed to replace Westinghouse 7300 analog modules in fossil and nuclear power plants. The module has enhanced functionality and can perform control algorithms rather than only one analog function. The physical configuration of the module varies so that it may mechanically and electrically fit in the plant's existing cabinets. It provides multiple analog and digital I/O. It is programmable by the user from function block (summer, multiplier, PID, etc.) level and/or using the BASIC language. The module also allows for on-line tuning by the user via an RS-232 interface.

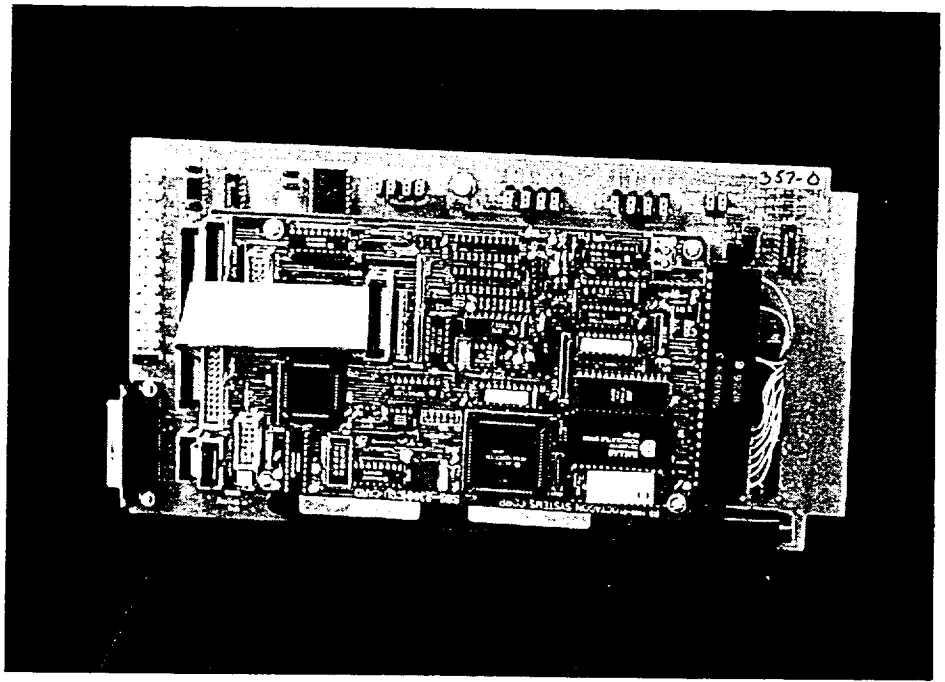


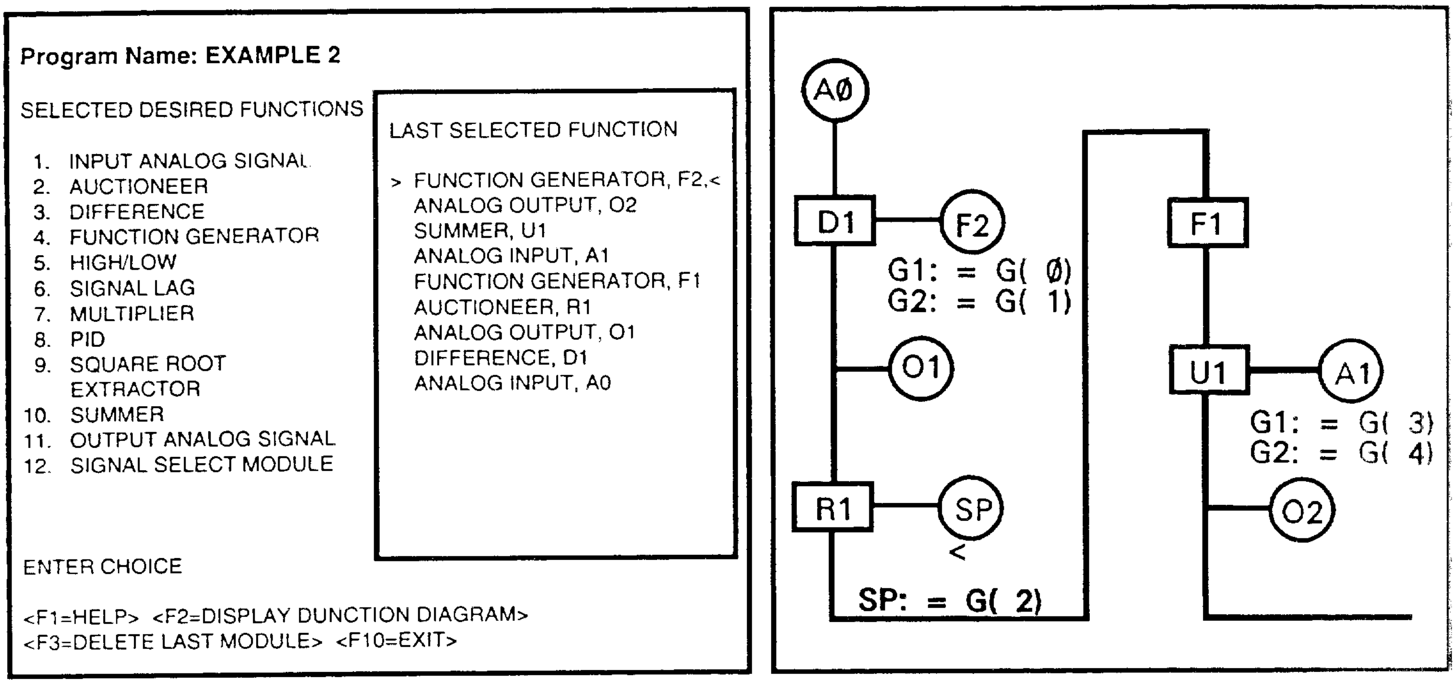
Figure 2 BWNT STAR Series 7000 module

The module contains a signal conditioning board and a computer board (see **Figure 2**). The signal conditioning board is sized to fit the dimensions of the rack in the cabinet and includes an identical connector to interface with the backplane. The computer board attaches to the signal conditioning board in a piggy-back configuration. The signal conditioning board has eight (8) analog inputs, two (2) analog outputs, eight (8) digital outputs, three (3) digital inputs, analog test points, and an RS-232 interface connector. The computer board, a product of Octagon Systems, includes a Z80 microprocessor, on-board programming with the Octagon's CAMBASIC™ , and 32K EEPROM for user program space.

