DEMONSTRATION OF A METHODOLOGY TO CAPTURE PERFORMANCE REQUIRMENTS IN A REACTOR CONTROL SYSTEM DESIGN

José March-Leuba, James A. Mullens, Richard T. Wood, and Carlton R. Brittain
Oak Ridge National Laboratory
P.O. Box 2008; Oak Ridge, TN 37831-6004
MarchLeubaJA@ornl.gov, MullensJA@ornl.gov, WoodRT@ornl.gov, BrittainCR@ornl.gov

This paper presents the results of a non-trivial application of the Control Engine methodology that was presented in Reference 1. For this application, we show that this methodology can successfully "capture" the performance requirements of a steam generator level control for the life of the plant, even as the plant ages or unexpected operating conditions are encountered. This paper presents a demonstration of this methodology for a non-trivial application that uses the full-blown engineering simulator of a pressurized water reactor (PWR) described in Reference 2.

In the past two decades there have been significant progress in the automated development of control algorithms and on computer aids for fast and reliable implementation of those algorithms. Indeed, there are a number of commercial off-the-shelf software packages that can help design and parameterize control systems. Some of these packages have graphical interfaces, which facilitate the use of already-validated software modules. It is not the goal of this research to reproduce this existing work. The purpose of this research is to extend the current state of the art, so that the performance requirements are captured in a Control Engine, which is used during the life of the facility to confirm that the original requirements are still met as plant conditions change. Capturing these requirements is of special relevance to the nuclear industry, where the plant life is often 40 to 60 years, and the system requirements are not always obvious.

In our proposed methodology, we achieve this goal by reformulating the performance requirements as mathematical constraints of a minimization problem. For example, one such constraint could be that the steam generator control system must survive an anticipated over-cooling event without scram. The Control Engine runs in the background in supervisory mode and continuously evaluates whether these constraints are satisfied given the current state of the plant. If they are not, it starts an iterative minimization calculation that suggests to the operator optimal control parameter settings or even different control strategies if the current one is inadequate. Since changes to the plant over its 40 to 60 year life are slow in nature, we do not envision the Control Engine running in a closed loop, and automatically changing control parameters or strategies. Its function is more of an advisory nature by producing some kind of an alarm when the original control-system performance-requirements are not satisfied under the present conditions (e.g., hardware failures, or plant reconfiguration.) In addition to the alarm, the Control Engine can also suggest new control system settings that would satisfy the performance requirements under the present plant condition.

For this demonstration, we have developed Control Engine prototype software using standard off-the-shelf minimization algorithms and we have coupled it to several simulation programs. For the application, we have chosen a complex, high-fidelity PWR simulator (Ref 2). This PWR simulator is a large Fortran code, which we have coupled to the control engine without modification; thus demonstrating that this technique can be applied to essentially any engineering simulator. For this example, the performance requirements are defined as avoiding scram for: (1) a 10% power reduction, and (2) a 40 degrees F reduction in feed-water temperature; these are arbitrary requirements and other may have been chosen.

The results of the Control Engine optimization for the two above transients are shown in Figures 1 and 2; which show the steam generator level during the simulated transient with the original control parameter settings and with the optimized parameters. The thermal power (i.e., steam flow to the turbine) is controlled very accurately during the transient, and we did not observe any unussual neutron-flux power oscillations in the reactor core. These results are obtained by iterating using the simulator with different control parameters and choosing those parameters that minimize the overall error for both transients. Note that by using this minimization technique, we do not require to linearize or Laplace-transform the reactor model. This provides us with two relevant features: (1) we can use existing complex models “as is”, and (2) non-linear or non-minimum phase phenomena, such as the well-known shrink and swell effect, are inherently taken into account.

In summary, we have implemented the Control Engine concept in a generic software program that can be interfaced with minor effort to different simulators. We have successfully demonstrated the concept for a non-trivial case representative of a real-world reactor application.

REFERENCES

1. J. March-Leuba and R. T. Wood "Automated Generation of Algorithms for Nuclear Power Plant Control Systems," Nuclear Power Instrumentation, Control and Human-Machine Interface Technologies, Washington DC, Nov 13-17, 2000

2. J. Michael Doster, "Complex System Simulator for the Time Dependent simulation of Nuclear Power Systems," Proceedings of the 1999 ASEE Annual Conference and Exposition, Charlotte, NC, June 1999.

ACKNOWLEDGMENET

This research was sponsored by the U.S. Department of Energy Nuclear Energy Research Initiative (NERI) program.

