

**NERI PROJECT 99-119.
TASK 1. ADVANCED CONTROL TOOLS AND
METHODS. FINAL REPORT**

**J. March-Leuba
J. A. Mullens
R. T. Wood
Oak Ridge National Laboratory**

September 2002

**Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
Managed by
UT-Battelle, LLC
For
U.S. Department of Energy
Under contract DE-AC05-00OR22725**

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

TABLE OF CONTENTS

1.	Introduction.....	1
2.	Control Engine Concept.....	1
3.	Control Engine Implementation.....	2
3.1	Selection of Design Requirements Related to Control System Performance	2
3.2	Implementation of Requirements in Mathematical Form.....	3
3.3	Development of a Control Algorithm Library	3
3.4	Development and Validation of Plant Models	4
3.5	Automated Control Design Development.....	4
3.6	Development of Control Architectures	4
3.7	Control Design Implementation.....	4
3.8	Development of Diagnostics Methods to Update the Plant Model.....	5
4.	Application to a Simplified Steam Generator Model.....	5
4.1	Nomenclature for Simplified Steam Generator Model.....	8
4.2	Subscripts	8
5.	Application to a Full-Scope PWR Simulator	8
6.	Attachments	14
	• Attachment I. Task 1.1 Adaptive Inverse Control Algorithms	14
	• Attachment II. Task 1.2 Control-Priority Mode Selection Algorithm.....	14
	• Attachment III. Task 1.3 Controllers for Feedwater Systems	14
	• Attachment IV. Task 1.4 Requirements-Driven Control System Design.....	14

LIST OF FIGURES

Figure 1.	Schematic diagram of the automated control design process.....	2
Figure 2.	Schematic diagram of simplified steam generator model	5
Figure 3.	The control engine automatically determines the optimal control algorithm and parameters that satisfy the time response requirements.....	7
Figure 4.	Simulation of a degraded condition. Inlet flow is limited to 150%, 125%, or 115%. Control engine diagnoses the condition and adjusts control parameters.....	7
Figure 5.	The Control Engine automatically calculates the level control strategy that satisfy all of the performance requirements	9
Figure 6.	The Control Engine calculates the optimal control strategy for multiple postulated transients and performance requirements	10
Figure 7.	Power-Dependent Optimal Controller.....	11
Figure 8.	Example of Control Engine decision following a failed feedwater pump. Control parameters are re-optimized to minimize overshoot for a postulated power step transient	12
Figure 9.	Example of Control Engine decision after diagnosis of an incipient failure of the steam flow sensor. Control parameters are re-optimized to prevent instability.	13

1. INTRODUCTION

Nuclear plants of the 21st century will employ higher levels of automation and fault tolerance to increase availability, reduce accident risk, and lower operating costs. Key developments in control algorithms, fault diagnostics, fault tolerance, and communication in a distributed system are needed to implement the fully automated plant. Equally challenging will be integrating developments in separate information and control fields into a cohesive system, which collectively achieves the overall goals of improved performance, safety, reliability, maintainability, and cost-effectiveness. Under the Nuclear Energy Research Initiative (NERI), the U. S. Department of Energy is sponsoring a project to address some of the technical issues involved in meeting the long-range goal of 21st century reactor control systems. This project, "A New Paradigm for Automated Development Of Highly Reliable Control Architectures For Future Nuclear Plants," involves researchers from Oak Ridge National Laboratory, University of Tennessee, and North Carolina State University.

This paper documents a research effort to develop methods for automated generation of control systems that can be traced directly to the design requirements. Our final goal is to allow the designer to specify only high-level requirements and stress factors that the control system must survive (e.g. a list of transients, or a requirement to withstand a single failure.) To this end, the "control engine" automatically selects and validates control algorithms and parameters that are optimized to the current state of the plant, and that have been tested under the prescribed stress factors. The control engine then automatically generates the control software from validated algorithms.

Examples of stress factors that the control system must "survive" are: transient events (e.g., set-point changes, or expected occurrences such a load rejection,) and postulated component failures. These stress factors are specified by the designer and become a database of prescribed transients and component failures. The candidate control systems are tested, and their parameters optimized, for each of these stresses. Examples of high-level requirements are: response time less than xx seconds, or overshoot less than xx% ... etc. In mathematical terms, these types of requirements are defined as "constraints," and there are standard mathematical methods to minimize an objective function subject to constraints. Since, in principle, any control design that satisfies all the above constraints is acceptable, the designer must also select an objective function that describes the "goodness" of the control design. Examples of objective functions are: minimize the number or amount of control motions, minimize an energy balance... etc.

2. CONTROL ENGINE CONCEPT

The vision of this control-system design process is described schematically in Figure 1. Our final goal is to develop methods to implement reliable control systems that automatically satisfy all the design requirements. As indicated by the arrow in Figure 1, this is accomplished by automating the design process to arrive at a plant implementation directly from the design requirements.

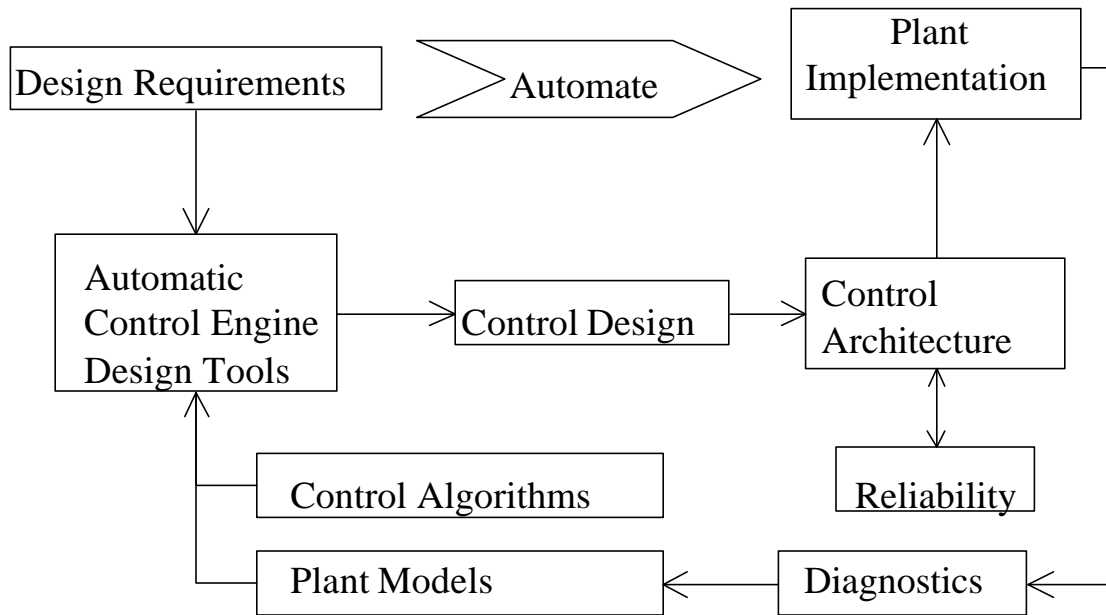


Figure 1. Schematic diagram of the automated control design process

The implementation of the methodology takes the bottom path of Fig. 1. For this implementation, the design requirements are fed into a control engine, which uses a library of control algorithms and validated plant models to arrive at the control design using an iterative optimization process. The control design is then implemented using validated control architectures, which are tested automatically to guarantee that the reliability requirements are met. Finally, during the lifetime of the plant, the plant model is maintained up to date (e.g, updated with component failures or mode changes) by an on-line diagnostics system.

3. CONTROL ENGINE IMPLEMENTATION

The automation of this design methodology requires the following steps:

3.1 Selection of Design Requirements Related to Control System Performance

The first step is to review the design requirements for applicability to the problem at hand. The automated design system will deal only with those design requirements that relate directly to the “performance” of the control system. The discarded requirements will have to be addressed separately in a manual fashion.

An example requirement is to specify that all the equipment must operate using a power supply of 110 Volts and 60 Hz. This is clearly a valid requirement, which must be specified when purchasing the control hardware, but is not directly related to control performance; thus, these types of requirements are not included in the automated control engine requirement set.

An example of a valid requirement is to specify that the control system must control the reactor without scram for a particular transient. For example, we could specify that a

feedwater controller should automatically handle a loss of a single heater without resulting in a reactor scram. Clearly, these types of requirements are directly related to control performance and must be included in the automated control engine requirement set.

The selection of “proper” design requirements is clearly one of the more important steps in the design process. It is also clear that some requirements will not be as clear cut as the examples above. In such cases, there must be an iterative process between the control engineers and the requirement “providers” to resolve differences. The clear advantage of our proposed automated control design process is that it frees the control engineers to concentrate on the high level goals and lets the computer do the tedious implementation work.

3.2 Implementation of Requirements in Mathematical Form

Once proper design requirements are identified, each must be documented in the form of a “subroutine.” This subroutine takes as input the results of a computer simulation and returns a value that indicates whether the particular requirement is satisfied. This documentation then becomes a mathematical description of the requirement, which is precise and reviewable. The set of subroutines that document all requirements then become a complete description of the control problem.

This step becomes in effect a second screening of the design requirements. If a design requirement cannot be written in the form a subroutine, it is probably not a proper control design requirement. The basic idea is that the requirement must be “testable.” In other words, given the results of a computer simulation, the control engineer must be able to determine unequivocally whether the requirement is satisfied or not; thus, a proper requirement should always be able to be documented in the form of a subroutine.

Clearly, to complete this second step, there must be an additional iterative process between the control engineers and the requirement “providers” until all requirements are documented in this mathematical form. This provides for yet another opportunity to review the completeness and appropriateness of the requirement set.

3.3 Development of a Control Algorithm Library

To automate the control system development, the control engine must have available a library of control algorithms. This library includes standard control algorithms such as proportional integral or advanced multivariate control schemes. The library may also include problem-specific algorithms; for example to control a steam-generator level, it would be logical to include a three-element controller in the library.

The control library can be as inclusive and sophisticated as desired. It may include problem-specific model-based controllers, neural networks, fuzzy controllers ... The only requirement is that those algorithms be parameterized; the control engine will select the optimal parameters for each algorithm and then pick the optimal algorithm so that the final design satisfies all requirements.

3.4 Development and Validation of Plant Models

Validated plant models are required to perform the computer simulations required for the automated control design. The plant models must be validated and must be kept up to date by the diagnostics system for the life of the plant.

3.5 Automated Control Design Development

The core of this research effort is the automated development of control designs that satisfy all the requirements. This is accomplished by using standard constrained minimization methods, where the requirements are the mathematical constraints, and the control engine minimizes an objective cost function that the designer specifies.

3.6 Development of Control Architectures

Our initial research addressing the control and information system architecture for future nuclear installations involves the development of functional requirements. The plant architecture that we envision can serve to establish an integration platform for functional capabilities and a distributed communications framework to support operations, maintenance, and engineering personnel at a 21st century nuclear power plant. The distributed network can provide the backbone to convey information from the data acquisition sources to the users and applications that process the information. The issues of architecture being addressed in this research are: (1) Provide a common, consistent interface to I&C systems; (2) Enable uniform, transparent access to distributed data sources; (3) Establish a computing environment that facilitates the integration of information and applications (e.g., diagnostics and control); (4) Define a system architecture that permits flexibility in implementation and expandability of functional capabilities; and (5) Define an approach to application support that lays the foundation for standardizing functions and interface conventions for the nuclear power industry.

Simply establishing network links among the various systems and installing workstations as network nodes addresses only part of the desired support for plant personnel at the of 21st century plant. A key goal of this research is to provide common functionality throughout the control and information system architecture. In this way, the efficiency of plant personnel in performing their tasks can be enhanced and the possibility of user error while interacting with multiple systems and data sources can be reduced. The proposed architecture concept is based on a layered approach with the capabilities and services of the application environment supporting the functionality of applications (e.g., diagnostics or controls).

3.7 Control Design Implementation

Once the automated control engine has identified a control design that meets all the requirements, the control design is implemented in the real-life plant using the validated control architectures.

3.8 Development of Diagnostics Methods to Update the Plant Model

Since the goal of this research is to maintain the design within its requirements for the life of the plant, it is imperative that the plant model be kept up to date. Thus, the implementation of these techniques requires the development of diagnostics methods that can identify component failures or degradations. Those component degradations should then be reflected in the plant model. The automated control engine is then implemented on-line with the updated models to guarantee that the original design requirements are still satisfied with the new plant configuration.

4. APPLICATION TO A SIMPLIFIED STEAM GENERATOR MODEL

As a demonstration of proof of principle, we have implemented this methodology using a standard off-the-shelf minimization algorithm. For this example, we have used the simplified steam generator model described schematically in Figure 2. This model is complex enough to be non-trivial, but simple enough to allow for a large number of fast calculations to define the proper strategies. The equations describing the dynamics of the steam generator are nonlinear:

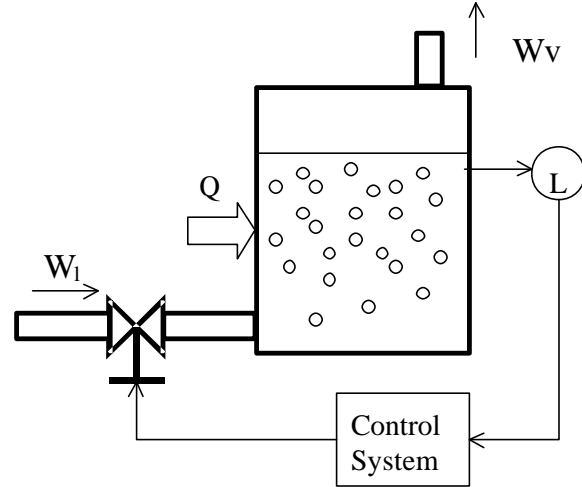


Figure 2. Schematic diagram of simplified steam generator model

$$\frac{dM}{dt} = W_l - W_v$$

$$\frac{dH}{dt} = Q + h_l W_l - h_v W_v$$

With the closure relations

$$\mathbf{a} = \frac{H - ALr_l h_l}{AL(\mathbf{r}_v h_v - \mathbf{r}_l h_l)} ; \quad \mathbf{r} = \mathbf{r}_l (1 - \mathbf{a}) + \mathbf{r}_v \mathbf{a} ; \quad L = \frac{M}{\mathbf{r}A} ; \quad W_v = \frac{W_{l0}}{\mathbf{a}_0} \mathbf{a}$$

For this proof-of-principle demonstration, the library of control algorithms includes only simple proportional or proportional-integral controllers. The control system can be described by the following equations

$$u = K_p (L_{set} - L) + K_i \int_0^t (L_{set} - L) dt$$

$$W_l = u \quad \text{if } 0 < u < u_{\max}$$

The maximum flow through the valve is limited by the full-open stem position, which introduces a hard nonlinearity. As design requirements, we selected the maximum 90% rise time for a step response, a maximum overshoot, and a maximum steady-state error. As objective cost function, we minimize the control effort, which is defined as

$$\text{Control Effort} = \int_0^t \left(\frac{du}{dt} \right)^2 dt$$

The control effort variable is proportional to the amount of work that the controller must perform. Typically, by minimizing this variable we obtain a minimum wear and tear of all control components and overall smoother performance.

The control engine immediately selected a proportional-integral controller with the optimized gain parameters that minimize the control actions while still satisfying the requirements. The control engine determined in all cases that a simple proportional control algorithm would not be able to satisfy the steady-state error requirement without an integral-control component, as expected.

Figure 3 shows an example response of the system for three different requirements. In all three cases, we have requested that the engine maintain an overshoot less than 2% for a 10% step demand in the level setpoint. The requirements are changed from case to case so that the response time is 5, 3, or 2 seconds respectively. The control engine automatically computes the optimal controller gains to satisfy those requirements and minimize the control effort. As it can be seen in Figure 3, the faster the system, the larger the control effort required.

Since the simplified steam generator model that we used is non-linear, it comes as no surprise that the optimized gain parameters are function of the operating conditions and Plant State. This feature has been used to demonstrate the adaptive nature of our control-design methodology. By running the control-engine and the diagnostic system in parallel with the real system, we are able to feed it the current operating conditions and the status of the failed components. Thus, the control-engine is able to determine on-line whether the original design requirements are met under the current operating conditions and Plant State. If the design requirements are not met, alternate control algorithms or parameters that meet those requirements are calculated. The alternate algorithms can be implemented on-line, or suggested to an operator, who makes the final decision about implementation.

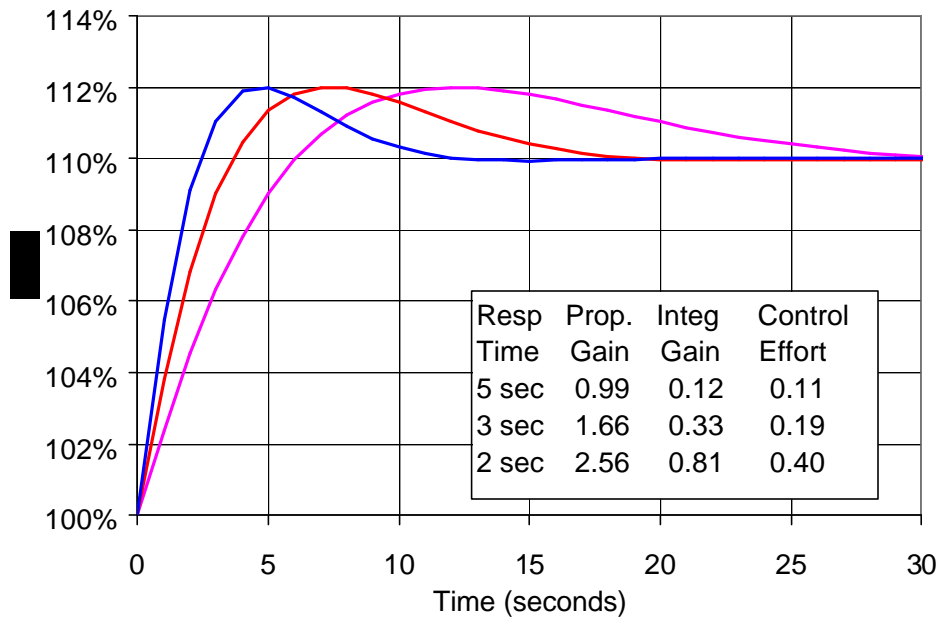


Figure 3. The control engine automatically determines the optimal control algorithm and parameters that satisfy the time response requirements.

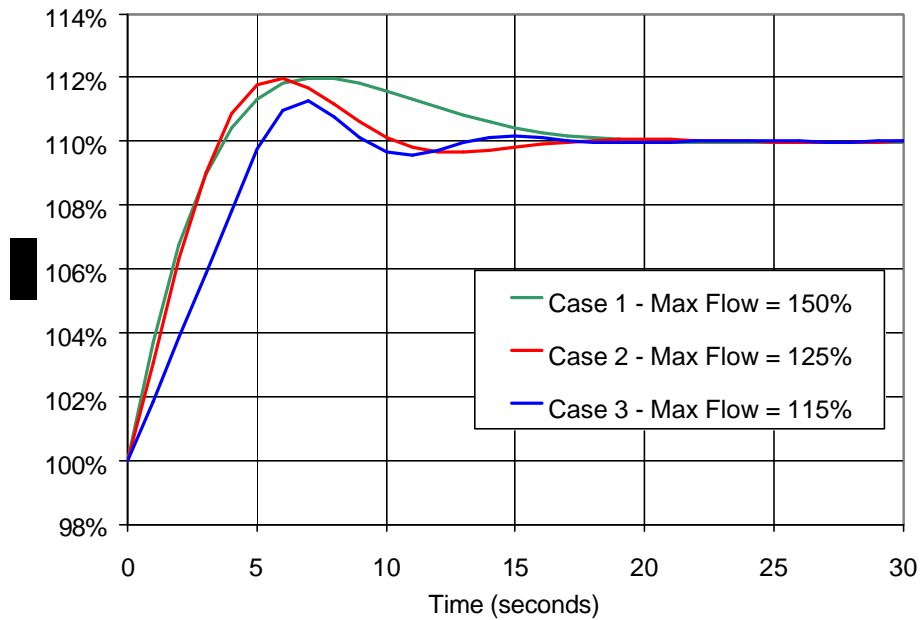


Figure 4. Simulation of a degraded condition. Inlet flow is limited to 150%, 125%, or 115%. Control engine diagnoses the condition and adjusts control parameters.

Figure 4 presents an example of the use of the control engine methodology in an adaptive form during the life of the plant. For this example, we have assumed that some time after implementation, the available flow through the input valve is limited. An example of this limitation would be caused by cavitation on the valve, which would impose a new requirement to limit the flow. Another example of limited flow would be the loss of one of the upstream feed pumps. For the example in Fig. 4, we have assumed three cases, the original, a case where the maximum inlet flow is limited to 125% of nominal, and a third case where the maximum inlet flow is limited to 115%. For all these cases, we require an overshoot lower than 2% and a time response lower than 3 seconds.

The control engine is assumed to be running in diagnostics mode. For case 2 (125% maximum flow,) the control engine detects that the original time-response requirements are not met, and it calculates new optimal parameters that satisfy the requirements. For case 3 (115% maximum flow,) the control engine determines that the requirements cannot be satisfied because the minimum response time achievable is ~4 seconds. Even though the control system is capable of maintaining the level at 100%, it cannot satisfy the requirements under the specified 10% step response transient. For this case, the operator would be notified and there are two options: either (1) the requirements would need to be relaxed, or (2) the plant would need to be fixed to allow for higher inlet flow.