An algorithm can be developed for the STAR Series 7000 module using BWNT's Software Application Management System (SAMS) on a personal computer. As shown in **Figure 3**, SAMS allows the user to construct control algorithms by selecting functional "building blocks" from a menu screen and then reviewing its graphical representation. The following functions are currently available with SAMS (new functions are added as needed):

- PID
- Summer
- Auctioneer
- Function Generator
- Signal Lag
- Inverter
- Bi-Stable
- Tri-Stable
- Square Root Extractor

SAMS outputs the control algorithm as a CAMBASIC program in an ASCII text file. The CAMBASIC



```
Program Name: EXAMPLE 2

SELECTED DESIRED FUNCTIONS

 1.  INPUT ANALOG SIGNAL
 2.  AUCTIONEER
 3.  DIFFERENCE
 4.  FUNCTION GENERATOR
 5.  HIGH/LOW
 6.  SIGNAL LAG
 7.  MULTIPLIER
 8.  PID
 9.  SQUARE ROOT
     EXTRACTOR
10.  SUMMER
11.  OUTPUT ANALOG SIGNAL
12.  SIGNAL SELECT MODULE

LAST SELECTED FUNCTION

> FUNCTION GENERATOR, F2,<
  ANALOG OUTPUT, O2
  SUMMER, U1
  ANALOG INPUT, A1
  FUNCTION GENERATOR, F1
  AUCTIONEER, R1
  ANALOG OUTPUT, O1
  DIFFERENCE, D1
  ANALOG INPUT, A0

ENTER CHOICE

<F1=HELP> <F2=DISPLAY DUNCTION DIAGRAM>
<F3=DELETE LAST MODULE> <F10=EXIT>
```