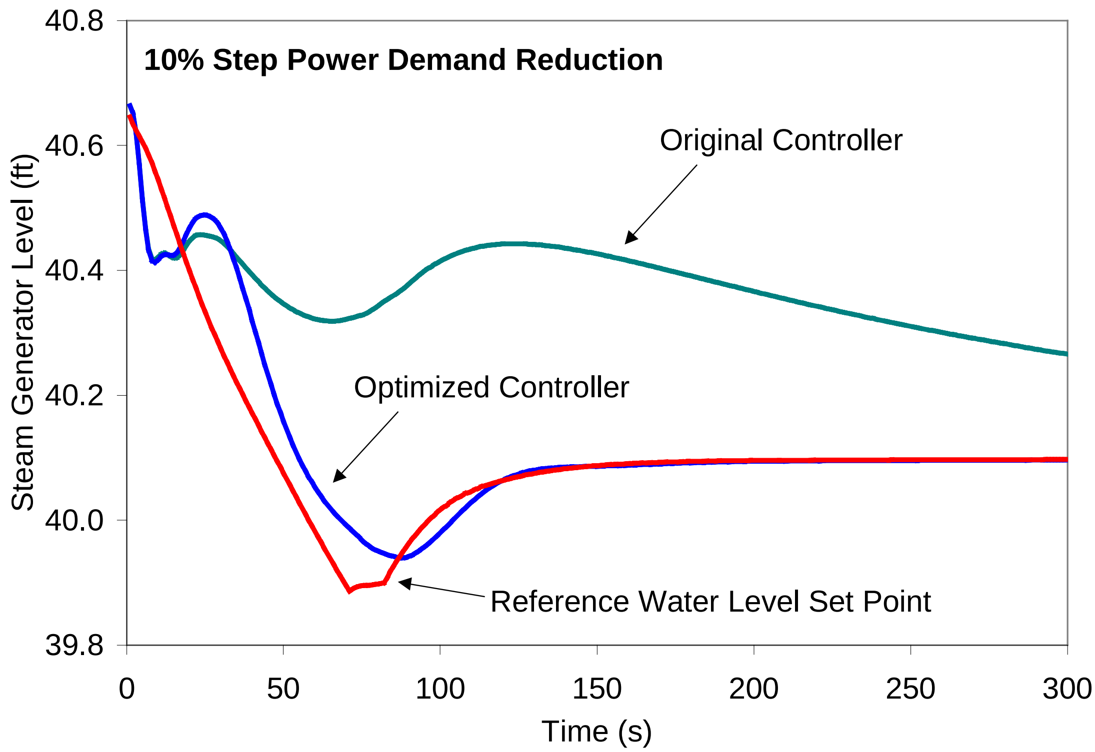

Fig. 1. The Control Engine automatically calculates the level control strategy that satisfy all of the performance requirements

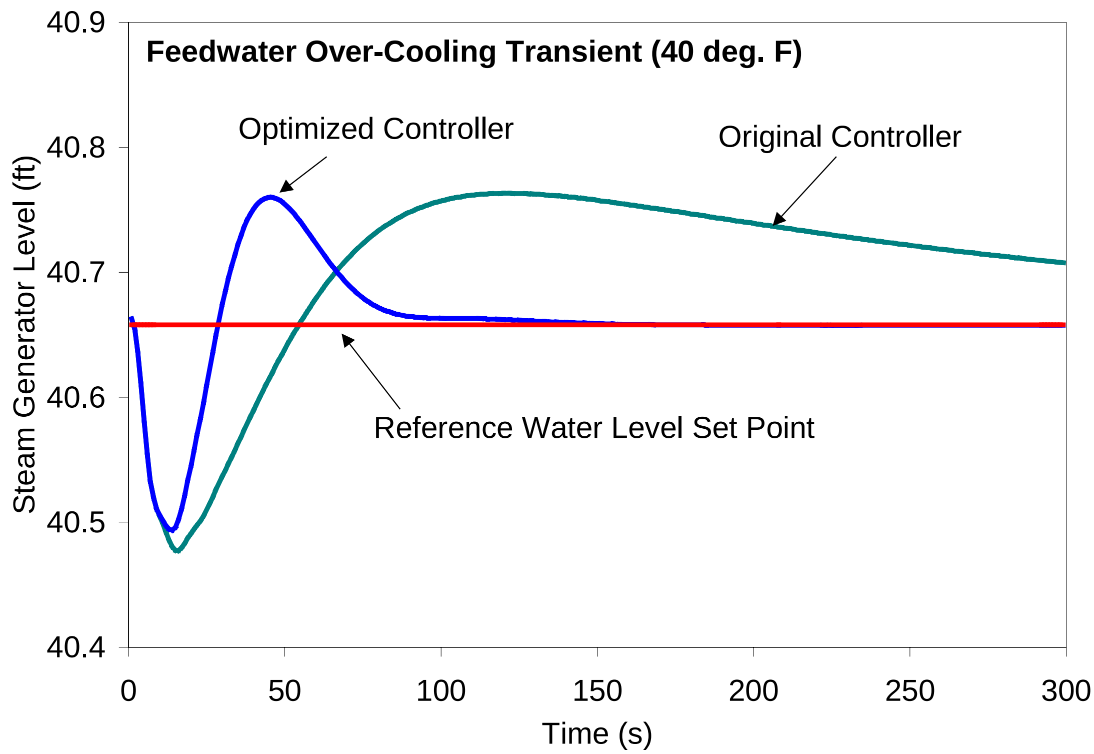

Fig 2. The Control Engine calculates the optimal control strategy for multiple postulated transients and performance requirements.