4.1 Nomenclature for Simplified Steam Generator Model

M	total mass
L	two-phase level
A	flow area
ρ	density
W	mass flow
α	void fraction
h	specific enthalpy
K	controller gain

4.2 Subscripts

l	liquid
v	vapor
max	maximum
set	setpoint
p	proportional
i	integral
0	initial condition

5. APPLICATION TO A FULL-SCOPE PWR SIMULATOR

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 developed in Task 3.1. 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 Figure 5 and Figure 6; 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 unusual 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

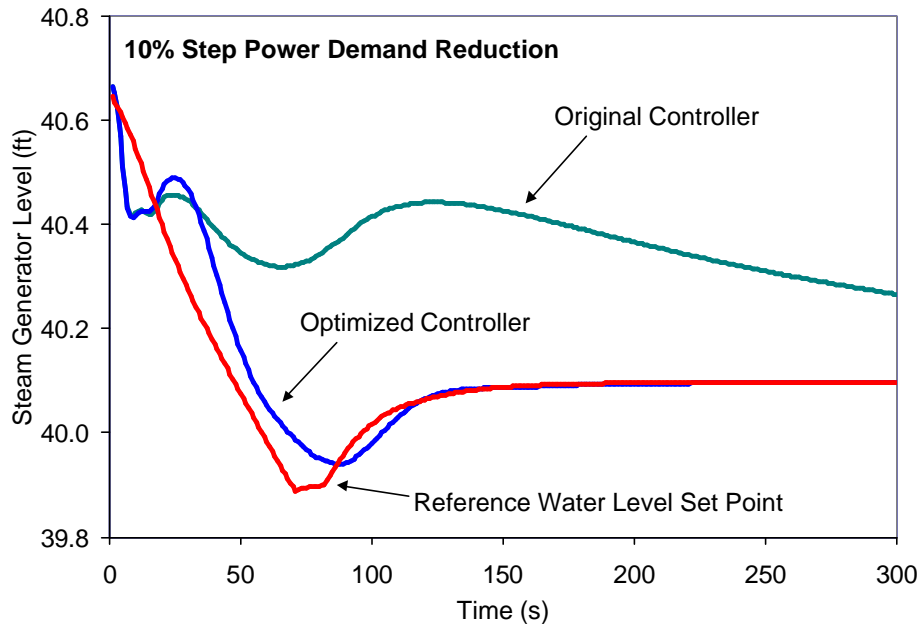


Figure 5. The Control Engine automatically calculates the level control strategy that satisfy all of the performance requirements

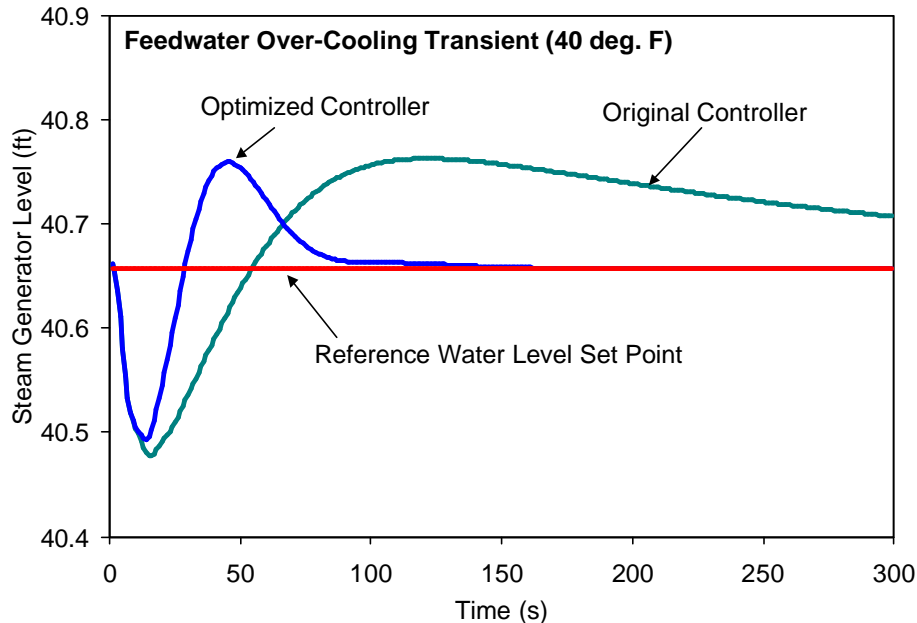


Figure 6. The Control Engine calculates the optimal control strategy for multiple postulated transients and performance requirements

Using a procedure similar to the one outlined above, we can automatically develop an optimized controller for our U-tube steam generator. Figure 7 shows one such controller. This is a typical three-element controller, where the steam generator level is controlled by setting up a feedback loop on the measured water level plus the mismatch between the steam flow and feedwater flows. By adding this flow mismatch in the three-element logic, these types of controllers can handle the well-known shrink and swell phenomenon, where the two-phase water level temporarily drops when additional cold feedwater is injected because it collapses steam voids.

Steam generator dynamics are non-linear and change significantly with operating power level as the void fraction changes. To compensate for those changes, the controller shown in Figure 7 was designed with power-dependent gains. In this way, the controller can be designed to be significantly more “aggressive” and have better performance at full power without losing stability at low power levels. As seen in Figure 7, our controller design provides a tradeoff between proportional, integral, and three-element gains as power changes. This results in greatly improved performance, especially at high powers where the steam generator operates most of the time.

What makes this controller unique, however, is not its performance characteristics, but the fact that it was designed completely automatically using the design engine of Task 1.4. In essence, we have captured the design requirements for this steam generator in the control engine and a new optimized controller can be calculated at any time in the life of the facility if conditions or equipment changes (e.g., component failures or replacements) occur.

$$\Delta FCV = k_p(P) \times \left[(L - L_{set}) + k_3(P) \times (w_l - w_s) \right] + k_i(P) \times \int (L - L_{set}) dt$$

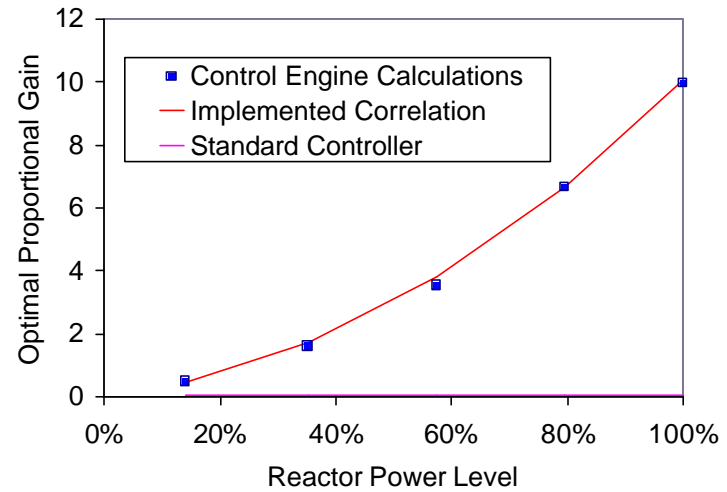
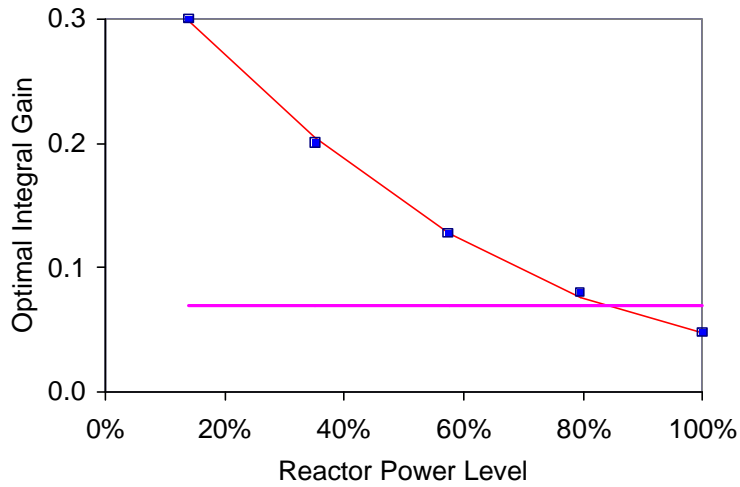
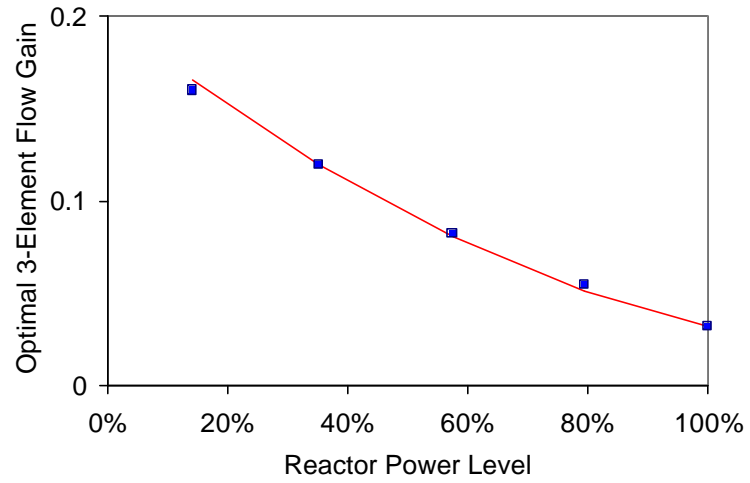


Figure 7. Power-Dependent Optimal Controller

Figure 8 and Figure 9 show examples of this feature. In the example of Figure 8, a feedwater pump is assumed failed, so that the maximum feedwater flow available to the controller is limited. The Control Engine automatically recognizes that, because of the maximum flow limitation, the original controller will produce a significant overshoot. The Control engine automatically suggests a new set of control parameters that are less aggressive and minimize the overshoot caused by the new plant condition. In the example of Figure 9, the diagnostics system has detected an incipient failure of a steam flow sensor. Then Control engine, then, evaluates the impact of this failure (by assuming completely failed) and concludes that if the sensor fails, the controller will become unstable. It then, automatically, re-computes optimized control parameters that do not use the steam flow sensor, and it suggests a re-optimized controller that will not be sensitive to the sensor failure should it occur. Note that this new controller is not as good as the original controller, but it is more fault-tolerant given the information we now know about the probability of failure of the sensor.

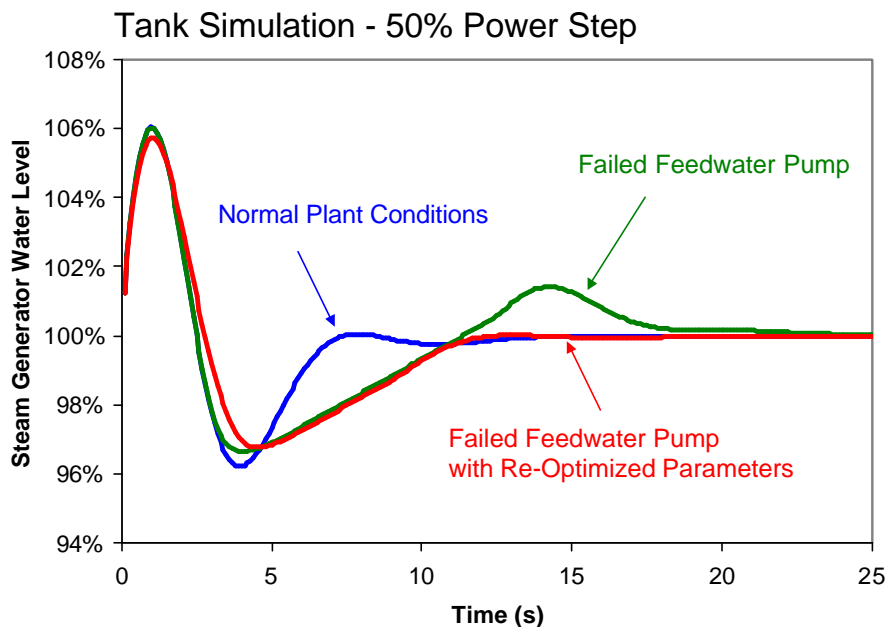


Figure 8. Example of Control Engine decision following a failed feedwater pump. Control parameters are re-optimized to minimize overshoot for a postulated power step transient

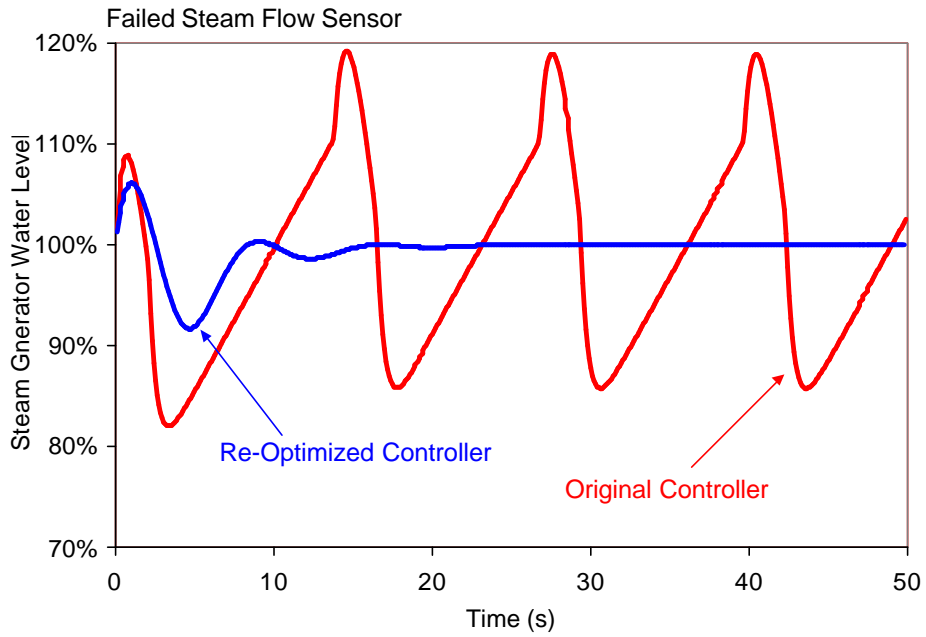


Figure 9. Example of Control Engine decision after diagnosis of an incipient failure of the steam flow sensor. Control parameters are re-optimized to prevent instability

6. ATTACHMENTS

The following attachments are included

- **Attachment I. Task 1.1 Adaptive Inverse Control Algorithms**
- **Attachment II. Task 1.2 Control-Priority Mode Selection Algorithm**
- **Attachment III. Task 1.3 Controllers for Feedwater Systems**
- **Attachment IV. Task 1.4 Requirements-Driven Control System Design**

Attachment I.

Task 1.1 Adaptive Inverse Control Algorithms

1. Introduction

Imperfections of system components, especially those of actuators and sensors, are among the factors that severely limit the performance of feedback control loops. Most often, a critical imperfection is a nonlinearity which poorly known, increases with wear and tear, and varies from component to component. Some of the most common imperfections found to be an integral part of the component design and are classified as dead-zone, backlash, and hysteresis. These are typically found in a wide variety of components and systems such as mechanical, hydraulic, pneumatic, magnetic, and piezoelectric.

The problem of nonlinearity is frequently encountered in common control engineering practices. The behavior of nonlinear elements tends to degrade the performance in feedback control loops. In addition, most of feedback control methods available in literature are designed for linear systems and cannot cope with the nonlinear characteristics of the system being controlled (plant). The complexity of nonlinear systems can be attributed primarily to nonlinear dynamics described in terms of partial differential equations with rather distributed parameters and most often they are time variant. Solving partial differential equations is quite difficult with high degree of complexity, and for that reason very few mathematical design and analysis tools can be directly applicable to nonlinear systems are in existence. The most common approach to design feedback control for nonlinear systems is to transform the nonlinear system into an equivalent linear system. This is commonly realized by approximating the nonlinear dynamics with a linear model around some operating point of interest; in which case all control design and analysis tools will apply. A careful measures need to be exercised to ensure that the applied control law is bounded enough to maintain the system performance in the linear region.

Another approach for developing control methodologies of plants described by nonlinear dynamics is the piecewise linearization technique. In this approach, a piecewise linearization method is applied to the nonlinear system through which approximation by linear segments along an operating trajectory is considered. Linearization techniques are typically performed on mathematical models representing the system dynamics, and as a result various errors are introduced causing the derived models to be inaccurate to a varying degrees. Among those errors is the error resulting from the model derivation process of a complex physical system, uncertainties in plant disturbances, and un-modeled dynamics.

This report documents possible technical approaches that can be used to address the subject of nonlinearity with emphasis on adaptive inverse control (AIC) as first step in designing robust feedback control system using linear methods. This nonlinear cancellation method is described in [12]. It is used as a linearization technique to facilitate the application of established linear control techniques for plants with nonlinear elements. The AIC methodology focuses primarily on nonlinearities of sensors and actuators that are typically used in industrial applications. The discussions presented in the following section include topics on linear feedback control methods such as multivariable control and robust control methods. More specifically, LQG/LTR and H-infinity are discussed as robust control techniques

Application of control methods in plants has been significantly improved in the last two decades because of advancements in the digital computational capabilities and the availability of many analytical design and analysis tools. Several software packages are commercially available with wide range of capabilities ranging form algorithms implementing classical control methods to complex algorithms implementing the most advanced control strategies.

2. Background

2a. Nonlinearity

Nonlinearities in control systems are frequently encountered due to the nature of most physical systems in which they are either inherited or are intentionally introduced. The inherited type of nonlinearities is often attributed to dead zone (Figure 2.1-a), hysteresis (Figure 2.1-b), saturation conditions (Figure 2.1-c), Coulomb friction (Figure 2.1-d), backlash (Figure 2.1-e), parameter aging, and various nonlinear relations between system variables. Dead-zone, hysteresis, and backlash are identified as the forms of nonlinearity that are typically associated with actuator and sensor devices. These devices are essential components in any control application and therefore, good understanding of their nonlinear behavior is essential to the design of control systems. Researchers over the last two decades have investigated these forms of nonlinearity in an effort to understand their characteristics and develop appropriate mathematical models that can be used in deriving various control strategies [8], [19-22], [35]. Numerous mathematical models have been developed, especially for backlash and hysteresis.

The intentionally introduced nonlinearities, on the other hand, are usually considered for specific purpose such as improving systems performance or to favorably influence economy in component selection.

One of the main difficulties in dealing with nonlinear behavior of systems is that the response of these systems does not follow the superposition principle ($f(a x + b y) = a f(x) + b f(y)$, where a, b are any constant scalar quantities) typically exhibited by linear systems behavior. Consequently, all mathematical tools used for design and analysis of linear systems cannot be directly applied to nonlinear systems. Among the characteristics of nonlinear behavior is the self-excited oscillation, called the *limit cycle*. This oscillatory response can be simply investigated by the assumption that the output of a nonlinear element for a sinusoidal input is a sinusoidal component at the input frequency.

Control system design for nonlinear systems is traditionally accomplished in the frequency domain using the describing-function method [1], [23]. The describing-function is defined as the input-output ratio of a nonlinear element in the frequency domain, just as the transfer function defines the input-output relation for a linear element in the frequency domain. More accurately stated, the describing function is defined as the ratio of the fundamental frequency components, of the response of a nonlinear system to a sinusoidal excitation, to the amplitude of the excitation as shown by Figure 2.2. In most feedback control problems, linear and nonlinear components are usually co-exist within the same feedback loop as shown in Figure 2.3. Describing-function approach, however, is not practical to apply because of the complex computational and graphical representation involved. Furthermore, this approach cannot be used when optimality and adaptability of the control strategy are desired.

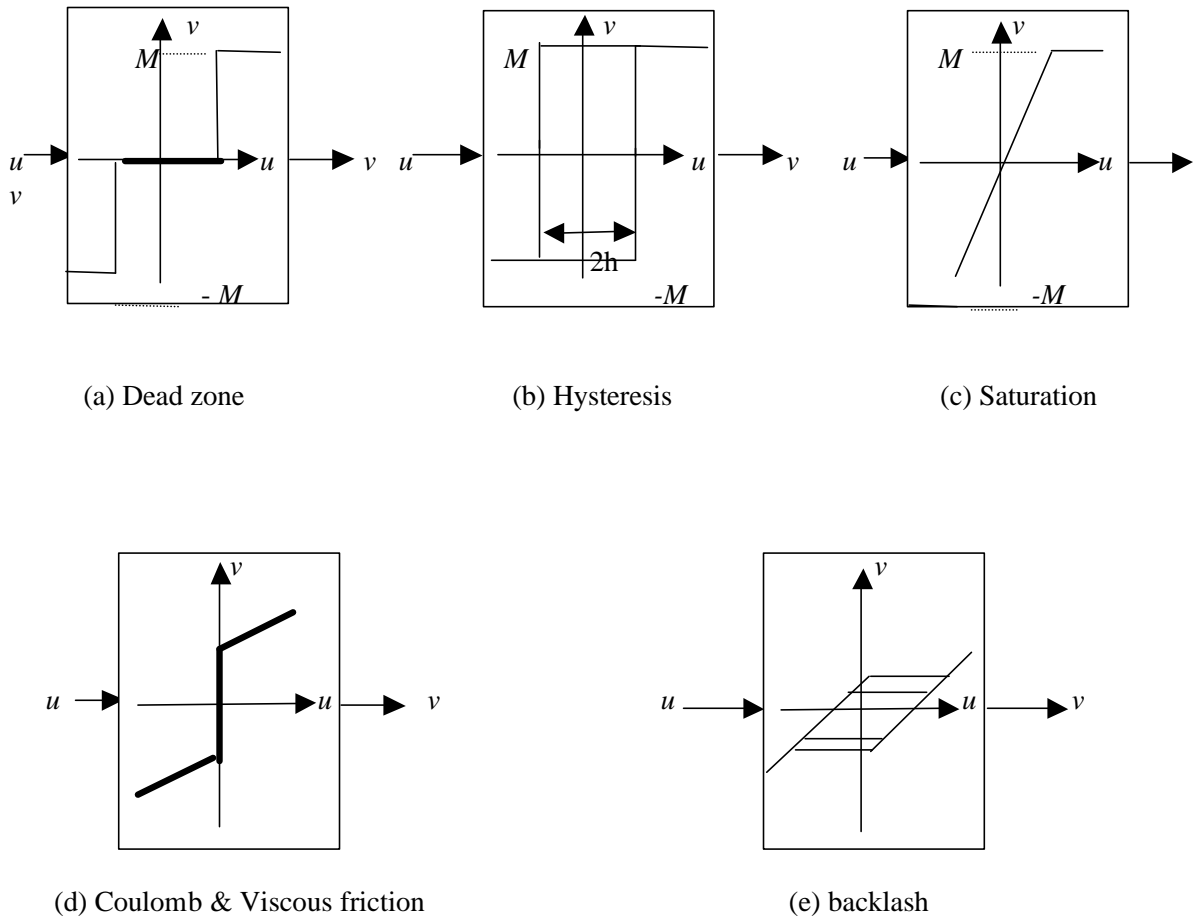


Figure 2.1 Example of nonlinear elements

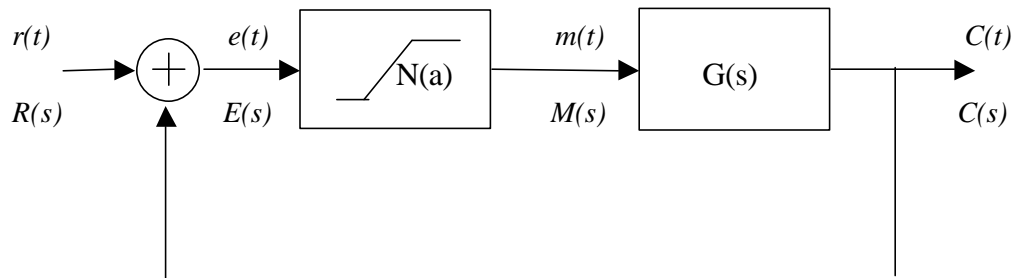


Figure 2.3 Feedback control system where r is a reference and the error $e(t) = r(t) - C(t)$, $N(a)$ is the describing-function of the nonlinear element, and $G(s)$ is the transfer function of the linear element

Linearization techniques are commonly used to analyze the behavior and to study the stability of nonlinear control systems around some operating point or range, by taking advantage of linear time- and frequency- domain methods (transfer function approach, frequency-domain analysis, frequency-domain stability criteria, state-space representation).