Figure 3 Control logic configuration using SAMS

language is a subset of the BASIC language. If the user desires, the program can be changed with any text editor to provide functions not mentioned above, such as H/A interface via the digital inputs

Once configured, the algorithm is loaded into the computer board's EEPROM using the on-line tuning portion of SAMS and the RS-232 interface on the module. With SAMS's on-line tuning, the user can change any of the constants and gains while on a workbench or in the plant's cabinets. When not changing the gains, SAMS provides the user with the current output of each function block within the control algorithm. **Figure 4** shows the tuning and monitoring screens of SAMS.

Simulation: The simulation is based upon a dynamic first-principles real-time model of a PWR with recirculating steam generators (SGs). The model features:

- a steam generator
- feedwater (FW) train

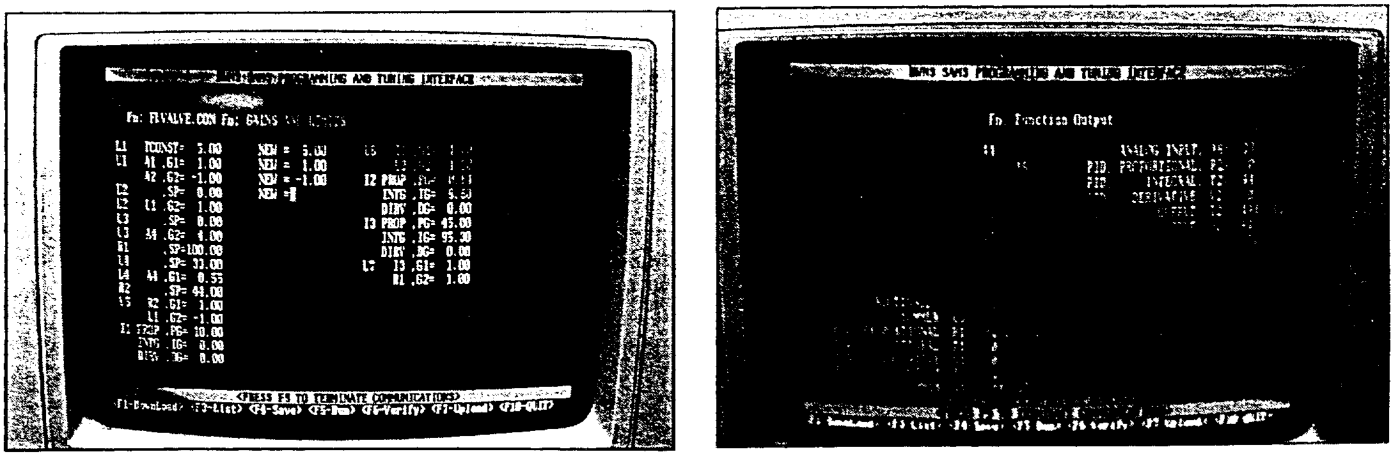


Figure 4 Tuning and monitoring screens of SAMS

- steam lines
- boundary conditions for primary-side flow and enthalpy, main FW pump suction pressure and enthalpy, and condenser pressure
- GUI.

The model can be extended to include a reactor, pressurizer, and other steam generators. The model runs on a 486 PC in real time, i.e., the simulation variables change at the same rate as the actual plant variables.

Models of other processes are also possible. The PWR model and similar power plant models are developed from the MMS, which includes pre-engineered and pre-tested modules for modeling PWRs, BWRs and fossil power plants. A module includes differential equations that describe heat transfer, pressure drop and conservation of mass, energy and momentum associated with the dynamic response characteristics of a power plant component. MMS incorporates features such as icon-based graphic model building, automatic parameter calculation, run-time interaction, and a choice of U.S. units or Standard International units. Icons are graphical elements and are used to represent MMS modules.