The level of complexity of control system design increases depending on the number of inputs and outputs of the plant. Single input single output systems (SISOs) are the most simpler to analyze because interaction among variables doesn't exist. Multivariable systems with more than one input and more than one output (MIMO), on the hand, require special techniques capable of handling interactions between controlled variables.

2b. Multivariable Control

An essential first step in the design of a control system is the derivation of mathematical description to model the physical system to be controlled. It is also customary in control engineering practices to represent a system and all of its associated components by a block diagram indicating the interaction interrelationships between the variables of interest as well as representing each component by its corresponding input-output relationship either in the frequency domain or in the time domain. This convenient means of system representation is applicable to both linear and nonlinear systems.

For SISO systems the concept of transfer function plays an important role in the characterization of linear time-invariant (LTI) systems. In general, the block diagram representation and signal flow graphs along with the transfer function form the basis of representing the input-output relationships of LTI system in control systems theory.

The transfer function of linear system between an input $r(t)$ and an output $c(t)$ is commonly denoted by $G(s)$, and defined as the Laplace transform of the impulse response $g(t)$. The basic mathematical definition of transfer function as depicted by the basic block diagram of Figure 2.4 is written as the ratio

$$C(s) / R(s) = G(s) / (1 + G(s) H(s)) \quad (2.1)$$

and the error

$$E(s) = R(s) - H(s) C(s) \quad (2.2)$$

For SISO $C(s)$, $R(s)$, $E(s)$ in equations (2.1) and (2.2) are scalar quantities.

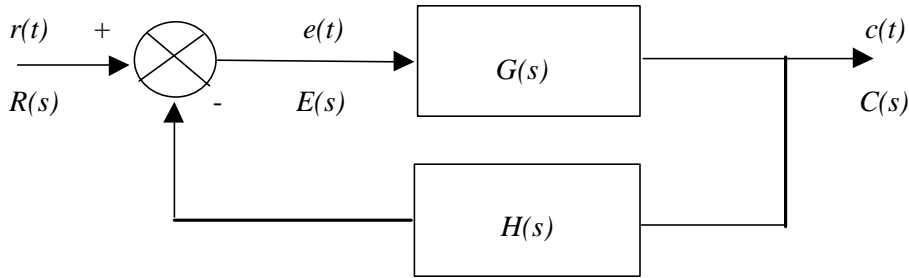


Figure 2.4 Basic block diagram of a feedback control system

In multivariable control problems, the control system usually has multiple inputs and outputs (MIMO). The concept of transfer function for SISO can be extended to the multivariable case with the exception that the scalar variables become vectors and the transfer function become transfer matrix as indicated by Figure 2.5.

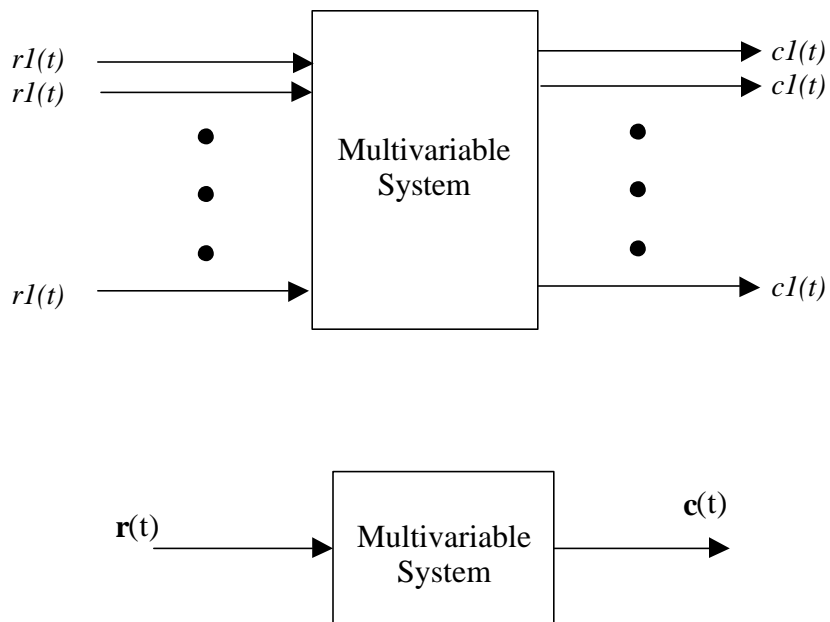


Figure 2.5 Block diagram representations of a multivariable system.

Where,

$\mathbf{r}(t)$ is (px1) input vector $[r_1(t), r_2(t), \dots, r_p(t)]^T$, and $\mathbf{c}(t)$ is (qx1) output vector $[c_1(t), c_2(t), \dots, c_q(t)]^T$.

The transfer function relationship between the input and the output of the multivariable system can then be expressed as

$$\mathbf{C}(s) = \mathbf{G}(s) \mathbf{R}(s) \quad (2.3)$$

Where

$\mathbf{C}(s)$, $\mathbf{G}(s)$, $\mathbf{R}(s)$ are all matrices defined as

$$\mathbf{C}(s) = \begin{bmatrix} C_1(s) \\ C_2(s) \\ \vdots \\ C_q(s) \end{bmatrix}, \quad \mathbf{R}(s) = \begin{bmatrix} R_1(s) \\ R_2(s) \\ \vdots \\ R_p(s) \end{bmatrix}, \quad \mathbf{G}(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \dots & G_{1p}(s) \\ G_{21}(s) & G_{22}(s) & \dots & G_{2p}(s) \\ \vdots & \vdots & \ddots & \vdots \\ G_{q1}(s) & G_{q2}(s) & \dots & G_{qp}(s) \end{bmatrix}$$

and the closed loop transfer matrix is given by,

$$\mathbf{G}_{cl}(s) = [\mathbf{I} + \mathbf{G}(s) \mathbf{H}(s)]^{-1} \mathbf{G}(s) \quad (2.5)$$

Where \mathbf{I} is the identity matrix. In this case, it should be noted that the closed loop transfer function cannot be expressed as a ratio $\mathbf{C}(s)/\mathbf{R}(s)$ since $\mathbf{C}(s)$ and $\mathbf{R}(s)$ are matrices.

For LTI systems with transfer matrices $\mathbf{G}(s)$ and $\mathbf{H}(s)$, the matrix function

$$\mathbf{J}(s) = \mathbf{I} + \mathbf{G}(s) \mathbf{H}(s) \quad (2.6)$$

is known as the return difference matrix, and the matrix function

$$\mathbf{L}(s) = \mathbf{G}(s) \mathbf{H}(s) \quad (2.7)$$

is known as the loop gain matrix.

Feedback control loops in multivariable systems are usually coupled to each other. In cases where cross coupling or interaction is weak relative to the desired control

performance, design methods for SISO systems may apply. This can be accomplished by decoupling the system in physical sense and treat each loop independently as separate SISO system. The overall output response of the system can then be equivalent to the sum of the output responses for each input as long as the system is linear where superposition principle applies. If, on the other hand, the interaction is strong enough to be dominant, the multivariable system must be considered as one entity. One of the design methods of multivariable feedback control system with good robustness properties is the linear quadratic Gaussian (LQG) with loop transfer recovery (LQG/LTR). This design method allows the treatment of the multivariable system as one entity by taking into consideration the interaction between the various loops as it's the case with practically most of physical systems and processes.

2c. Robust Control

One important aspect of control engineering is the robustness property of feedback control system. Robustness is defined as the ability of the control system (or the controller) to maintain overall system performance despite of variations in plant physical parameters, operating conditions and other uncertainties. A Robust feedback control system has reasonably large stability margin, good disturbance attenuation, and low sensitivity to parameter variations. The stability margin of a control system is measured in terms of the overall gain margin and phase margin, both of which determine the degree of stability of the system under varying operating conditions.

The problem of robust design is an important consideration since all physical systems (plants) to be controlled will definitely experience variations over time due to changes in environmental and operating conditions (wear due to aging, errors due to calibration, installation, human factors). Other sources of uncertainty are due to modeling error resulting from describing complex plants and processes by mathematical expressions representing their dynamics, linearization and approximations methods used to simplify the analysis, and unmodeled dynamics.

The design of linear controllers for plants with strong nonlinearities by linear control design methods often leads to unsatisfactory control performance. Furthermore the model of the plant is mostly not well known because of the uncertainties of the model parameters. For example, complex reaction processes or nonlinear valve characteristics cause strong nonlinearities and uncertainties of parameters. Robust control design methods take into account these nonlinearities and uncertainties to guarantee robust stability and robust performance.

Classical feedback control methods developed mainly for SISO, such as Nyquist, root-locus, Bode, and Nichols plots, are all frequency domain techniques designed to provide robustness through the feedback loop. Robust control theory has advanced in the last two decades where many techniques appeared in the literature and found their way into the real world. Among the most recently developed, are μ -synthesis, covariance control, convex optimization, H_∞ , and the latest development of linear matrix inequalities (LMIs) [9-10], [27].

2d. Linear Quadratic Gaussian (LQG)/ Loop Transfer Recovery (LTR) design Method

The LQG/LTR is an optimal control method that can be applied to linear time-invariant SISO and MIMO systems to provide feedback control system performance having good robustness properties such as reasonably large stability margin, good disturbance attenuation, and low sensitivity to parameter variations. In addition to these good robustness properties, this design method has additional capability of minimizing the effects of unmodeled high-frequency dynamics, neglected nonlinearities, and reduced-order model.

The LQG structure consists basically of a Kalman filter state estimator and a linear quadratic regulator (LQR) as shown by Figure 2.6. The plant disturbances and measurement noise are represented by two uncorrelated white noise processes $w(k)$ and $v(k)$ respectively with normal (Gaussian) distribution and zero mean. The Kalman filter estimator, by design, has been shown to have good robustness properties for plant perturbations at the plant output whereas the LQR has been shown to have good robustness properties for perturbations at the plant input. In spite of the fact that the individual components of the LQG has good robustness properties, the LQG controller collectively as a structure would not guarantee robustness properties at either the plant input or the plant output [14].

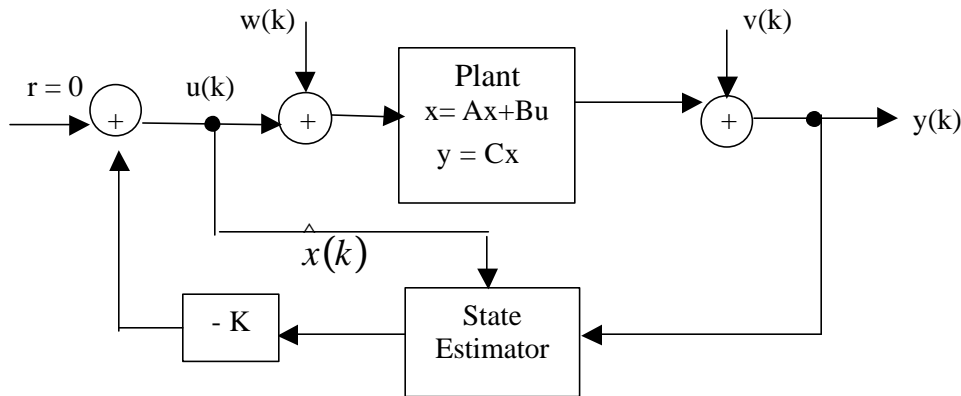


Figure 2.6 Simplified LQG block diagram.

In the linear quadratic regulator problem, the plant is represented by a linear state space model either as time-variant or as time-invariant, depending on the dynamic behavior of the plant. Considering the case of discrete linear time-invariant form of the plant, the state space model is usually expressed by state difference equation and output (observation) equation as

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \quad \text{where } k = 0, 1, 2, \dots, N \quad (2.8)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (2.9)$$

The objective in designing optimal control system is to determine a control law $\mathbf{u}(t)$ that is optimal in some sense in order to obtain a desired plant performance. In the LQR problem, the control input to the plant takes the optimal form

$$\mathbf{u}(k) = -\mathbf{K}\mathbf{x}(k) \quad (2.10)$$

where \mathbf{K} is an optimal gain schedule that needs to be calculated based on minimizing a quadratic cost function in the form

$$J_N = \sum_{k=0}^N \mathbf{x}^T(k)\mathbf{Q}(k)\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}(k)\mathbf{u}(k) \quad (2.11)$$

where N is finite and $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ are positive definite and positive semi definite weighting matrices respectively. Both matrices are usually selected to be symmetric to simplify the computation without any loss of generality. The quadratic cost function (2.11) is considered for the simplicity in its development and computations as well as the logical sense it offers, through the flexibility in selecting \mathbf{Q} and \mathbf{R} , to obtain the desired performance. Minimizing the first term of (2.11) ensure that the states will be driven to the zero state as quickly as possible, where as minimizing the second term will ensure that the control \mathbf{u} will not be too large for all practical purposes.

There are several approaches to minimize the cost function J_N , one of which is the principle of optimality developed by Richard Bellman [7], [25] and another similar approach is the minimum principle. Both of these approaches are discussed in details in [7].

The LQG design when it is considered in conjunction with the LTR procedure serves to recover robustness at either the input or the output of the plant. If robustness is desired at the input of the plant, the first consideration is to design a nominal robust LQR satisfying the design constrains. Consequently, an LTR design procedure should be incorporated by designing a Kalman filter gain that recovers the robustness at the input to the plant to approximate that of the nominal LQR design. If, however, robustness is desired at the output of the plant, a nominal robust Kalman filter design is made to satisfy the performance constraints. Consequently, an LTR design procedure is used to design an LQR gain that recovers robustness at the output of the plant to approximate that of the nominal Kalman filter design.

Software packages are commercially available and are designed to facilitate the control design process by providing users with systematic implementation of the control method. Some of the software equipped with complex algorithms based on the most advanced control strategies, such as, LQG/LTR, H-infinity, μ -synthesis, etc. One of the most widely used software package in industry, academia, and scientific fields is MATLABTM. It consists of individual toolboxes capable of implementing control system design based on both classical and advanced control theories as well as statistical analysis methods. Figure 2.7 illustrates how the LQG problem can be solved using the MATLAB platform.

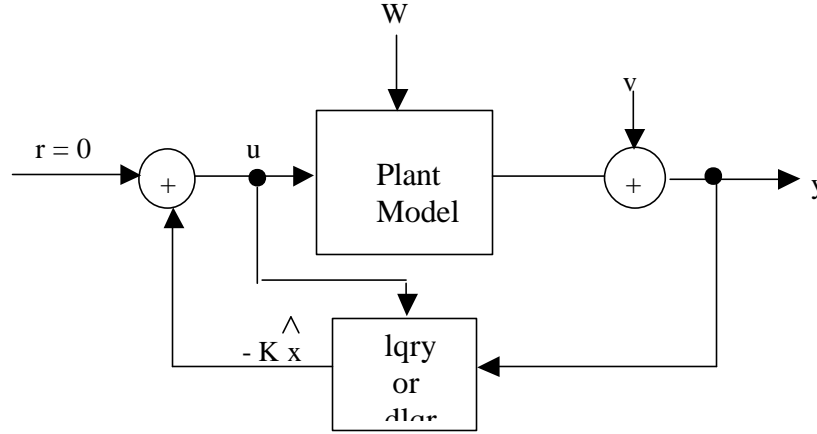


Figure 2.7. Block diagram illustrating solving the LQG problem by MATLAB.

One of MATLAB toolboxes is the *Control System Toolbox* which has two call functions dedicated to implement LQG design procedure in both continuous- and discrete-time domains (lqry, dlqr). In the discrete form, the plant is described by the discrete state space representation (state difference equation and the discrete output equation). The function dlqr solve the discrete-time, linear-quadratic regulator problem and its associated discrete Riccati equation (DRE).

Given the plant state space equations (2.8) & (2.9), the Syntax

$$[K, S, e] = dlqr (A, B, Q, R, N),$$

calculates the optimal gain matrix K such that the state-feedback law $u(k) = -K \mathbf{x}(k)$ minimizes the quadratic cost function

$$(2.12) \quad J = \int_{k=1}^{\infty} \mathbf{x}^T(k) \mathbf{Q}(k) \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{R}(k) \mathbf{u}(k) + 2 \mathbf{x}^T(k) \mathbf{N} \mathbf{u}(k)$$

The default value $N = 0$ is assumed when N is omitted.

In addition to the state-feedback gain K , dlqr returns the solution S of the associated DRE

$$\mathbf{A}^T \mathbf{S} \mathbf{A} - \mathbf{S} - (\mathbf{A}^T \mathbf{S} \mathbf{B} + \mathbf{N}) (\mathbf{B}^T \mathbf{S} \mathbf{B} + \mathbf{R})^{-1} (\mathbf{B}^T \mathbf{S} \mathbf{A} + \mathbf{N}^T) + \mathbf{Q} = 0, \quad (2.13)$$

and the closed-loop eigenvalues $e = \text{eig}(\mathbf{A} - \mathbf{B} * \mathbf{K})$.

The continuous-time function lqr provides similar solution when the plant is described by its continuous-time state-space form

$$\mathbf{x}' = \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u} \quad (2.14a)$$

$$\mathbf{y} = \mathbf{C} \mathbf{x} + \mathbf{D} \mathbf{u} \quad (2.14b)$$

The continuous-time solution to the state-feedback law $\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t)$ minimizes the quadratic cost function

$$J = \int_0^{\infty} (\mathbf{y}^T \mathbf{Q} \mathbf{y} + \mathbf{u}^T \mathbf{R} \mathbf{u} + 2 \mathbf{y}^T \mathbf{N} \mathbf{u}) dt \quad (2.15)$$

2e. H-infinity control design concept

H-infinity (H_{∞}) is a feedback control system design and synthesis method used to optimize the performance and robustness of MIMO control systems. In comparison with other optimal control methods such as the LQG, this method optimizes both performance and robustness whereas the LQG alone without the LTR procedure optimizes only performance but not robustness. In this method robustness objectives are stated in terms of singular value inequalities.

One of the major research efforts in control theory during the past decade has been the development of the "H-infinity optimal control theory". It fundamentally addresses issues of worst-case controller design for linear plants subject to unknown disturbances and plant uncertainties, including problems of disturbance attenuation, model matching, and tracking [12], [28]. The H_{∞} is the Hardy space which is defined as the space of all complex-valued functions of a complex variable s that are analytic and bounded in the open right-half complex plan, $\text{Re } s > 0$.

The basic design problem using this technique can be stated as: how to design a controller that minimizes the H_{∞} -norm of a given closed-loop transfer matrix. Where the H_{∞} -norm of a transfer matrix is defined as the maximum of its largest singular value over all frequencies. The problem statement can be restated in a more explicit formulation as: given a nominal LTI description of the control plant together with singular values bounds on an appropriate uncertainty model and on the performance objectives, design a fixed-parameter LTI controller that meets at least the nominal performance requirements and that achieves robust stability. The optimization part of the development process offers some flexibility in investigating the inherent trade-offs between performance and robust stability. Although the H_{∞} design process vary from application to application, a generalized framework has been developed as guidance to the synthesis process. Discussions on the generalized framework referred to in the literature as the *Standard Problem* are beyond the scope of this report. Simplified treatment of the standard problem and H_{∞} concepts can be found in [5].

H_{∞} appears as a natural and powerful method to control systems with unstructured uncertainties. However, its results are largely over-dimensioned when applied to systems with structured perturbations. This is due largely to lack of detailed information about the origin and influences of the uncertainties. In this case, the μ -synthesis (structured singular value) provides more adequate results.

Synthesis of an H_{∞} controller can be considered as a two-step design process. In the first step, the plant is augmented with weighting functions, which reflects the desired specifications for control system. The weighting functions are usually selected in order to force a controller design that meets the performance requirements and robustness to high frequency unmodeled dynamics. In

the second step, the controller is constructed for the augmented plant. The mathematical solution of H_8 synthesis has been well treated and successfully explored for the general case since the late eighties. In the first step, the H_8 synthesis process cannot be solved for a generalized case but must be carried out for each plant individually. Some weighing schemes, however, have been developed for various typical design problems, which can be used as guidance or building blocks toward a specific application. Theoretical foundation for the general framework for solving the H_8 design problem is discussed in [5], [27].