A typical dynamic simulation model development starts with the use of a Microsoft Windows™ -based graphic pre-processor (called MMS for Windows™) on a PC. An interconnection diagram, showing the arrangement of selected modules that represent the plant configuration, is prepared using a mouse to select, place and interconnect appropriate icons on the screen. Each icon is associated with a data form for entering physical and operating data for the represented plant component. The form includes a description of each item and a space for entering the value. To automatically calculate the values of module parameters, input values are entered in the data form. On activating the "auto-parameterization" function, the calculated parameters appear in the same form. Extensive on-line help is available.

After pre-processing, the model source code is automatically generated in Advanced Continuous Simulation Language (ACSL)[3], a powerful high-level simulation language. This code includes calls to ACSL macros corresponding to the modules used in the model, the calculated parameters, any custom ACSL coding, and custom FORTRAN or C routines included in model definition. The model code is then automatically translated to FORTRAN using the ACSL translator, and the FORTRAN code is compiled and linked to produce the executable model. The translator sorts and arranges the model calculations in the correct order - a great help in model building. The model developer needs only to focus on correctly specifying the MMS modules. The modeling system has an open architecture for including custom FORTRAN and C routines, such as routines for GUI and communication with other devices. Finally, the model is debugged and fine-tuned by running various transients and evaluating the responses. ACSL provides a powerful run-time interaction facility featuring:

- plotting
- parameter adjustment
- display of values of model variables
- a run-time choice from several numerical integration algorithms where the time step and/or order are fixed or variable
- linear analysis at steady state (e.g., linearized model, eigenvalues, poles and zeros, Nichols chart, Nyquist and Inverse Nyquist plots, and root locus)

- interface with MATLAB[4] (a popular control analysis/synthesis software)
- design optimization environment called OPTDES.

Applications involving man-machine interaction require the simulation to run in real time. The speed of the simulation depends upon the scope of the model, the level of detail, the nature of equations required to describe the processes, and the computing power available for running the simulation (i.e., 386, 486, workstation, or minicomputer). Most power plant models have fast dynamics (i.e., small time-constants), such as pressure and flow dynamics, and slow dynamics (i.e., large time-constants), such as temperature dynamics. Fast dynamics need to be processed more frequently and require more computer time than slow dynamics. The fastest dynamics (i.e., smallest time-constant) in the model governs the frequency of processing the model equations; thus a conventional model usually requires some modifications to allow the simulation to run in real time.

Control algorithms: Each STAR module controls one control function. The three control functions are:

- Function 1 (replaces 13 analog module functions):
  SG level control by modulating:
    the main FW valve using the three-element algorithm at high power
    the bypass FW valve using the single-element algorithm at low power
- Function 2 (replaces 10 analog module functions):
  Control of pressure differential between the FW header and the steam header by modulating the FW pump speed
- Function 3 (replaces 3 analog module functions)
  Steam header pressure control by modulating the turbine throttle valve.

Figure 5 shows the block diagram of FW pump control configuration.

Since the controller logic is in BASIC language, changes can also be made at the BASIC level. Even formal model-based control algorithms directly written in BASIC can be used. The on-line tuning system permits monitoring the calculated values at the output of various function blocks and changing gains and other parameters.

The level of a recirculating SG has a non-minimum phase response and therefore is quite difficult to control. Further, feedwater flow and steam flow measurements are too noisy at low power levels to be useful for feedback. Also FW pumps and valves have strong interaction. Thus, advanced control algorithms may be appropriate for obtaining desired response. Using the linear analysis capability of the simulation, the process response characteristics can be determined and used for control algorithm design. Also a linearized model of the process (e.g., state-space time-domain model) can be derived and input to MATLAB control analysis and design software tool for developing model-based control algorithms, such as linear regulator.

Tuning: The on-line tuning system that is a part of the STAR modules permits change of parameters, such as setpoints and controller gains, to tune the controller using methods like Ziegler-Nichol's Ultimate Response Method[5] whose target is a 1/4 decay ratio closed-loop response. Advanced methods, such as OPTDES for simultaneous optimization of multiple parameters, are also possible.