H_8 control concept has undergone substantial progress during the nineties and emerged as a method that can be applied to practical problems [11], [13], [37]. Several efficient numerical algorithms have been developed and are readily available to simplify the computation efforts in synthesizing H_8 optimal and suboptimal controllers and provide a somewhat systematic procedure for implementation.

Algorithms used to support H_8 concepts and controller synthesis, by solving the standard optimization problem, require high computing and storage capacity. Use of commercially available software is desirable because of the degree of complexity in the theory supporting the H_8 techniques. Two such packages are readily available:

MatlabTM – It provides two different toolboxes implementing algorithms for the synthesis of H_8 controllers solving the standard optimization problem: 1) “Robust Control” toolbox providing frequency domain solution, also known as the 1984 Solution (Francis and Doyle, 1987) using the Youla parameterization method; 2) “ μ -Analysis and Synthesis” toolbox providing the state space solution, also known as the two Riccati equations solution (Glover and Doyle, 1989).

MatrixxTM – It provides 1) module for “Robust Control Design” in which tools are provided for the system analysis using H_8 concepts, and for H_8 controller synthesis solving the standard optimization problem; 2) module for “ X_μ ” in which tools for the μ synthesis and analysis are provided.

3. Adaptive Inverse Control

3a. Background

The adaptive inverse control method discussed in this report is focused on compensating for type of nonlinearities associated with sensors and actuators. This section describes in a rather general sense the concept of nonlinear compensation as introduced in [12]. The methods developed under this concept are based on three main forms of nonlinearities characterized as the dead-zone, backlash, and the hysteresis that might be present at the input or/and the output of a linear part of a plant. More complex forms of nonlinearities can be possibly characterized as a combination of these elementary forms of imperfections. The presence of such nonlinearities influences the performance by affecting static accuracy of the control systems. When using linear controllers with plants having input or output nonlinearities of these types, static accuracy will be limited by the width of the dead-zone, backlash, or hysteresis. Any attempts to improve the performance by increasing the gain of the feedback loop would lead to sustained oscillations which consequently may contribute to more wear in mechanical, electromechanical, and hydraulic components and systems. Backlash and hysteresis are more harmful than dead-zone for because of their inherent phase lag.

There are two main approaches to remedy the problems attributed to nonlinearity. One approach is to produce components free from nonlinearity imperfections, but these are costly to manufacture. The other approach is to use inexpensive components that are commercially available and develop a form of intelligent control system capable of compensating for the system nonlinearity imperfections. Performance of the control system can be further improved by incorporating an adaptation feature to the design in order to compensate for changes in the nonlinear characteristics that may take place over time due to wear and tear, and unanticipated variations to the system parameters.

Adaptive control of nonlinear systems has recently made significant advances. Wide range of Adaptive controlling schemes have been developed for special classes of nonlinear systems such as pure-feedback systems and feedback linearizable systems [6], [16], [18]. The study of adaptive control for systems with unknown dead-zones at the input was considered in [24] in which an adaptive method was proposed with full state measurement. Cases where only single output is accessible for measurement were addressed using direct control method in [31, 32], for actuator nonlinearities and in [33-35] for sensor nonlinearities.

The adaptive inverse control concept is based on approximating nonlinearities by piecewise linear models that are parameterized by unknown break-points (b_r, b_l) or crossing points (c_t, c_b, c_r, c_l), depending on the form of nonlinearity, and slopes (m_r, m_b, m_r, m_l) as shown in Figure 3.1. The inverse control method requires a linearized model of the nonlinearity that can be used as the basis for constructing a linear compensator containing a fixed or adaptive inverse dynamics to cancel the effect of the unknown nonlinear characteristics. The compensator dynamics represent the inverse effects of the nonlinearity to be compensated for, and it can be fixed or adaptive. Because of the adaptive feature of this method, a universal linearized model can be derived with the capability to make automatic adjustment to certain parameters. In general, it would be difficult to develop one linearized model that can represent one form of nonlinearity for all applications. This is true due to the physical change in the nonlinear character overtime as a result of the wear and tear that are naturally occur during operation

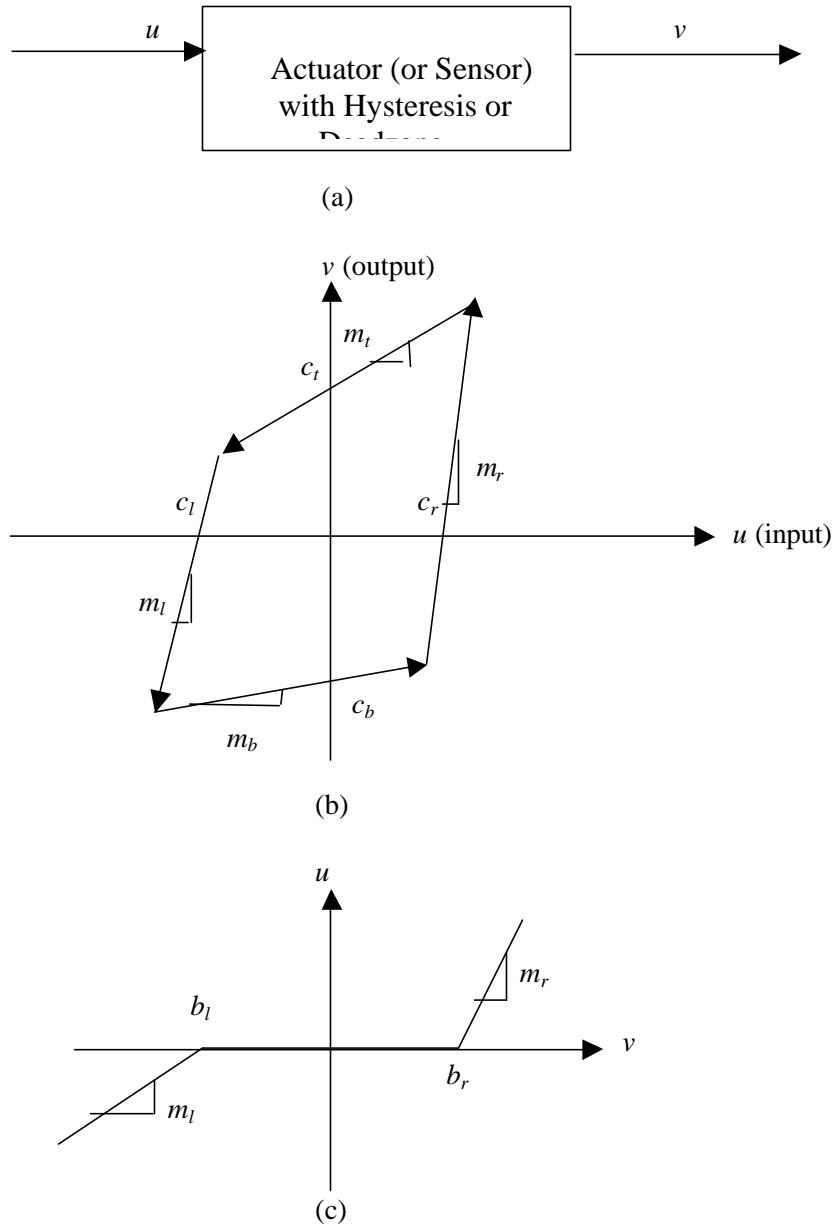


Figure 3.1. A piecewise model of Hysteresis showing eight parameters (four crossing points and four slopes)

3b. Adaptive Inverse Control Concept

Actuator and sensor nonlinearities are among the key factors limiting both static and dynamic performance of feedback control systems. Backlash in gears and other mechanical components, for example, prevents accurate positioning and may lead to chattering and limit-cycle instabilities. This in turn increases backlash. In general, nonlinearity could be present at the input of the plant, output of the plant, or at both the input and the output as illustrated by Figures 3.2(a), 3.2(b), 3.3(c), where $N(\cdot)$, $N_i(\cdot)$, $N_o(\cdot)$ represent the type of nonlinearities exhibited by the physical components. The goal of the inverse methodology is to develop an inverse scheme capable of

canceling the effects of the nonlinearities; thus allowing a linear controller structure to be used to achieve improved output-tracking performance.

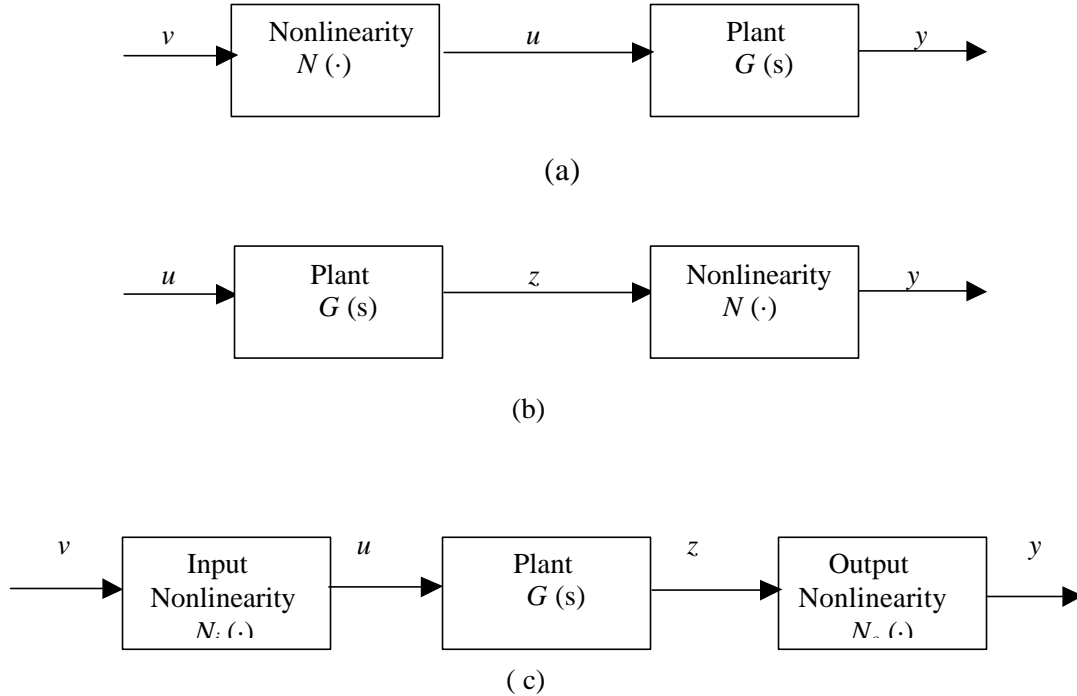


Figure 3.2. Plant Models with (a) input nonlinearity, (b) output nonlinearity, (c) input and output nonlinearity.

The underlying concept of adaptive inverse control strategy is to construct linear controllers in the form of a control algorithm including a class of nonlinear compensators characterized by a fixed or adaptive inverses of the most commonly encountered nonlinear elements in sensors and actuators. Development of this type of nonlinear compensators depends mainly on the existence of appropriate inverse models capable of canceling the effects of the nonlinearity to a practical degree of accuracy and ease of implementation. These issues were initially raised and considered in [26], [27]. It has been shown in [10] that inverse models exist for all three forms of nonlinearities with the dead-zone being the simplest inverse model due to its static nature. Backlash and hysteresis and their inverses are more complex because of the dynamic character they exhibit and that there is no universal model that can represent these nonlinearities for all systems and components.

As depicted by Figure 2.9, the existence of an inverse model $NI(\cdot)$ for a given nonlinearity $N(\cdot)$ implies that

$$u = N(NI(u_d)) = u_d,$$

where $N(\cdot)$ represents the nonlinear element and $NI(\cdot)$ is its true or desired inverse. However, true inverses are not usually known and parameterized estimates $NI(\cdot)$ must be derived which

can be either fixed or adaptive. It should also be noted that when the inverse is exact, then $u = N(NI(u_d)) = u_d$ achieves the control objectives as if $N(\cdot)$ does not exist.

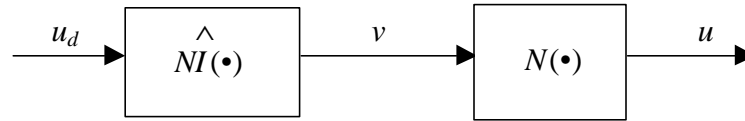


Figure 3.3. Compensation of input nonlinearity $N(\cdot)$.

The main objective in developing adaptive inverse compensation schemes is to account for the uncertainties by understanding the characteristics of the nonlinearities and their variation with time. In some situation, however, when the nonlinearities are approximately known, compensation schemes based on fixed approximate inverses may be satisfactory. This would lead to various development approaches depending on the knowledge of the linear dynamics of the plant and the characteristics of the nonlinearity.

Four approaches were presented in [10]:

- Both the nonlinear part $N(\cdot)$ and the linear part $G(s)$ are known.
- The nonlinear part $N(\cdot)$ is known and the linear part $G(s)$ is unknown.
- The nonlinear part $N(\cdot)$ is unknown and the linear part $G(s)$ is known.
- Both the nonlinear part $N(\cdot)$ and the linear part $G(s)$ are unknown.

In the first two cases the compensation scheme utilizes an exact inverse to cancel the nonlinearity. In these cases, fixed or adaptive linear controller structure can be used to achieve output tracking to a given reference signal. In the other two cases the compensation scheme uses a detuned inverse, whose parameters are fixed estimates of the unknown exact inverse parameter and a fixed or adaptive linear controller structure can be used. Such compensation scheme ensures closed-loop signal boundedness if the inverse slope parameter errors are small.

3c. The underlying Control problem

Although this report is devoted in the most part to the concept of adaptive inverse control as a technique to cancel nonlinear imperfections for sensors and actuators, the ultimate objective is to address the technical issues confronting control engineers in designing a robust control system for a plant with sensor and actuator nonlinearities.

In this section a systematic approach will be outlined which serves as a guidance in designing a robust control system for plants with sensor and actuator nonlinearities. The procedure should not be considered as a design algorithm since each application has its own design constraints based on the plant and its operating requirements. The design objective may also be different from plant to plant. Therefore a generic algorithm is not feasible for all applications.

The nonlinear cancellation procedure is based on the assumption that the plant is represented by a linear-time-invariant dynamics and that the nonlinearity appears at the input of the plant, output or both. The recommended procedure is illustrated by the flow chart of Figure 4.1.

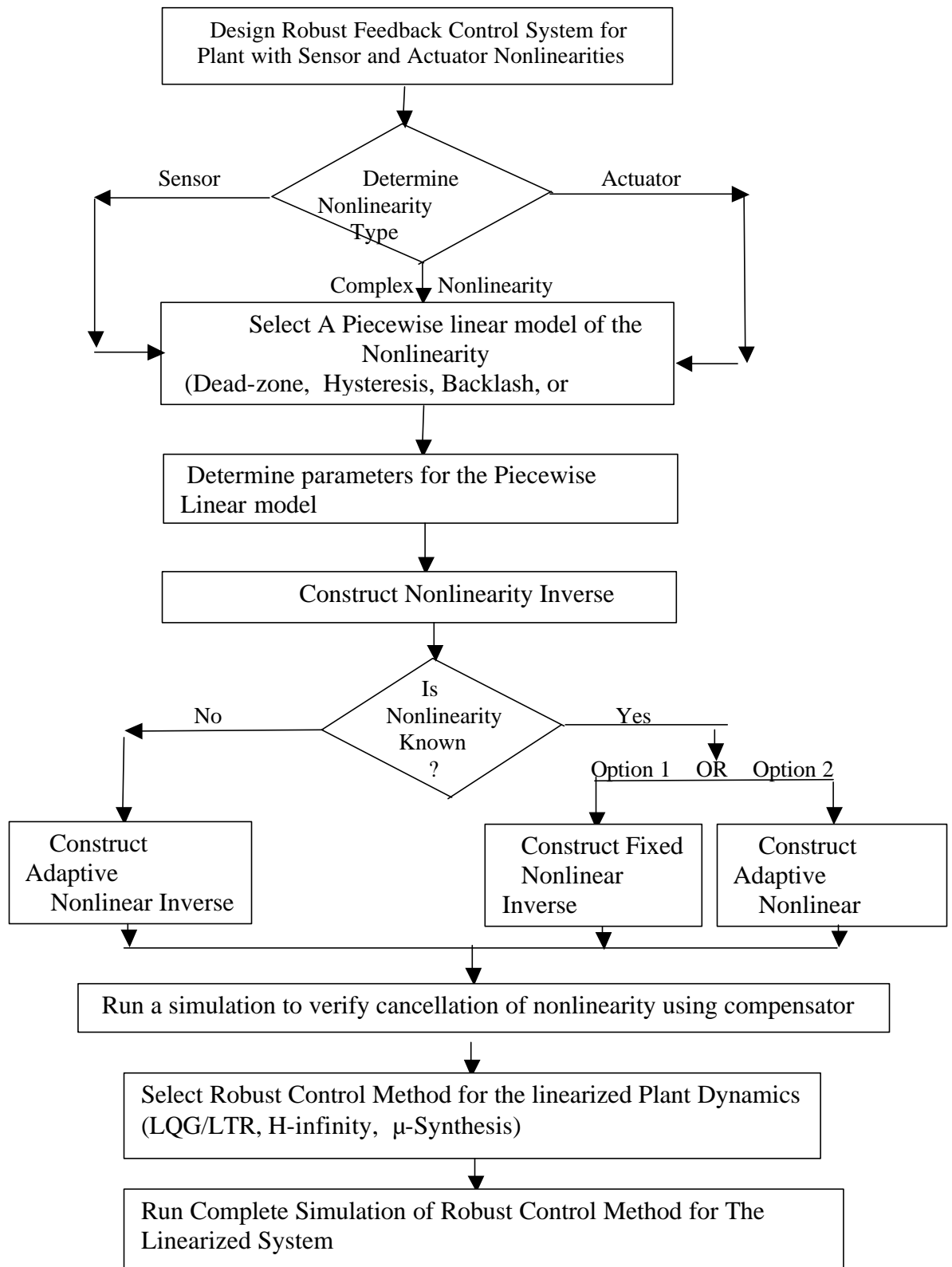


Figure 4.1 Flow chart outlining design steps for robust feedback control system of plants with sensor and actuator nonlinearities

3d. Dead Zone Compensation

As discussed previously, our objective is to develop inverse algorithms to cancel the effects of nonlinear sensors and actuators. In this section we will develop an algorithm for dead zone compensation. The actuator is represented schematically in Figure 3.3.

The output (u) is described by the equation

$$u(t) = \begin{cases} m_r(v(t) - b_r) & \text{for } v(t) \geq b_r \\ 0 & \text{for } b_l < v(t) < b_r \\ m_l(v(t) - b_l) & \text{for } v(t) \leq b_l \end{cases}$$

The input/output relationship of an actuator with a dead zone nonlinearity is shown in Figure 3.4. In this figure the output (u) follows the demand (v) until $v = b_l$ at which point u remains constant until $v = b_r$.

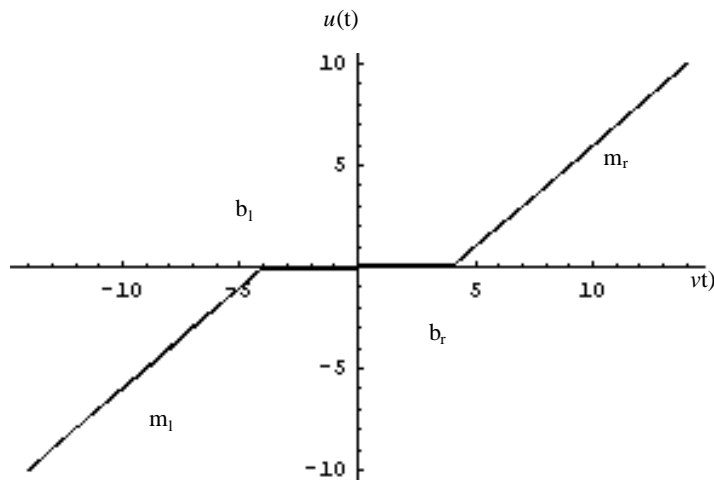


Figure 3.4 Input/output relationship of actuator with dead zone

In Figures 3.5 and 3.6 we show the input and output of an actuator with a dead zone nonlinearity. The input is specified by the equation

$$u_d(t) = 10 \sin(1.3t)$$

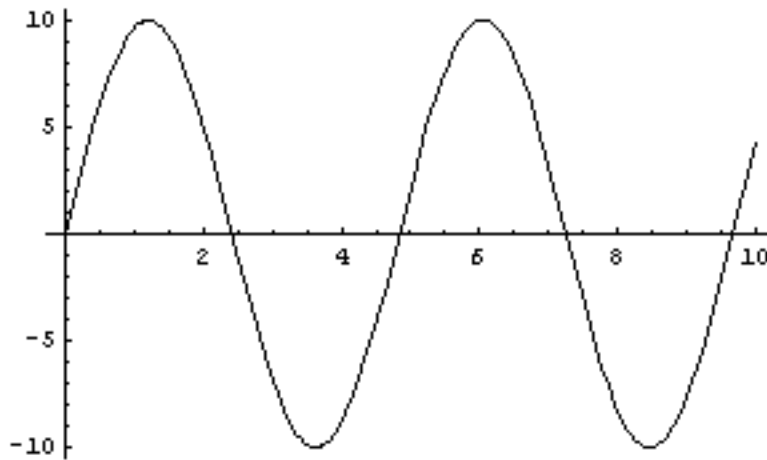


Figure 3.5. Input signal to actuator.

The effect of the dead zone can be seen in the graph where $u=0$ and in the magnitude of u . We will compensate for the dead zone by inserting an inverse algorithm for the dead zone as shown in Figure 3.3. The dead zone inverse can be described mathematically by the equation

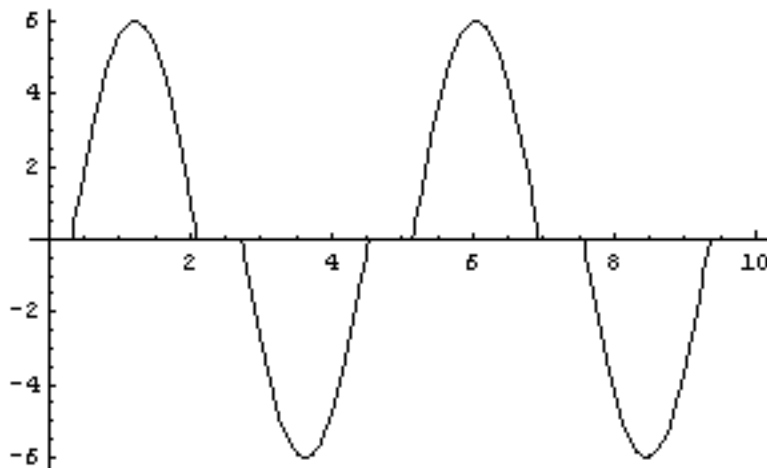


Figure 3.6. Output signal from nonlinear actuator with dead zone.

$$v(t) = \begin{cases} \frac{u_d(t) + m_r b_r}{m_r} & \text{for } u_d(t) > 0 \\ 0 & \text{for } u_d(t) = 0 \\ \frac{u_d(t) + m_b l}{m_l} & \text{for } u_d(t) < 0 \end{cases}$$

Figure 3.7 shows the actuator output (u) with after compensation. It is clear from comparing Figures 3.5 and 3.7 that the output tracks the input (u_d)

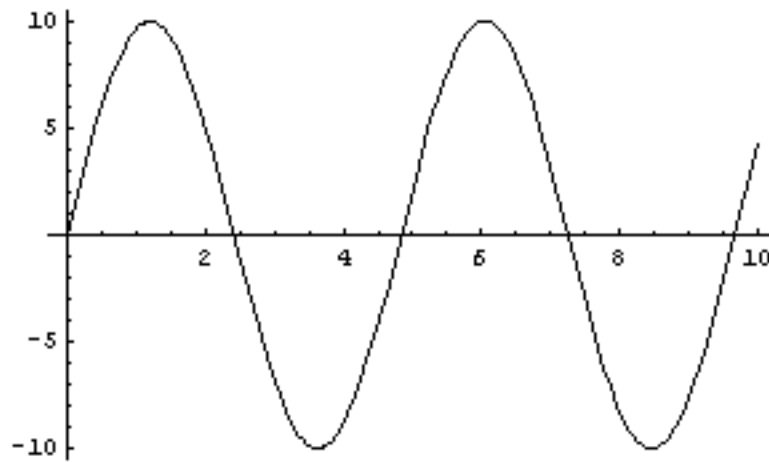


Figure 3.7. Output signal from nonlinear actuator with dead zone compensation.

3e. Backlash Inverse

Backlash is a common problem in actuators which use gears. The backlash in the system makes precise positioning difficult. In this section we develop the backlash algorithm and demonstrate its effectiveness.

Figure 3.8 will be used to represent the meshing of gear teeth (although slightly exaggerated for clarity). In this figure object A represents the (gear) tooth that is driving object B. As A moves to the right, it will come into contact with object B, which will begin to move to the right. As A reverses direction and begins to move to the left, B will remain stationary until A comes into contact with the left side of B. Then both objects will begin to move to the left. The relationship between the position of A (v) and B (u) is represented in Figure 3.9. It should be clear that the position of B is dependent on the position of A, the direction A is moving, and the previous position of B.

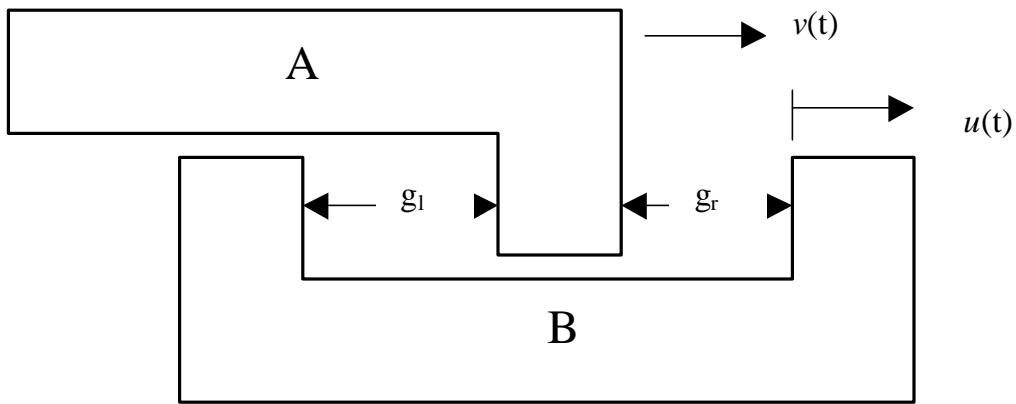


Figure 3.8. Representation of meshing gear teeth.

To position of B as a function of time can be represented by the formula

$$u(t) = \begin{cases} v(t) & \text{for } \frac{dv}{dt} > 0 \text{ and } u(t-1) < v(t) \\ v(t) + g & \text{for } \frac{dv}{dt} < 0 \text{ and } u(t-1) - g > v(t) \\ u(t-1) & \text{otherwise} \end{cases}$$

where $g = g_r + g_l$

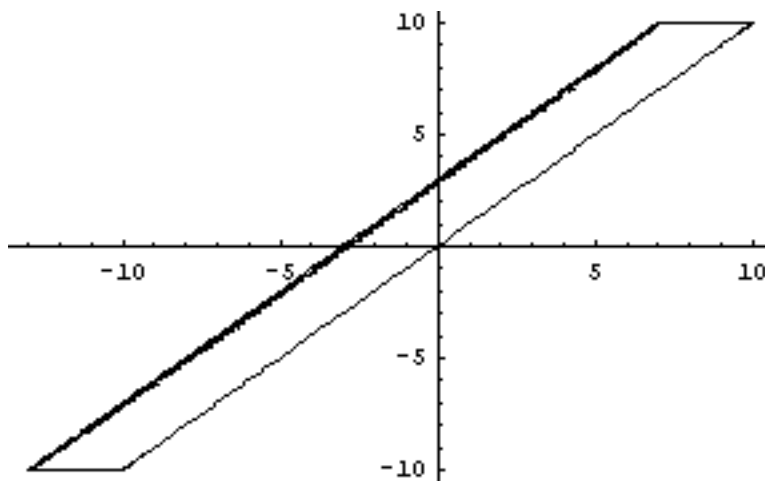


Figure 3.9. Input/output relationship for actuator with backlash nonlinearity.

Figure 3.10 shows the input (desired output) specified by the equation

$$v(t) = 10 \sin(1.3t)$$

Figure 3.11 shows the output of an actuator with backlash for the input shown in Figure 3.10. It is clear from figure that the output is “clipped” and minimum value is limited by the gap (g) in the mesh.

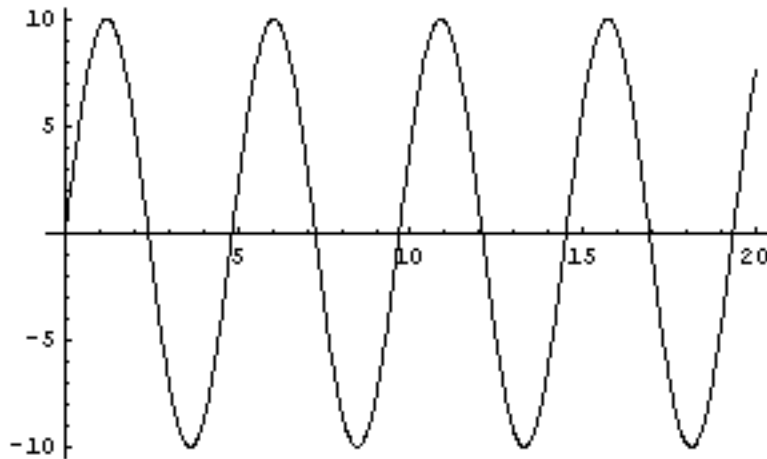


Figure 3.10. Input signal to actuator with backlash nonlinearity.

4We will compensate for the backlash by inserting a nonlinear compensator. The output of the compensator is described by the following equation

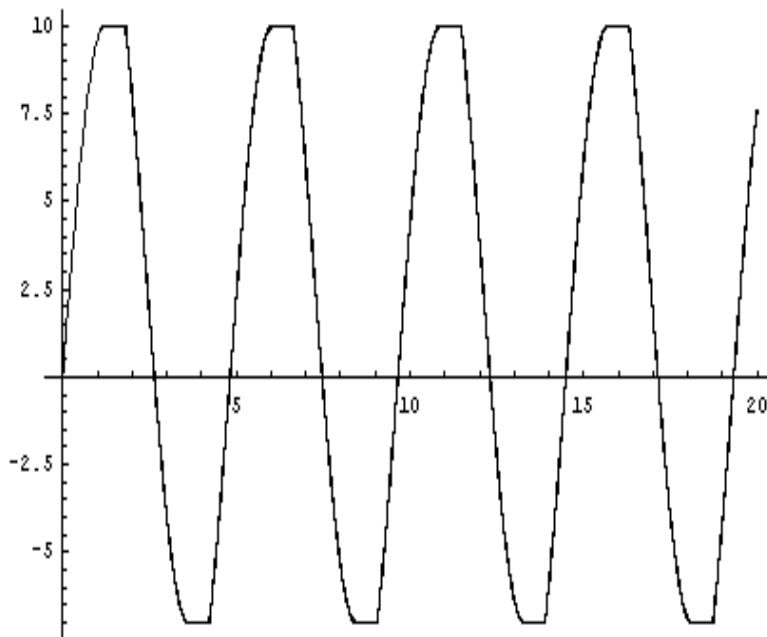


Figure 3.11. Output signal from actuator with backlash nonlinearity

$$v(t) = \begin{cases} u_d(t) & \text{for } u(t-1) \leq u_d(t) \text{ and } \frac{du_d}{dt} > 0 \\ u_d(t) - g & \text{otherwise} \end{cases}$$

Figure 3.12 shows the output (u) following addition of the nonlinear compensator. It can be seen that the “clipping” and minimum value error have been corrected.

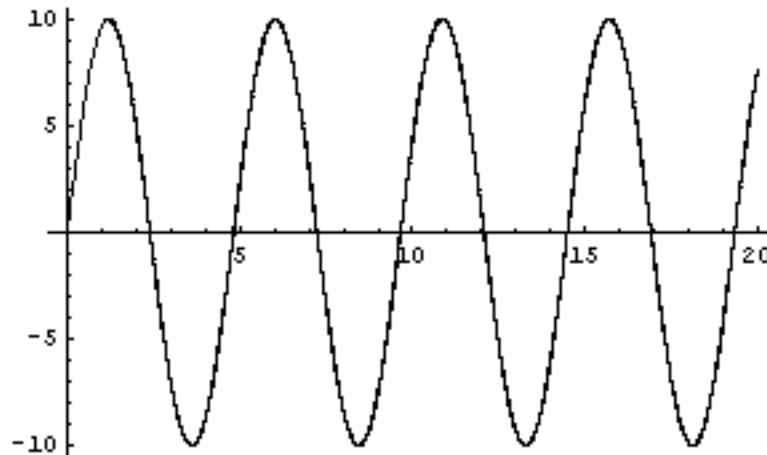


Figure 3.12. Output of actuator with nonlinear backlash following addition of nonlinear compensator.

Additional work is to be done with the dead zone and backlash compensation to add an adaptive feature to adjust the parameters of the compensation to changes (and differences) in the plants to be controlled. Also, additional compensators will be developed and implemented to build this library of routines to be used for nuclear plant control development.

4. Commercially Available Control-Design Software

Nonlinear Control Design (NCD) Blockset

NCD provides a time-domain-based optimization approach to control system design. This tool, designed for use with Simulink block diagrams, automatically tunes system parameters based on user-defined time-domain performance constraints. The NCD Blockset provides an interactive GUI for modifying system parameters and specifying performance constraints. A constrained optimization scheme is formulated and solved using information from Simulink system, time-domain constraints, tunable variables, and uncertainty. The optimization scheme performs successive simulations on the Simulink system and changes the tunable variables in an attempt to better meet the time response specifications.

Nonlinear Control Toolbox

The Nonlinear Control Toolbox was created by Jens Moller-Pedersen and Martin Pagh Petersen (part of masters thesis), The software was developed as a toolbox to facilitate the analysis of H-infinity problems for nonlinear control (NC) systems in conjunction with Maple software package. The NC-Toolbox includes functions for basic system manipulation, H-infinity synthesis functions and functions to evaluate general controllers.

The toolbox has been developed on HP-UX 900 system and requires Maple And Matlab with control and μ -toolbox in order to run. If μ -toolbox is not available, The NC-toolbox can be converted for use with Matlab standard control toolbox. The toolbox can export systems and controllers or simulation with Simulink, but access to Simulink is not required.

Matlab

MatlabTM from MathWorks, Inc., is a software package widely used by academic, scientific, and industrial sectors in control design and analysis. The software consists of individual toolboxes including the 'Control System Toolbox'. It also has two different toolboxes implementing algorithms for the synthesis of H_8 controllers solving the standard optimization problem: 1) "Robust Control" toolbox providing frequency domain solution, also known as the 1984 Solution (Francis and Doyle, 1987) using the Youla parameterization method; 2) " μ -Analysis and Synthesis" toolbox providing the state space solution, also known as the two Riccati equations solution (Glover and Doyle, 1989).

Matrixx

MatrixxTM – from Integrated Systems provides 1) module for "Robust Control Design" in which tools are provided for the system analysis using H_8 concepts, and for H_8 controller synthesis solving the standard optimization problem; 2) module for " X_μ " in which tools for the μ synthesis and analysis are provided.

Computer-aided Systems and Control analysis and Design (CASCADE)

CASCADE is an expert system developed by the University of Tennessee at Knoxville (UTK) for the U. S. Department of Energy (DOE). The expert system is a tool used in synthesizing the LQG/LTR eliminating the need for numerical programming that is typically involved in developing the algorithm.

References

- [1] A. Gelb and W. E. Velde, “*Multiple-Input Describing Functions and Nonlinear Design*,” New York: McGraw-Hill Book Company, 1968.
- [2] Ali Sabri and Ben Chen, “*H₂ Optimal Control*,” Printice Hall, 1995
- [3] Atherton, DP, “*Nonlinear Control Engineering*,” Van Nostrand Reinhold, 1982
- [4] Bernard Widrow and Eugene Walach, “*Adaptive Inverse Control*,” Printice Hall, 1995.
- [5] Bruce A. Francis, “*A Course in H_∞ Control Theory*,” Springer-Verlag, 1987
- [6] Camion, G. and G. Bastin, “*Indirect adaptive state feedback control of linearly parametrized nonlinear systems*,” international Journal of Adaptive Control and Signal Processing, vol. 4, pp. 345-358, 1990.
- [7] Charles L. Phillips and H. Troy Nagles, Jr., “*Digital Control Systems Analysis AND Design*,” N.J. : Printice-Hall, 1984.
- [8] Chua, L. O. and S. C. Bass “A generalized hysteresis model,” *IEEE Trans., Circuit Theory*, vol. 19, no. 1, pp. 36-48, 1972
- [9] David Banjerdpongchai, and Jonathan P. How, “Parametric Robust H_∞ Control Design Using LMI Synthesis,” *Proceedings of AIAA*, 1996
- [10] David Banjerdpongchai, and Jonathan P. How, “Parametric Robust H_∞ Controller Synthesis: Comparison and Convergence Analysis,” *AACC*, 1998
- [11] E. A. Jonckheere, Wwo-Ruey Tu and Chung-Kuang Chu, “H-infinity control of crippled aircrafts with throttles only,” *IFAC 1996 Triennial World Congress*, Session 8a-08, Flight Control Systems, San Francisco, California, pp. 219-224, June 30 -July 5, 1996.
- [12] Gang Tao and Peter V. Kokotovic, “*Adaptive Control of Systems with Actuator and Sensor Nonlinearities*,” John Wiley and Sons, Inc., New York, 1996.
- [13] G. Papageorgiou, “Robust H-infinity Loop Shaping and Aerospace Applications,” PhD Thesis, July 1998.
- [14] G. V. Murphy and J. M. Bailey, “Robust Control Technique for Nuclear Power Plants,” Instrumentation and Controls Division, Oak Ridge National Laboratory, *ORNL/TM-10916*, March 1989.
- [15] H. Kwakernaak and R. Sivan, “*Linear Optimal Control Systems*,” Wiley-Interscience, New York, 1972.

- [16] Isidori, A., “*Nonlinear Control Systems*,” 3rd ed., Springer-Verlag, Berlin, 1995.
- [17] Jean-Jacques Slotine and Weiping Li, “*Applied Nonlinear Control*,” Printice Hall Engineering/Science/Mathematics, 1990
- [18] Kanellakopoulos, I., “Passive adaptive control of nonlinear systems,” *International Journal of Adaptive Control and Signal Processing*, vol. 7, no. 5, pp. 339-352, 1993
- [19] Kransnoselskii, M. A. and A. V. Pokrovskii, “*Control systems with Hysteresis*,” Springer-Verlag, Berlin, 1983.
- [20] Mareels, I. M. Y. and R. R. Bitmead, “Nonlinear dynamics in adaptive control: chaotic and periodic stabilization,” *Automatica*, vol. 22, no. 6, pp. 641-655, 1986.
- [21] Mayergoyz, I. D., “*Mathematical Models of Hysteresis*,” Springer-Verlag, Berlin, 1991.
- [22] Merritt, H. E., “*Hydraulic Control Systems*,” John Wiley & Sons, Inc., New York, 1967.
- [23] Nonlinear Control and Servo Systems, Lecture notes, Department of Automatic Control, Lund Institute of Technology
- [24] R.A. Freeman and P.V. Kokotovic, “*Robust Nonlinear Control Design*,” Springer-Verlag, 1996
- [25] R. Bellman, “*Adaptive Control Process: A Guided Tour*,” Princeton, N.J. : Princeton University Press, 1961.
- [26] Recker, D., P. V. Kokotovic, D., Rhode and J. Winkelman, “Adaptive nonlinear control of systems containing a dead-zone,” Proc. of the 30th *IEEE Conference on Decision and Control*, pp. 2111-2115, Brighton, England, 1991.
- [27] Robust Control Techniques, *Tutorial Document, Group for Aeronautical Research and Technology in EUROpe (GARTEUR)*, April 4, 1997.
- [28] Stephen Boyd, “Robust Control Tools: Graphical User-Interfaces and LMI Algorithms,” *Report published by Information Systems Laboratory, Stanford University*, April 23, 1994
- [29] Tao, G. and P. V. Kokotovic, “Adaptive control of systems with backlash,” *Automatica*, vol. 29, no. 2, pp. 323-335, 1993.
- [30] Tao, G. and P. V. Kokotovic, “Adaptive control of plants with unknown hysteresis,” *IEEE Trans. on Automatic Control*, vol. 40, no. 2, pp. 200-212, 1995
- [31] Tao, G. and P. V. Kokotovic, “Adaptive control of systems with backlash,” *Automatica*, vol. 29, no. 2, pp. 323-335, 1993.

- [32] Tao, G. and P. V. Kokotovic , “ Adaptive control of plants with unknown hystereses ,” *IEEE Trans. on Automatic Control*, vol. 40, no. 2, pp. 200-212, 1995.
- [33] Tao, G. and P. V. Kokotovic, “Adaptive control of systems with unknown output hystereses ,” *Proc. of the 1994 American Control Conference* , pp. 870-874, Baltimore, MD.
- [34] Tao, G. and P. V. Kokotovic, “Adaptive control of plants with unknown output backlash,” *IEEE Trans. on Automatic Control*, vol. 40, no. 2, pp. 326-330, 1995.
- [35] Tao, G. and P. V. Kokotovic, “Adaptive control of plants with unknown output dead-zones,” *Automatica*, vol. 31, no. 2, pp. 287-291, 1995.
- [36] Vaughan, N. D. and J. B. Gamble, “The modeling and simulation of a proportional solenoid valve,” the *ASME Winter Annual Meeting*, Dallas, TX, November 1990.
- [37] Urs Cristen, Martin F. Weilenmann, and Hans P. Geering, “Design of H₂ and H_∞ Controllers with Two Degrees of Freedom,” *Proceedings of American Control Conference*, pp. 2391-2395, Baltimore, MD, June 29- July 1, 1994.

List Of Symbols and Abbreviations

ARE	Algebraic Ricatti Equation
CLTI	Continuous Linear Time Invariant
DLTI	Discrete Linear Time Invariant
$G(s)$	Plant transfer function
H_8	Hardy Space
LQ	Linear Quadratic
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
LTR	Loop Transfer Recovery
MIMO	Multiple Input Multiple Output
MISO	Multiple Input Single Output
$N(\cdot)$	Actuator or sensor nonlinearity
$NI(\cdot)$	Actuator or sensor inverse nonlinearity
ORNL	Oak Ridge National Laboratory
PID	Proportional Integral Derivative
SISO	Single Input Single Output
$\ A\ $	Norm of matrix A
$\ y\ $	Norm of vector y

Attachment I.