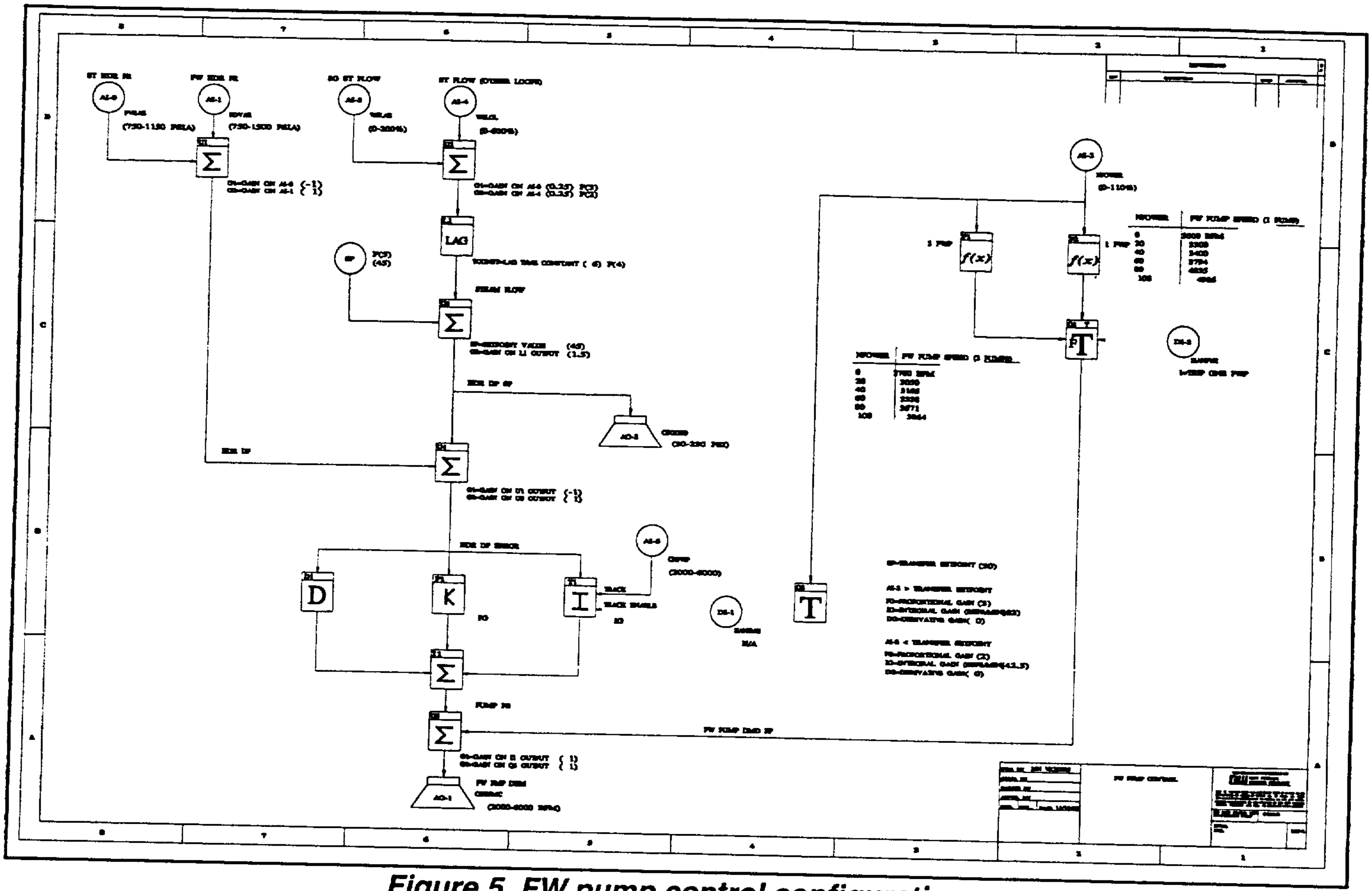


Figure 5 FW pump control configuration

As an example, the controllers in the STAR modules were tuned with the Ziegler-Nichol's method. To start controller tuning at a steady state, the integral gain is set to zero, the proportional gain is set to a low value, and a small step in setpoint is made. If the response as seen on the GUI is damped, the proportional gain is increased and the step change repeated until the response shows sustained oscillation; this response is called the Ultimate Response (UR). The value of the proportional gain at which UR is obtained is called the Ultimate Gain (UG) and the period of oscillation is called Ultimate Period (UP). The UP is measured by plotting the response over an expanded time scale using the ACSL "PLOT" command. **Figure 6** shows the approximate UR for the FW valve loop and the calculated controller parameters.