Task 1.2 Control-Priority Mode Selection Algorithm

Control priority mode selection is a strategy that allows autonomous agents to select the best control mode for a given condition. The control system designer provides detailed information needed to select the highest priority regulation objectives required to maintain the controllability of the full system for different failure states of actuators. For example, when one actuator saturates or is placed in manual, that actuator no longer responds to automatic control signals. The controllability of the full system is reduced in rank by one degree due to the loss of one actuator. The highest regulation objective is changed to maintain the most important objective in response to a reduced controllability rank of the full system. Standard linear methods for multi-input, multi-output control do not address the problem of signal boundedness for the actuator or manual/auto transition of actuators singly. The control priority strategy is a methodology for recognizing the condition and providing control modes specifically designed for the diminished control capability of the system. The goal of this task is to develop methods such that, when given the control priorities in an appropriate form, the control priority algorithm implements the control modes and switching logic automatically.

The need for control priority arises in multi-variable control systems. The origin of the idea for control priority strategy was the integrated control system for Babcock & Wilcox-designed nuclear steam supply systems. A simplified description of this system shows that it consists of three controllable systems:

- 1) a nuclear reactor,
- 2) a feedwater system, and
- 3) a turbine.

In normal, fully-automatic, operation the control system has the following actuator goals:

- 1) reactor control rods are used to regulate coolant temperature,
- 2) feedwater valves are used to control power, and
- 3) turbine valves are used control steam pressure.

If one of the actuators is placed in manual or reaches an unresponsive condition then the control system needs to modify control goals to account for the lost of an actuator. For example, if the turbine is placed in manual, the control goal of the feedwater system is changed from controlling power to controlling pressure. Similarly, if the control rods are placed in manual, the goal of the feedwater system is changed from controlling power to controlling reactor temperature.

This illustrative example is quite simple. Because the priority scheme was simple, it was possible to develop the control schemes and switching logic for the diminished systems manually without great difficulty. The specific goal of this task is to construct an algorithm such that a module or system can construct the switching logic and control methods automatically from the priorities alone. This degree of on-line control logic design will allow a whole new class of real time reconfigurable multivariate control systems to be handled.

The concept of control priority mode switching is to formulate the information requirements to uniquely determine system control goals in response to changes in the status of control actuators or input signals. The ordered priority of regulation objectives

with suitable controllability/observability restrictions appears to be sufficient to reconfigure the control system algorithm automatically.

The following hierarchical structure has been used to map control system components for a mode selection algorithm. The basic element in this structure is called a socket. The highest level socket is called a system_socket. The system_socket contains five or more elements:

- 1) the number of sockets required by the system,
- 2) the number of components available for use by the control system,
- 3) the number of control algorithms available,
- 4) the control algorithm being used,
- 5) a label defining the control system, and
- 6) other elements as needed.

The next level in the structure is the control_socket. The control_socket structure contains two or more elements:

- 1) the number of components assigned to a specific socket,
- 2) an array of component indexes pointing to the specific components assigned for use by this control_socket, and
- 3) other elements as needed.

The lowest level is the components structure. The number of components should be larger than the number of sockets if redundancy is required by the control system. The component structure contains seven or more elements:

- 1) socket number,
- 2) status,
- 3) time placed in service,
- 4) number of times repaired,
- 5) parameter one of a probability distribution function (PDF),
- 6) parameter two of a probability distribution function,
- 7) a component label, and
- 8) other elements as needed.

We have used a Weibull PDF to predict mean time to failure for the components; thus, parameter1 is the alpha and parameter 2 is the beta for the Weibull PDF. The Weibull PDF will be described later. Each system has additional information stored in a system level database. The additional system data base entries contain a variety of information describing the control system. For instance, the socket activation sequence for turning the control system on, the deactivation sequence for turning the system off, the number of control-sockets for this system, the control algorithms to be used, and other specific system information as determined by the control designer.

A control system can be defined using the hierarchical structure defined above. For instance, a control system providing liquid level control is defined using this structure. The level control system is designed using six control_sockets that are populated with two configuration valves, a pump, a control valve, a level set point, and a level measurement as indicated in Table 1.2.1 and there are three control_sockets that have redundant components. Selection of a pump, control valve, and level measurement as

redundant components is a means used by the control system designer to improve system reliability.

The system_socket is populated by

- 1) number of sockets = 6,
- 2) number of components = 9,
- 3) number of control algorithms = 2,
- 4) control algorithm being used = 1, and
- 5) system label = "LCS100".

The control_socket is defined by an array of 6 sockets to match the number of sockets defined by the system_socket structure. Each of the control_socket elements is defined by:

control_socket 1

- 1) number of components assigned to socket 1 = 2, and
- 2) component indexes for socket 1 = [1 and 2];

control_socket 2

- 1) number of components assigned to socket 2 = 1, and
- 2) component indexes for socket 2 = 3;

control_socket 3

- 1) number of components assigned to socket 3 = 2, and
- 2) component indexes for socket 3 = [4, 5];

control_socket 4

- 1) number of components assigned to socket 4 = 1, and
- 2) component indexes for socket 4 = 6;

control_socket 5

- 1) number of components assigned to socket 5 = 1, and
- 2) component indexes for socket 5 = 7; and

control_socket 6

- 1) number of components assigned for socket 6 = 2, and
- 2) component indexes for socket 6 = [8, 9].

An example of component_ structure values for each of the nine components is shown in Table 1.2.1.

Table 1.2.1 Component_Structure

Component number	1	2	3	4	5
Socket number	1	1	2	3	3
Status	1	1	1	1	1
Time placed in service	5	12	5	5	5
Number of times repaired	0	0	0	0	0
PDF Parameter 1	10.0	9.0	12.0	8.0	10.0
PDF Parameter 2	4.0	4.0	6.0	5.0	4.0
Component label	“PUMP1”	“PUMP2”	“CONFV1”	“CTRLV1”	“CTRLV2”

Table 1.2.1 Component Structure (Continued)

Component number	6	7	8	9
Socket number	4	5	6	6
Status	1	1	1	1
Time placed in service	10	5	5	5
Number of times repaired	0	0	0	0
PDF Parameter 1	9.0	12.0	8.0	8.0
PDF Parameter 2	4.0	6.0	5.0	5.0
Component label	“CONFV2”	“LVLSP”	“LVLM1”	“LVLM2”

Now that the structure has been defined, the next step is to define the task of the mode selection algorithm. The mode selection algorithm determines which control algorithm to use and which of the available components to activate.

Plant observability plays a key role in a mode selection algorithm therefore, it designed to be data driven using this hierarchical structure. The primary structure element used by the algorithm for dynamic mode selection is the component status. If the status is one then the component is operational. Thus, a simple boolean equation is used to evaluate the status of the level control system,

LCS100 = (PUMP1 “OR” PUMP2) “AND” CONFV1 “AND” (CTRL1 “OR” CTRL2)
“AND” CONFV2 “AND” LVLSP “AND” (LVLM1 “OR” LVLM2).

In this example, two control algorithms are available for the level control system as indicated in the system_socket level data. The first control algorithm will always be the normal default control algorithm for the system. In this example, the first control algorithm is a standard PID type control to control level using control valve to zero the error between set-point and level measurement. The condition required for this control algorithm is that at least one control valve is working and at least one pump is operational. The second control algorithm is available if both control valves fail to respond properly to the control signal and one of the control valves can be opened. Under this condition the second control algorithm will turn on the pump when the level error fall below some predetermined value and turn off the pump when the level error exceeds some predetermined value. This information is used in the mode selection algorithm to determine which of the available control algorithms to be used to control the system.

The component activation algorithm uses the component_structure level data. The component status is the primary input to determine if the component is available for use. This status is changed by the diagnostic system that is being used to monitor system performance or by maintenance that have removed the component from service for repair. Other systems as desired by the control designer may be given access to the status parameter. If all components available for the control system are operational, then the redundant component with the highest reliability is selected for used. The two PDF parameters, the time placed in service, and the number of times repaired are used to obtain an estimate for the redundant component reliability. In addition, the total system reliability are estimated using these parameters to provide operator information about predicted system reliability.

Information theory provides a parameter, called entropy, which can be used to estimate system complexity. [Min and Chang] Using an entropy measure along with a method to estimate time to failure will provide the mode selection algorithm information for component selection. The time placed in service, the number of times repaired, and the PDF parameters are used in the entropy measure and to estimate time to failure used in the next level of evaluation. The diagnostic system being developed by UTK will supply additional information that is used in mode selection algorithm.

System entropy is dependent upon the number of components in the system and the probability of failure of each component. For the same probability of failure, entropy increases with the number of components in the system. If the failure probability of one component increases, the system entropy also increases. The entropy parameter includes both system complexity and system reliability. Thus selecting the system with the lowest entropy provides a way to weight both complexity and reliability into the selection

criteria. This statement is true when the failure probability for all components is less than 0.5.

The component entropy is determined by

$$E_i \equiv -P_i \log_2 P_i - (1 - P_i) \log_2 (1 - P_i), \quad (1.2.1)$$

where,

E_i = component entropy, and
 P_i = failure probability of component i.

It has been shown that the entropy of a system can be determined by the sum of the component entropies in a system,

$$E \equiv \sum_{i=1}^{nc} E_i \quad (1.2.2)$$

where,

E = system entropy, and
 nc = number of components.

Diagnostic entropy is a measure that indicates the difficulty in determining the state of a system. The diagnostic entropy is determined using conditional probabilities, where the conditional condition is the system is not in the operational state. If the system being analyzed has redundant ways of satisfying the control requirements, the probability of the system to exist in the desired state is determined by

$$PDS \equiv \sum_{j=1}^{ns} \left(\prod_{i=1}^{ncj} P_{ji} \right) \quad (1.2.3)$$

where,

PDS = probability that the system exists in the desired state,
 P_{ji} = probability that component i in system j is in the desired state (working) ,
 ns = the number of redundant systems that satisfy control requirements,
 ncj = the number of components in redundant system j,
 j = index for redundant systems,
 i = component index.

Now examine the system to determine the probability that the system exists in an undesired state, having one or more of the components in the failed condition. The following equation is used to determine the probability that the system is in an undesired state,

$$PF \equiv \sum_{k=1}^{nt} P_k \quad (1.2.4)$$

where,

PF = probability that the system is in a failed condition,
nt = number of failure combinations ($=2^{nc} - 1$),

$$P_k \equiv \prod_{i=1}^{nc} (ss_{ki} \times P_i + \overline{ss_{ki}} \times (1 - P_i)) \quad (1.2.5)$$

where,

$ss_{ki} = 1$ when component i is operational and 0 when failed,
k = condition counter,
nc = number of components in system.

The probability, PF, determined by equation (1.2.4) is used to normalize the probabilities for component failure combinations determined by equation (1.2.5) to yield

$$PN_k \equiv \frac{P_k}{PF} \quad (1.2.6)$$

The diagnostic entropy, ED, is then determined by

$$ED \equiv \sum_{k=1}^{nt} (-PN_k) \times \log_2(PN_k) \quad (1.2.7)$$

The probabilities for components can be modeled using any of the standard probability density functions such as Weibull or Gamma. [Barlow and Proschan] The present analysis uses a Weibull distribution, (1.2.8), to obtain an estimate of component failure time. [Hogg and Tanis] The Weibull probability density function is

$$g(w) \equiv \frac{\mathbf{a} w^{\mathbf{a}-1}}{\mathbf{b}^{\mathbf{a}}} e^{-\left(\frac{w}{\mathbf{b}}\right)^{\mathbf{a}}} \quad (1.2.8)$$

where,

w = time parameter,
 \mathbf{a} = parameter,

b = parameter.

The Weibull distribution for $a = 10$. and $b = 4$.is shown in Fig. 1.2.1.

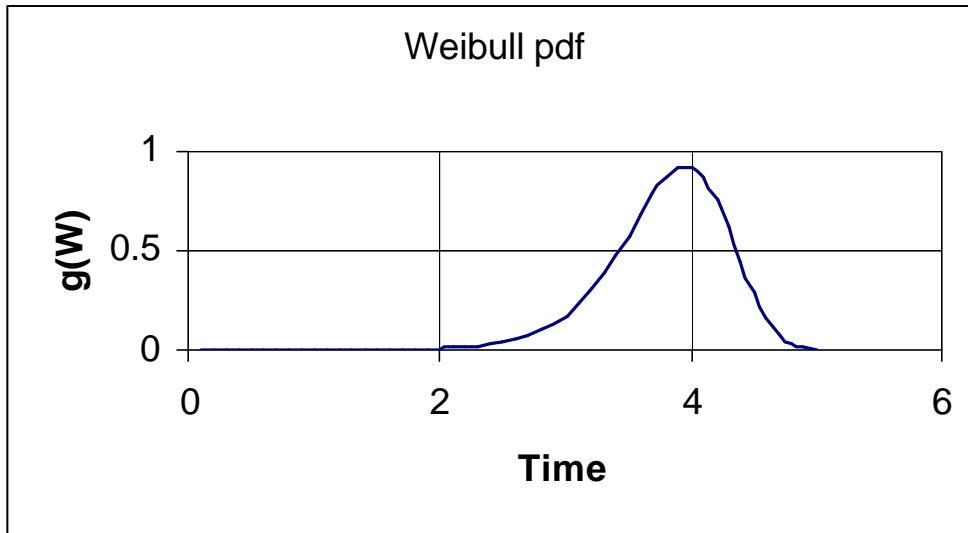


Fig.1.2.1 Weibull Probability Density Function

The cumulative distribution, $G(w)$, is determined by

$$G(w) = \frac{\int_0^w g(w) dw}{\int_0^{\infty} g(w) dw} \quad (1.2.9)$$

The cumulative Weibull distribution shown in Fig 1.2.2 can be used to predict the changes in failure probabilities for system.

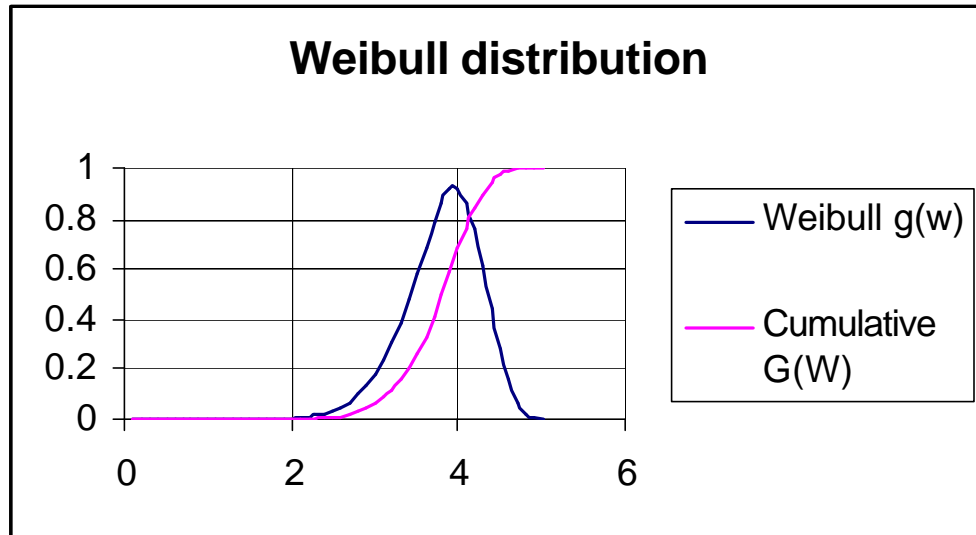


Fig. 1.2.2. Cumulative Weibull distribution use for failure predictions

Example to demonstrate the entropy calculations required by a mode selection algorithm.

Each component will have data base entries that will contain a , b , w_0 , and other information that can be used in tracking the reliability of the component. The Weibull parameters are used to adjust probability data to match the expected component probabilities and w_0 is set to the time placed in service for the component use in the system. This parameter is reset when the component is replaced. A set of rules is developed to determine which control goal to be used. Inputs to this algorithm are generated by the diagnostic system, which will provide information about component malfunctions or systems being switched to manual. Once the control goal has been determined. The next decision is based on which redundant path will provide the most reliable components to meet the control goal.

Selected components will be evaluated by the diagnostic system. The evaluation is used to generate an effective failure probability. The effective failure probability is compared to the failure probability generated by a cumulative Weibull function. The largest value is used in the entropy calculation to determine to most reliable control sections to use.

Table 1.2.1 shows nine components that have been used to control the system for this example. The nine components are used to satisfy the control goals of the level control system.

TABLE 1.2.2 Control System Components

Component	Failure Probability	Entropy
1	.1	.46899559
2	.2	.72195809
3	.1	.46899559
4	.2	.72195809
5	.2	.72195809
6	.1	.46899559
7	.05	.28639696
8	.2	.72195809
9	.25	.81127812

Table 1.2.2 provides the failure probabilities and the entropy values for the components assigned to the level control system. Components 1 and 2 are a redundant set assigned to socket 1. Since component one has a lower entropy, it is selected by the mode selection algorithm. This selection can be changed if the diagnostic evaluation indicates some problem with component 1. The next redundant pair is components 4 and 5. They have the same value of entropy. Diagnostic system information and the number of times repaired are evaluated to make the selection decision. The last redundant pair in this system is components 8 and 9. If the entropies are not the same but the differences are small then additional information is needed to make the component selection decision. The component entropy for the six selected components is shown in Table 1.2.3

TABLE 1.2.3 System Entropy for Selected Components

Socket	Component	Entropy
1	1	.46899559
2	3	.46899559
3	4	.72195809
4	6	.46899559
5	7	.28639696
6	8	.72195809

System entropy is the sum of selected component entropies and equal to 3.13723991.

References

[Min and Chang] Bok-Ki Min and Soon Heug Chang, "System Complexity Measure in the Aspect of Operational Difficulty," IEEE Transactions On Nuclear Science, Vol. 38 No. 5, pp 1035-1039, 1991.

[Hogg and Tanis] Robert V. Hogg and Elliot A Tanis, "Probability and Statistical Inference," Second Edition, 1983.

[Barlow and Proschan] Richard E. Barlow and Frank Proschan, "Statistical Theory of Reliability and Life Testing Probability Models," 1975.

Attachment III
Task 1.3 Controllers for Feedwater Systems

TABLE OF CONTENTS

SUMMARY	1
1. Introduction	1
2. Model Development	3
4. Simulation Results	9
5. Conclusions	11
Acknowledgments	Error! Bookmark not defined.
References:	12
Appendix	13

SUMMARY

Poor control of steam generator water level of a nuclear power plant may lead to frequent nuclear reactor shutdowns. These shutdowns are more common at low power where the plant exhibits strong non-minimum phase characteristics and flow measurements at low power are unreliable in many instances. There is need to investigate this problem and systematically design a controller for water level regulation. This work is concerned with the study and the design of a suitable controller for a U-Tube Steam Generator (UTSG) of a Pressurized Water Reactor (PWR) which has time varying dynamics. The controller should be suitable for the water level control of UTSG without manual operation from start-up to full load transient condition. Some preliminary simulation results are presented that demonstrate the effectiveness of the proposed controller. The development of the complete control algorithm includes components such as robust output tracking, and adaptively estimating both the system parameters and state variables simultaneously. At the present time all these components are not completed due to time constraints. A robust tracking component of the controller for water level control is developed and its effectiveness on the parameter variations is demonstrated in this study. The results appear encouraging and they are only preliminary. Additional work is warranted to resolve other issues such as robust adaptive estimation.

1. Introduction

For the steam generator (SG) in a nuclear power plant, the main goal of its control system is to maintain the SG water level at a desired value by regulating the feed-water flow rate. Conventional feed-water control schemes cannot provide satisfactory performance within the required wide operating range of 0 to 100% of the specified load. A large proportion of reactor shutdowns at operating nuclear plants, which causes a severe economic loss, has been mainly caused by ineffective feed-water control. Therefore, development of better control schemes is very important. A review of past PWR plant operation experiences indicates that unplanned reactor trips due to SG level control have been significant contributors to plant unavailability. During low power operation, the level control is complicated by the thermal reverse effects known as “shrink and swell”. Due to the destabilizing vapor content in the tube bundle region, the water level measured in the downcomer temporarily reacts in a reverse manner to water inventory change. Increased feed-water flow adds mass to the SG, which would be expected to increase the measured downcomer water level, and does not increase it at high power. But at low power, the cold feed-water addition can cause a decrease in the vapor content of tube bundle, a shift in the liquid from the downcomer to tube bundle, and a temporary decrease in level (shrink). Similarly, a decrease in feed-water flow can cause a temporary increase in water level (swell). These reverse effects are confusing for either manual or automatic operation. The only true indication of water inventory change is the flow mismatch between steam and feed-water.

The water level of the steam generator must not be allowed to rise too high in order to prevent the excessive moisture carryover and the pressure buildup of the

containment in the break of secondary side flow loop. Also, the low water level should be prevented in order to avoid the uncovering of the U-tubes in the secondary side. Therefore, the control of the steam generator water level is important to determine power plant responses in the event of changes in the operating load. The proposed controller is designed to ensure a satisfactory automatic control for the SG water level from low power to full power.

Different approaches to the control of steam generator water level of PWR have been reported in the literature. Kothare et al. [1] and Kothare et al. [2] applied Model Predictive Control (MPC) technique for SG level control. MPC is an open-loop design procedure. This technique works as follows. At sampling time k a plant measurement of the output is obtained. An estimator takes the measured output value and the manipulated input to obtain the estimate of the plant state. An optimizer takes the estimated plant state and computes a sequence of control moves by minimizing an objective function. A more general Linear Quadratic Gaussian (LQG)/Loop Transfer Recovery (LTR)-based controller is proposed by Menon and Parlor [3]. The controller in [3] is developed using local linearization of the SG model and then gain-scheduled to cover the entire range. In their design the linearized model around each equilibrium point is scheduled and fitted as a function of the difference in the hot- and cold-leg temperatures of the UTSG primary side. Water level control of steam generator at low power using Model Reference Adaptive Control method is reported in [4]. In this work a dynamic reference model is used to generate the reference index of performance for the plant in terms of feed-water flow-rate, steam flow-rate and water level. Na [5] and Na and No [6] presented adaptive observer to simultaneously estimate the flow errors and the parameters of the steam generator model. These estimated parameters are then used to design the control action by minimizing a quadratic cost function. A discrete model is used in [4-6] and the controller is designed to work for low power. Research works on robust tracking and observer-based robust controller are reported in [7-10].

We use continuous time model of the plant and apply Linear Quadratic Regulator (LQR) technique to design an optimum controller that forces the plant output (water level) to follow a desired water level pattern. Our model is represented as a function of feed-water flow-rate, steam flow-rate, and water level unlike [3]. The detailed discussion on LQR technique is available in [11]. Robust tracking is achieved for a certain range of parameter drifts. MATLAB software package is used to generate computer simulation results. The results validate the effectiveness of the controller in water level control of the steam generator. We have attempted to develop a robust adaptive observer for the UTSG level control but due to time constraints it is not complete at this time. For various techniques on robust adaptive observer and optimum controller please refer to [12] and [13] respectively.

2. Model Development

A steam generator shows complicated dynamic behaviors with nonlinear characteristics. Some theoretical models based on thermodynamic experiments and/or energy conservative equations have been developed to use for operator training simulator and accident analysis and so on. However, these are inadequate as mathematical models for designing controllers due to complexities. The controller design and the resulting controller performance on the actual plant are both strongly dependent on the accuracy of the mathematical model used to describe the plant. However, a highly accurate model is generally also highly complex and nonlinear, and therefore leads to difficulties in controller design. For the purpose of controller design, the model should be simple and at the same time relatively accurate in describing the principal dynamics of the U-Tube Steam Generator (UTSG). The difficulties in designing an effective level control system for the steam generator arise from a number of factors: (1) the inverse response behavior of the plant, particularly at low operating power due to the so-called “swell and shrink” effects; (2) variation of plant dynamics with operating power; (3) unreliable flow measurements at low power which preclude effective use of feed-forward control; (4) constraints on the throughputs of the main and bypass valves which operate on the manipulated variable, the feed-water flow-rate to the SG. In this study we use the model which has been widely used by many researchers [5] for control purposes. The model is a linear fourth-order model whose parameters depend on reactor power level. The transfer function relating the feed-water flow-rate and the steam flow-rate to the water level is given by,

$$Y(s) = \frac{G_1}{s}(q_w(s) - q_v(s)) - \frac{G_2}{1 + t_2 s}(q_w(s) - q_v(s)) + \frac{G_3 s}{t_1^{-2} + 4p^2 T^{-2} + 2t_1^{-2} s + s^2} q_w(s) \quad (1)$$

where,

$Y(s)$, $q_w(s)$, and $q_v(s)$ are water level, feed-water flow-rate, and steam flow-rate respectively, and t_1 , t_2 , and T are damping time constants and oscillation period respectively.

$\frac{G_1}{s}$ is the mass capacity effect of the UTSG. It integrates the flow difference to calculate the change in water level. This term accounts for the level change due to feed water inlet to steam generator and the steam outlet from it. This quantity means the actual water capacity which critically affects the removal capability of the primary heat. G_1 is a positive constant and does not depend on load. $-\frac{G_2}{1 + t_2 s}$ is the thermal negative effect caused by “swell and shrink”. Since these phenomena exhibit exponential responses for step changes of the feed water flow-rate and the steam flow-rate, they are described by a first-order equation. G_2 is positive and dependent on load. As load increases G_2

decreases. The third term is the mechanical oscillation effect caused by the inflow of the feed-water to the UTSG. This is a mechanical oscillation term due to momentum of the water in the downcomer. All the water removed from the steam is returned to the downcomer and is recirculated. The recirculating water has large momentum acting against relatively small flow-rate changes. When the feed-water flow-rate is suddenly decreased, the water level in the downcomer falls initially and then begins to oscillate. This is due to the momentum of the water in the downcomer keeping the recirculating flow going down initially and then slowing down. The mechanical oscillation disappears completely after a small multiple of the damping time constant. The variable G_3 is positive.

We divide the steam generator dynamics into four linearized regions with respect to operating power level and assume that the dynamics vary linearly over these regions. These variations of the plant parameters with respect to power level are presented in the graphical form. The actual plant parameters may vary differently, so we study the performance of the controller under the situations when the parameter drifts from what is projected by linear interpolation. Systematic approaches such as LQR method is used to derive the control law where some objective functions are minimized to derive an optimal controller. The main objective of the controller is to maintain the water level in the steam generator under various operating levels. We also show the effect of parameter drift on the water level through computer simulation results. To design the proposed controller we transform the plant dynamics into a suitable state-space form.

The state equations are defined as follows:

$$\begin{aligned}
\mathbf{d}\dot{x}_1(t) &= G_1(\mathbf{d}q_w(t) - \mathbf{d}q_v(t)) \\
\mathbf{d}\dot{x}_2(t) &= -t_2^{-1}\mathbf{d}x_2(t) - \frac{G_2}{t_2}(\mathbf{d}q_w(t) - \mathbf{d}q_v(t)) \\
\mathbf{d}\dot{x}_3(t) &= -2t_1^{-1}\mathbf{d}x_3(t) + \mathbf{d}x_4(t) + G_3\mathbf{d}q_w(t) \\
\mathbf{d}\dot{x}_4(t) &= -(t_1^{-2} + 4p^2T^{-2})\mathbf{d}x_3(t)
\end{aligned} \tag{2}$$

and the output (water level) is

$$\mathbf{d}y_p(t) = \mathbf{d}x_1(t) + \mathbf{d}x_2(t) + \mathbf{d}x_3(t) \tag{3}$$

If we define $\mathbf{d}x_p(t) \triangleq [\mathbf{d}x_1 \ \mathbf{d}x_2 \ \mathbf{d}x_3 \ \mathbf{d}x_4]^T$, the dynamics of the steam generator system can then be reduced to the following state-space equations:

$$\begin{aligned}
\mathbf{d}\dot{x}_p(t) &= A_p\mathbf{d}x(t) + B_p\mathbf{d}q_w(t) + F_p\mathbf{d}q_v(t) \\
\mathbf{d}y_p(t) &= C_p\mathbf{d}x(t)
\end{aligned} \tag{4}$$

where A_p , B_p , and C_p matrices are given as:

$$A_p = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & a_{43} & 0 \end{bmatrix}, \quad B_p = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ 0 \end{bmatrix}, \quad F_p = \begin{bmatrix} d_1 \\ d_2 \\ 0 \\ 0 \end{bmatrix}, \quad C_p = [1 \quad 1 \quad 1 \quad 0] \quad (5)$$

where,

$$a_{22} = -\mathbf{t}_2^{-1}, \quad a_{33} = -2\mathbf{t}_1^{-1}, \quad a_{34} = 1, \quad a_{43} = -(\mathbf{t}_1^{-2} + 4\mathbf{p}^2 T^{-2}), \quad b_1 = G_1, \quad b_2 = -G_2 \mathbf{t}_2^{-1}, \\ b_3 = G_3, \quad d_1 = -G_1, \quad d_2 = \frac{G_2}{\mathbf{t}_2}. \quad (6)$$

The approximate linearized model can be given by,

$$\begin{aligned} \mathbf{d}\dot{x}_p(t) &= A_0 \mathbf{d}x_p(t) + B_0 \mathbf{d}u_p(t) + F_0 \mathbf{d}w(t) \\ \mathbf{d}y_p(t) &= C_0 \mathbf{d}x(t) \end{aligned} \quad (7)$$

where $\mathbf{d}u_p(t) \stackrel{\Delta}{=} \mathbf{d}q_w(t)$ and $\mathbf{d}w(t) \stackrel{\Delta}{=} \mathbf{d}q_v(t)$.

In the subsequent derivations \mathbf{d} 's will be removed for clarity. We will write the system equations as:

$$\begin{aligned} \dot{x}_p(t) &= A_0 x_p(t) + B_0 u_p(t) + F_0 w(t) \\ y_p(t) &= C_0 x_p(t) \end{aligned} \quad (8)$$

where x_p, q_w, q_v , and y_p are the variations in plant state, feed-water flow-rate, steam flow-rate, and the system output.

One of the objectives is to design a state feedback controller so that the system is internally stable and its output (actual water level) asymptotically tracks the reference input (desired water level). This output tracking is achieved using a dynamic compensator through the introduction of a vector q defined below.

$$\dot{q} = y_p - y_r = C_0 x_p - y_r \quad (9)$$

where y_r is the reference input.

The state-space equations for the augmented system may be given by,

$$\begin{aligned} \dot{x}_{pq}(t) &= A_{pq} x_{pq}(t) + B_{pq} u_{pq}(t) + F_{pq} w(t) + H_{pq} y_r \\ y_{pq}(t) &= C_{pq} x_{pq}(t) \end{aligned} \quad (10)$$

where, $x_{pq} = [x_p \quad q]^T$, $y_{pq} = y_p$ and

$$A_{pq} = \begin{bmatrix} A_0 & 0 \\ C_0 & 0 \end{bmatrix}, B_{pq} = \begin{bmatrix} B_0 \\ 0 \end{bmatrix}, C_{pq} = [C_0 \quad 0], F_{pq} = \begin{bmatrix} F_0 \\ 0 \end{bmatrix}, \text{ and } H_{pq} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad (11)$$

3. Controller Design

Output Tracking:

To minimize the effects of parameter approximation one leads to minimize the cost functional,

$$J_1 = \int_{t_0}^{\infty} [x_{pq}^T(t) Q_1 x_{pq}(t) + u_{pq}^T(t) R_1 u_{pq}(t)] dt \quad (12)$$

where Q_1 and R_1 are constant weighting matrices that must be selected by the designer. The constant state weighting matrix Q_0 is selected to be symmetric and at least positive semi-definite and the control weighting matrix R_1 is selected to be symmetric and positive definite. Under these assumptions the value of J_1 is nonnegative.

The optimal control vector $u_{pq}(t)$ is generated from the state perturbation $x_{pq}(t)$ by a linear constant gain feedback

$$u_{pq}(t) = -Kx_{pq}(t) \quad (13)$$

where K is a constant feedback gain matrix given by

$$K = R_1^{-1} B_{pq}^T P_1 \quad (14)$$

and P_1 is a constant symmetric positive definite matrix which is the solution of the algebraic matrix Riccati equation,

$$P_1 A_{pq} - A_{pq}^T P_1 - Q_1 + P_1 B_{pq} R_1^{-1} B_{pq}^T P_1 = 0 \quad (15)$$

Then we can show that

$$u_{pq} = -Kx_{pq}(t) = -K_1 x_p - K_2 q \text{ where } K = [K_1 \quad K_2] \quad (16)$$

The existence and uniqueness of solution for the above equation are guaranteed by the following assumptions:

1. (A_{pq}, B_{pq}) is a controllable pair,.
2. $(A_{pq}, Q_1^{1/2})$ is an observable pair.

Under these assumptions the closed loop system

$$\dot{x}_{pq}(t) = (A_{pq} - B_{pq}K)x_{pq}(t) + F_{pq}w(t) + H_{pq}y_r \quad (17)$$

is asymptotically stable. This implies $\dot{q} \rightarrow 0$ as $t \rightarrow \infty$ and $y_p \rightarrow y_r$.

The complete system with tracking controller can be represented by the following block diagram.

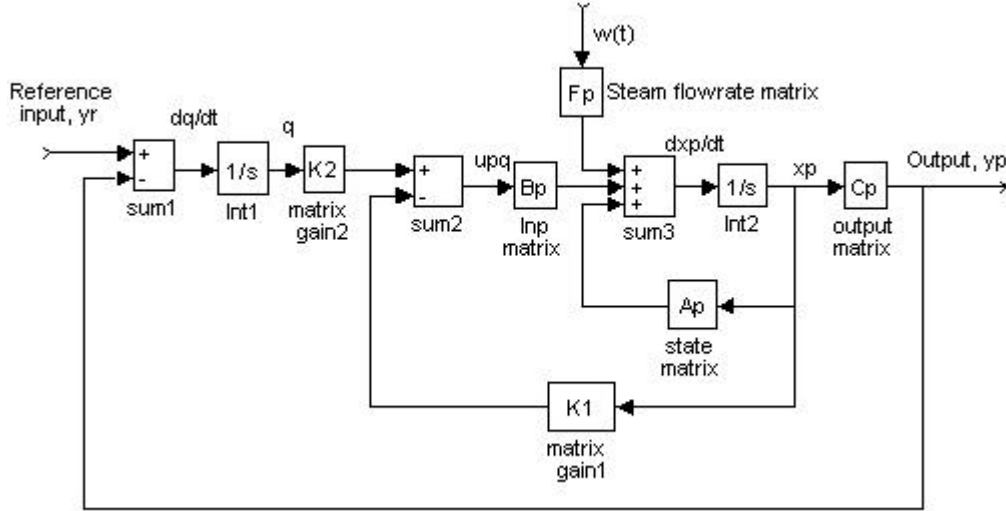


Fig. 1. Block diagram of the control system for output tracking

The dynamic parameters with respect to operating power linearized at different power level [14] is shown in the table below:

Table 1

q_v (kg/s)	57.4	180.8	381.7	660	1435
P (%power)	5	15	30	50	100
G_1	0.058	0.058	0.058	0.058	0.058
G_2	9.63	4.46	1.83	1.05	0.47
G_3	0.181	0.226	0.310	0.215	0.105
t_1	41.9	26.3	43.4	34.8	28.6
t_2	48.4	21.5	4.5	3.6	3.4
T	119.6	60.5	17.7	14.2	11.7

The elements of A_p , and B_p matrices are graphed below to show their variations with operating power.

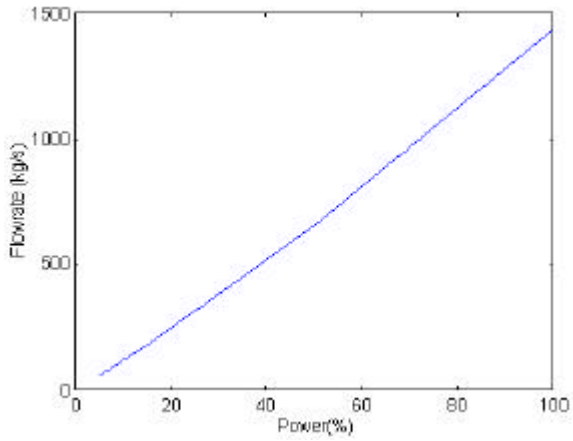


Fig. 2. Steam flowrate variation with power

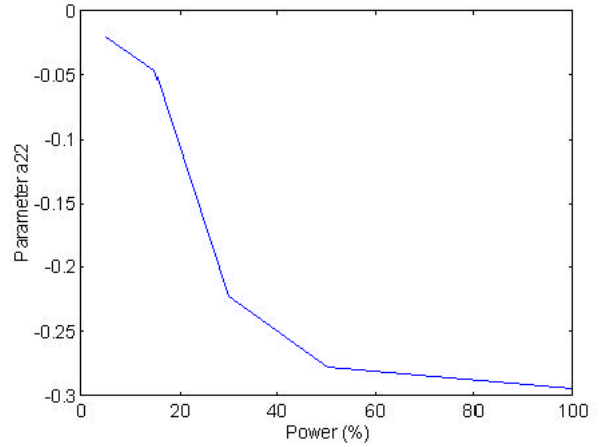


Fig. 3. Variation of element a22 with power

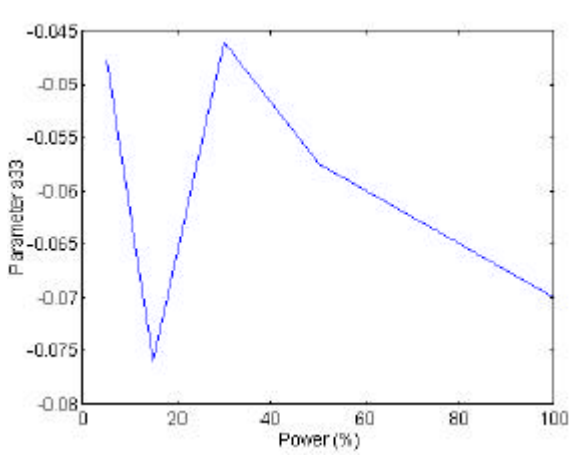


Fig. 4. Variation of element a33 with power

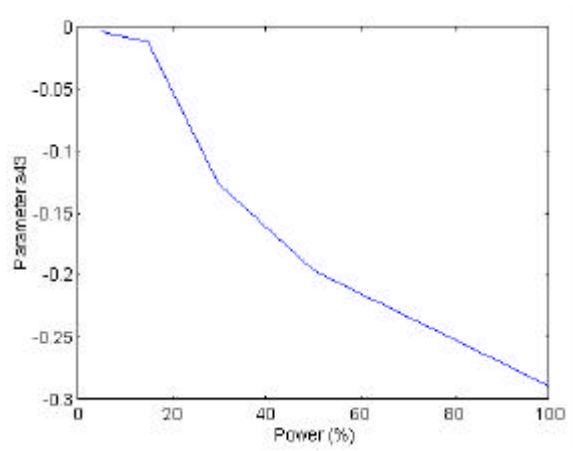


Fig. 5. Variation of element a43 with power

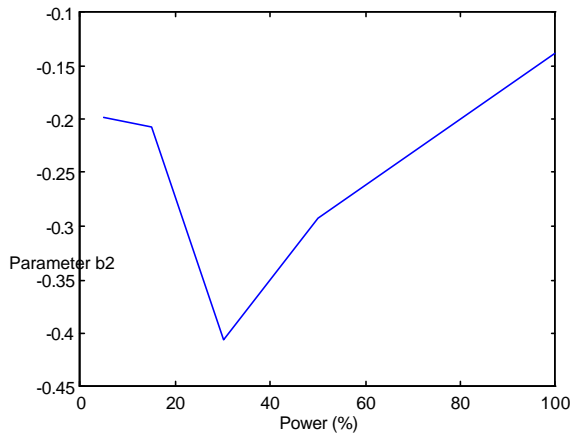


Fig. 6. Variation of element b2 with power

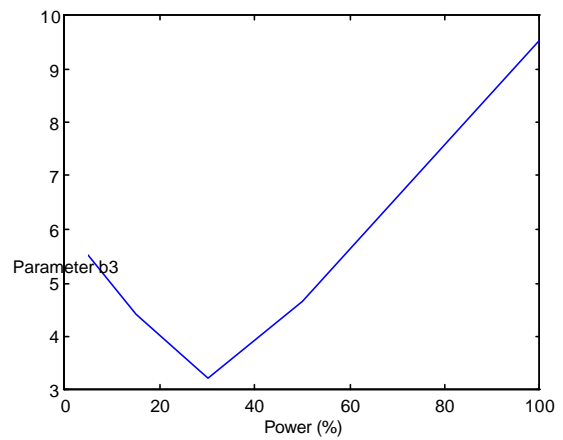


Fig. 7. Variation of element b3 with power

4. Simulation Results

We used a linear parameter varying model of UTSG of which parameters depends on the reactor power level. The model is linearized over four regions. The regions are divided according to operating power as: Region I for $0\% \leq \text{power} \leq 15\%$, Region II for $15\% \leq \text{power} \leq 30\%$, Region III for $30\% \leq \text{power} \leq 50\%$, and Region IV for $50\% \leq \text{power} \leq 100\%$. Over each region the elements of model matrices are assumed to vary linearly. These elements over each of these regions are obtained from the graph shown in figures 3, 4, 5, 6, and 7. The simulation studies are carried out for instances when the actual plant parameters drift from the linearized model parameters. It can be seen that variation of element b_2 of B_p matrix has the most destabilizing effect on the system response. This is because it contains the term G_2 and t_2 whose value determine the non-minimum phase characteristic of the steam generator. The simulation results are displayed in order to show the effects of these two parameters particularly on the effectiveness of the controller and the quality of the system response. As the power of operation becomes higher the effect of perturbation of parameters becomes less. For each simulation, a step change of 10% in power is considered after the reactor has passed 100 seconds at steady-state condition. Since the power of operation is directly proportional to the steam flow-rate, a step change of 10% of steam flow-rate is taken as the load demand. We also assume that water flow-rate and steam flow-rate are equal before the change in power demand occurs, that means the change in water level is zero. Figs. 8 and 9 display the steam generator water level change, and water and steam flow-rate change respectively when the reactor operating power is at 5% and no parameter perturbation is assumed. Figs. 10 and 11 portray the system response at the same power of operation (5%) but the parameters G_2 and t_2 are perturbed such that the element b_2 is increased by 3%. A substantial change in system behavior is noticed due to the drift of system parameter, such as larger overshoot and more oscillations before settling down to the desired value. Even when the parameter drifts, the controller is capable of maintaining the change in water level to zero and hence the water flow-rate change equals the steam flow-rate change. If the element b_2 drifts further the controller may not be robust enough to maintain the water level to the desired value. So we can say that the controller is robust enough to maintain the desired system response if the parameter perturbation lies within certain bounds. A similar phenomenon is also observed when the operating power is at 10%. For comparison, we have displayed the simulation results for 3% parameter perturbation at both 5% and 10% operating power (Figs. 10, 11, 12, and 13). At the 10% operating power the response is better because the water level settles down to zero level faster at the cost of increased overshoot which is expected. We can conclude that the water level oscillations tend to reduce to zero level faster but the overshoot becomes larger as the operating power increases. A case when the parameter perturbation is large enough to cause sustained oscillations at 10% power is depicted in Fig. 14. For our model, these sustained oscillations occur when the parameter drifts by 5%. We have further demonstrated in Figs. 15 and 16 that at 30% operating power a similar transient behavior (compared to the cases of 5% and 10% operating power) can be achieved at larger (30%) parameter perturbation. If we compare the responses portrayed in Figs. 16 and 17, it is noticed that water level convergence to zero level is much quicker at the operating power of 50% than at 30% for the same perturbation amount.