OTHER CONFIGURATIONS

The simulation PC and the communication link in the HIL system described here can be configured in many ways to suit user needs. Some options are:

- A regular stand-alone PC with I/O boards mounted in expansion slots as shown in Figure
- A notebook PC with a docking station mounted in the STAR rack (the docking station will have the I/O boards mounted in the expansion bus)
- A notebook PC that can connect to a data acquisition system mounted in the STAR rack over an RS485 serial cable[6]

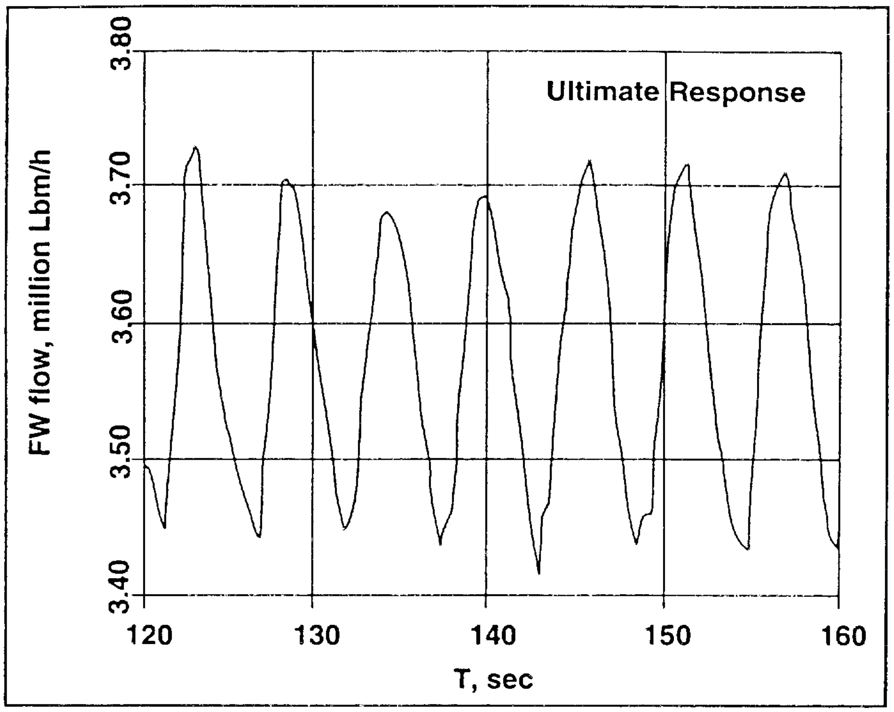


Figure 6 Tuning of FW valve control loop

Ultimate gain (UG) = 1.35
Ultimate period (UP)
= 10 sec/(1.8 cycles x 60 sec/min)
= 0.0926 min
Proportional gain = 0.45 UG
= 0.6075
Integral gain = UG/(UP/1.2) = 7.87

- A dedicated PC mounted in the STAR rack (see **Figure 7**).

CONCLUSION

The HIL system described in this paper is ideal for utility instrumentation and control training and maintenance departments and universities. The system provides a safe, yet realistic and extremely economical, learning and development environment. An HIL system can be made to match most user needs in terms of the controller, simulation, GUI, communication, and configuration. Linear analysis capability and interface with MATLAB permits evaluation of new control algorithms. The high-level controller language (BASIC) permits evaluation of new control concepts and strategies. The system can be used for staging field modifications in a safe environment before implementing them on the actual process[7].

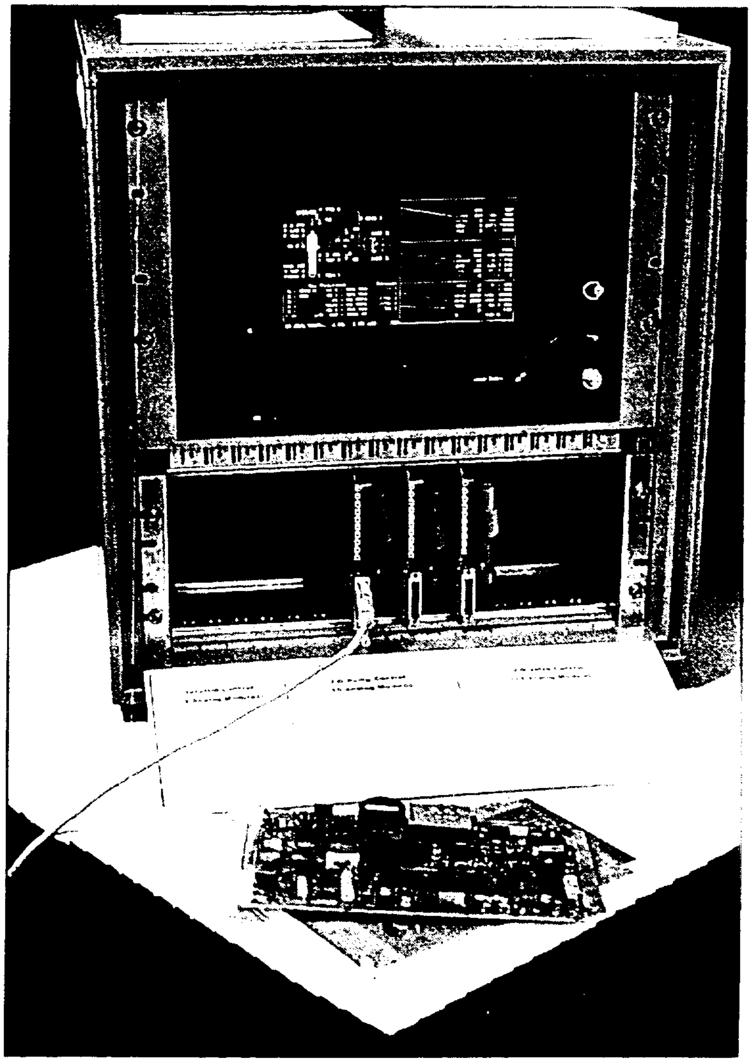


Figure 7 A standalone configuration for the HIL control development and tuning system

ABBREVIATIONS

ACSL	Advanced Continuous Simulation Language
FW	Feedwater
GUI	Graphical user interface
HIL	Hardware-in-the-loop
I/O	Input/output
MMS	Modular Modeling System
PC	Personal computer
PWR	Pressurized water reactor
PID	Proportional-integral-derivative
SAMS	Software Application Management System
SG	Steam generator
UG	Ultimate gain
UP	Ultimate period
UR	Ultimate response

REFERENCES

1. "PC-based Simulation In Control System Design," Keith Pritchard, Control Engineering, Mid-February 1993, pp. 31-32.

2. MMS for Windows User's Guide. B&W Nuclear Technologies, P. O. Box 10935, Lynchburg, VA, USA 24506-0935, 1993.

3. ACSL Reference Manual. Mitchell & Gauthier Associates, 200 Baker Avenue, Concord, Mass. 01742-2100, USA, 1991.

4. MATLAB. The Math Works, Inc, 24 Prime Park Way, Natick, MA 01760, USA.

5. Reference Guide to PID Tuning. Control Engineering, Cahners Reprint Services, 1350 E. Touhy Ave., Des Plaines, IL 60018, USA.

6. "Data Acquisition: Matching Tools to the Application," Harry Furness, Control Engineering, March 1993, pp. 66-69.

7. "Low-Cost/Risk Approach for Verifying Modifications to Dynamic Plant Operation and Instrumentation & Control System Resulting from Steam Generator Replacement," S. S. Godbole and G. F. Malan, 4th International Conference on Simulation Methods in Nuclear Engineering, Montreal, Canada, June 2-4, 1993.