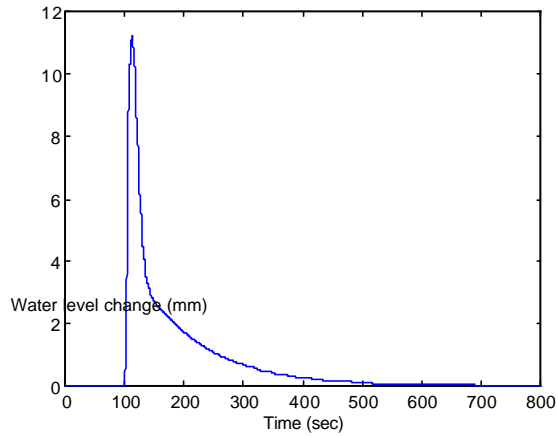


Fig. 8. Water level Change at 5% power

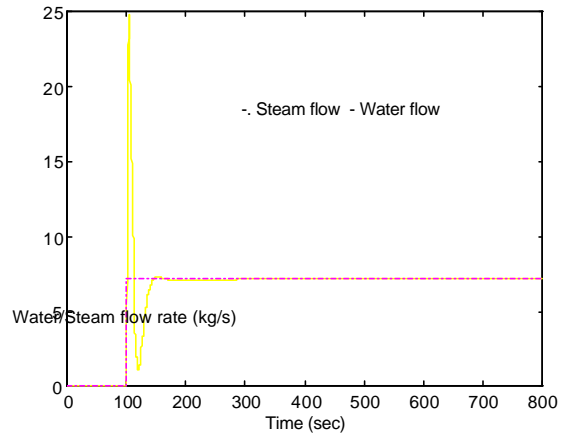


Fig. 9. Water/Steam flowrate change at 5% power

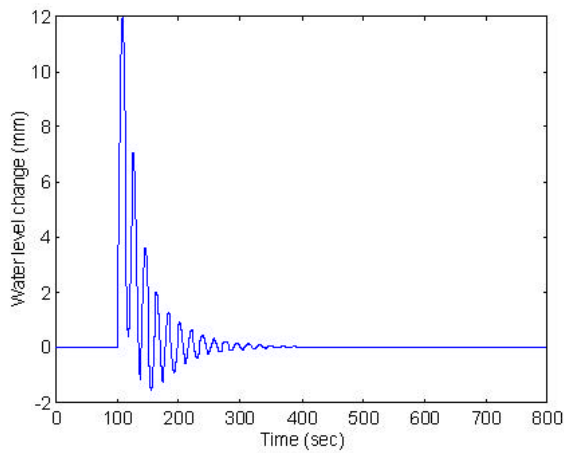


Fig. 10. Water level change at 5% power with 3% perturbation

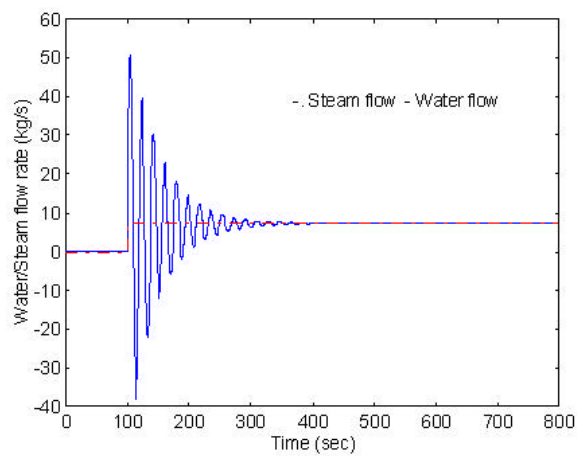


Fig. 11. Water/Steam flowrate change at 5% power with 3% perturbation

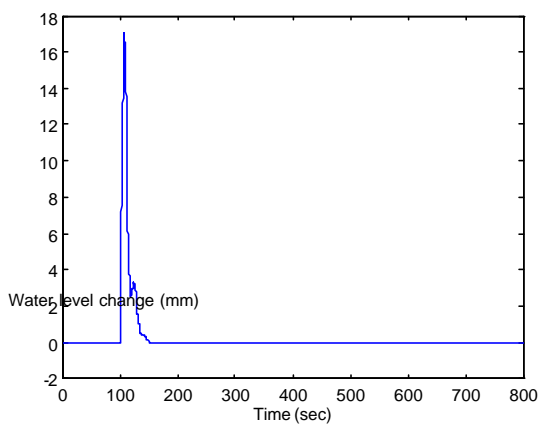


Fig. 12. Water level change at 10% power with 3% perturbation

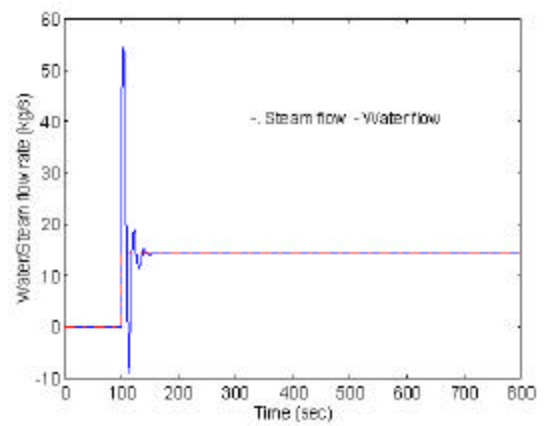


Fig. 13. Water/Steam flowrate change at 10% power with 3% perturbation

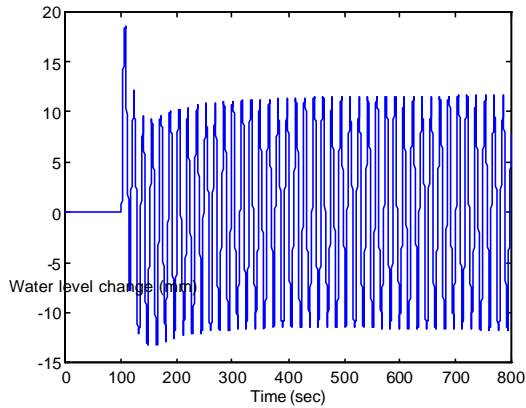


Fig. 14. Water/Steam flowrate change at 10% power with 5% perturbation

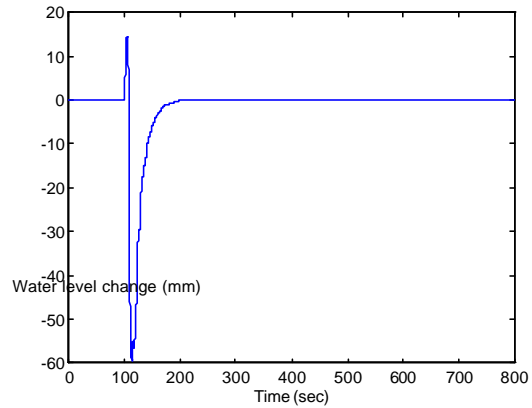


Fig. 15. Water level change at 30% power with 30% perturbation

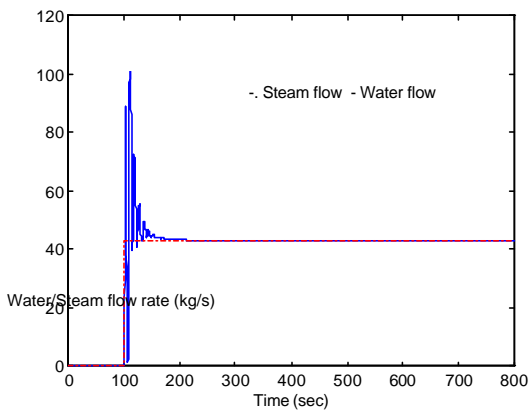


Fig. 16. Water/Steam flowrate change at 30% power with 30% perturbation

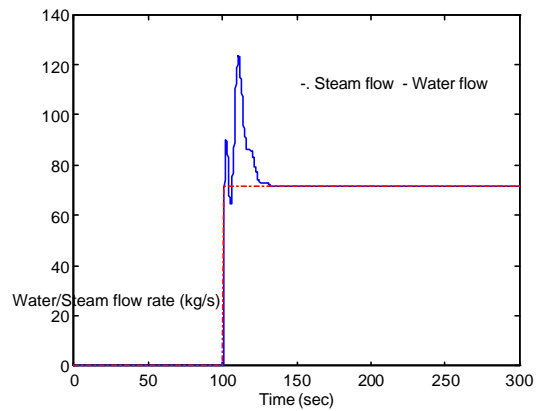


Fig. 17. Water/Steam flowrate change at 50% power with 30% perturbation

5. Conclusions

A robust tracking controller is developed for a UTSG water level control system of a nuclear reactor. Automatic control of the water level in the steam generator from 0% to 100% of the load, and when large-scale perturbation occurs, is assured. It is shown that the tracking error can be reduced to zero if the parameter perturbation is bounded within a certain value. Simulation results are provided for some cases to validate the effectiveness of the controller. The control system matched the water flow to the steam flow, under various operating conditions and parameter uncertainties. It may be concluded from this that the control system enables the level to be kept within limits and that the transient phenomenon are overcome within reasonably short time. Further studies are needed to develop an adaptive state and parameter estimator for the UTSG system.

Work on developing such an adapter has started and is not completed at this time. The steps towards developing a state estimator or observer is presented in the appendix.

References:

- [1] Kothare, M. V., Mettler, B., and Morari, M., "Linear Parameter Varying Model Predictive Control for Steam Generator Level Control," *Computers Chem. Engng.*, Vol. 21, 1997, pp. S861-S866.
- [2] Kothare, M. V., Mettler, B., Morari, M., and Bendotti, P., "Level Control in the Steam Generator of a Nuclear Power Plant," *IEEE Trans. Control System Technology*, Vol. 8, No. 1, Jan. 2000, pp. 55-69.
- [3] Menon, S. K. and Parlos, A. G., "Gain-Scheduled Nonlinear Control of U-Tube Steam Generator Water Level," *Nuclear Science and Engineering*, 111, 294-308 (1992).
- [4] Na, M. G., and No, H. C., "Design of model reference adaptive control system for steam generators," *Nuclear Engineering and Design*, 122 (1990), 301-311.
- [5] Na, M. G., "Design of steam generator water level controller via the estimation of flow errors," *Ann. Nuclear Energy*, Vol. 22, No. 6 pp. 367-376, 1995.
- [6] Na, M. G., and No, H. C., "Design of adaptive observer-based controller for the water level of steam generator," *Nuclear Engineering and Design*, 135 (1992) 379-394.
- [7] Schmitendorf, W. E., "Designing Stabilizing Controllers for Uncertain Systems Using the Riccati Equation Approach," *IEEE Trans. Automatic Control*, Vol. 33, No.4, pp. 376-379, April 1988.
- [8] Peterson, I. R., "A Riccati Equation Approach to the Design of Stabilizing Controllers and Observers for a Class of Uncertain Linear Systems," *IEEE Trans. Automatic Control*, Vol. AC-30, No. 9, pp. 904-907, September, 1985.
- [9] Schmitendorf, W. E. and Barmish, B. R., "Robust Asymptotic Tracking for Linear Systems with Unknown Parameters," *Automatica*, Vol. 22, No. 3, pp. 355-360, 1986.
- [10] Schmitendorf, W. E., "Design of Observer-Based Robust Stabilizing Controllers," *Automatica*, Vol. 24, No. 5, pp. 693-696, 1988.
- [11] Athans, M., "The Role and Use of the Stochastic Linear-Quadratic-Gaussian Problem in Control System Design," *IEEE Trans. Automatic Control*, Vol. AC-16, No. 6, December 1971.
- [12] Ioannou, P. A. and Sun, J., "*Robust Adaptive Control*," Prentice-Hall, Inc., 1996.
- [13] Sage, A. P. and White, C. C., "Optimum systems control," Prentice-Hall, Inc., 1977.
- [14] Kim, M. K., Shin, M. H., and Chung, M. J., "A gain-scheduled L_2 control to nuclear steam generator water level," *Annals of Nuclear Energy* 26 (1999) 905-916.

Appendix

Observer Design:

The implementation of the controller requires that the full state of the system is available. We like to estimate the states from the output measurements. By Kalman filter the optimal estimate $\hat{x}_p(t)$ of $x_{pq}(t)$ can be generated by

$$\dot{\hat{x}}_p(t) = A_0 \hat{x}_p(t) + B_0 u_{pq}(t) + L[y_{pq}(t) - C_0 \hat{x}_p(t)]$$

The filter gain matrix L is given by

$$L = P_2 C_0^T R_2^{-1}$$

where P_2 is the solution of the algebraic Riccati equation

$$A_0 P_2 + P_2 A_0^T + Q_2 - P_2 C_0^T R_2^{-1} C_0 P_2 = 0.$$

The observer state equation can then be reduced to

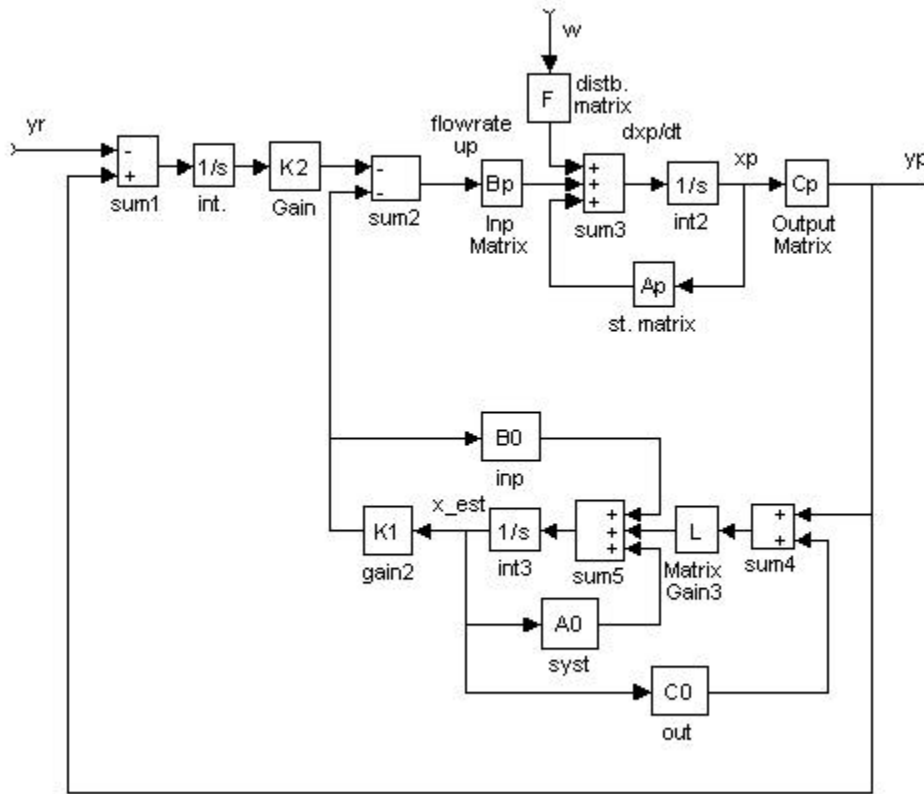
$$\dot{\hat{x}}_p(t) = (A_0 - B_0 K_1 - L C_0) \hat{x}_p + L C_p x_p$$

where K_1 is the gain required ensure output tracking, derived before.

The overall system will consist of the following equations

$$\begin{aligned} \dot{x}_{pq}(t) &= A_{pq} x_{pq}(t) + B_{pq} u_{pq}(t) + F_{pq} w(t) + H_{pq} y_r \\ \dot{\hat{x}}_p(t) &= (A_0 - B_0 K_1 - L C_0) \hat{x}_p(t) + L C_p x_p(t) \\ y_{pq}(t) &= C_{pq} x_{pq}(t) \text{ and} \\ u_{pq}(t) &= -K_1 \hat{x}_p(t) - K_2 q(t) \end{aligned}$$

The complete system may be represented by the block diagram shown below.



Overall System:

We define states as: $x = [x_{pq} \quad \hat{x}_p]^T$

Then the system equations may be written as

$$\begin{aligned} \dot{x} &= Ax + Ew + Gy_r \\ y &= Cx \end{aligned}$$

where

$$A = \begin{bmatrix} A_{pq} & -B_{pq}K_1 \\ LC_{pq} & (A_{pq} - B_{pq}K_1 - LC_{pq}) \end{bmatrix}, \quad E = \begin{bmatrix} F_{pq} \\ 0 \end{bmatrix}, \quad G = \begin{bmatrix} H_{pq} \\ 0 \end{bmatrix}, \quad C = [C_{pq} \quad 0]$$

Attachment IV
Task 1.4 Requirements-Driven Control System Design

TABLE OF CONTENTS

1.	Introduction.....	1
2.	Control Engine Concept.....	1
3.	Control Engine Implementation.....	2
3.1	Selection of Design Requirements Related to Control System Performance.....	2
3.2	Implementation of Requirements in Mathematical Form.....	3
3.3	Development of a Control Algorithm Library.....	3
3.4	Development and Validation of Plant Models	4
3.5	Automated Control Design Development.....	4
3.6	Development of Control Architectures	4
3.7	Control Design Implementation	4
3.8	Development of Diagnostics Methods to Update the Plant Model	5
4.	Application to a Simplified Steam Generator Model	5
4.1	Nomenclature for Simplified Steam Generator Model	8
4.2	Subscripts	8
5.	Application to a Full-Scope PWR Simulator.....	8
6.	Literature Review for Requirements-Driven Control Design	14
6.1	Introduction.....	14
6.2	The scope of automatic, system-wide control.....	15
6.2.1	Continuous-Time Control	15
6.2.2	Discrete-Event (logic) Control	15
6.2.3	Hybrid control	16
6.2.4	Plant System Classification Method	18
6.2.5	Control Methods that Apply to Automated Hybrid Control	21
6.3	Need for automated requirements capture.....	22
6.3.1	Capture and Design Process	23
6.3.2	Potential mechanisms for capturing requirements	24
6.3.3	Categories of information that constitute control system requirements	25
6.4	References	27
6.4.1	General	27
6.4.2	References From ORNL/TM9500	28
6.4.3	General Reference for Hybrid Control and Intelligent Agents	31

LIST OF FIGURES

Figure 1. Schematic diagram of the automated control design process	2
Figure 2. Schematic diagram of simplified steam generator model	5
Figure 3. The control engine automatically determines the optimal control algorithm and parameters that satisfy the time response requirements.....	7
Figure 4. Simulation of a degraded condition. Inlet flow is limited to 150%, 125%, or 115%. Control engine diagnoses the condition and adjusts control parameters.....	7
Figure 5. The Control Engine automatically calculates the level control strategy that satisfy all of the performance requirements	9
Figure 6. The Control Engine calculates the optimal control strategy for multiple postulated transients and performance requirements.....	10
Figure 7. Power-Dependent Optimal Controller.....	11
Figure 8. Example of Control Engine decision following a failed feedwater pump. Control parameters are re-optimized to minimize overshoot for a postulated power step transient	12
Figure 9. Example of Control Engine decision after diagnosis of an incipient failure of the steam flow sensor. Control parameters are re-optimized to prevent instability.....	13
Figure 10. Hybrid control consists of a mixture of continuous and discrete (logic) control functions that coordinates operation of an entire system over a wide range of conditions and states.....	17
Figure 11. A typical subsystem component may contain both continuously variable and discretely addressable elements. An example might be a feedwater subsystem for which there can be continuous (feedback) control to valve position and pump speed to maintain pressure and flow and binary control over lubricating pumps, cooling, and circuit breaker state	17
Figure 12. Hierarchical hybrid control applied to nuclear power plant systems	20
Figure 13. The automated plant control system has boundaries with all major plant subsystems including human operators and maintainers.....	20
Figure 14. Capturing design requirements and converting them into a final working control system is a part of a larger multi-step process for producing a comprehensive automatic control system.....	23

LIST OF TABLES

Table 1. Major plant system classification by relation to plant output.....	18
Table 2 Listing of representative control methods for both continuous and discrete processes.	21
Table 3 The critical steps to creating a requirements-driven control design.	22
Table 4. Several distinct disciplines contribute to automated control requirements capture and subsequent conversion to control system.....	24
Table 5. Comparison of traditional computer programming with intelligent agents.....	25
Table 6. Categories of information for the design requirement capture process.....	26

1. INTRODUCTION

Nuclear plants of the 21st century will employ higher levels of automation and fault tolerance to increase availability, reduce accident risk, and lower operating costs. Key developments in control algorithms, fault diagnostics, fault tolerance, and communication in a distributed system are needed to implement the fully automated plant. Equally challenging will be integrating developments in separate information and control fields into a cohesive system, which collectively achieves the overall goals of improved performance, safety, reliability, maintainability, and cost-effectiveness. Under the Nuclear Energy Research Initiative (NERI), the U. S. Department of Energy is sponsoring a project to address some of the technical issues involved in meeting the long-range goal of 21st century reactor control systems. This project, "A New Paradigm for Automated Development Of Highly Reliable Control Architectures For Future Nuclear Plants," involves researchers from Oak Ridge National Laboratory, University of Tennessee, and North Carolina State University.

This paper documents a research effort to develop methods for automated generation of control systems that can be traced directly to the design requirements. Our final goal is to allow the designer to specify only high-level requirements and stress factors that the control system must survive (e.g. a list of transients, or a requirement to withstand a single failure.) To this end, the "control engine" automatically selects and validates control algorithms and parameters that are optimized to the current state of the plant, and that have been tested under the prescribed stress factors. The control engine then automatically generates the control software from validated algorithms.

Examples of stress factors that the control system must "survive" are: transient events (e.g., set-point changes, or expected occurrences such a load rejection,) and postulated component failures. These stress factors are specified by the designer and become a database of prescribed transients and component failures. The candidate control systems are tested, and their parameters optimized, for each of these stresses. Examples of high-level requirements are: response time less than xx seconds, or overshoot less than xx% ... etc. In mathematical terms, these types of requirements are defined as "constraints," and there are standard mathematical methods to minimize an objective function subject to constraints. Since, in principle, any control design that satisfies all the above constraints is acceptable, the designer must also select an objective function that describes the "goodness" of the control design. Examples of objective functions are: minimize the number or amount of control motions, minimize an energy balance... etc.

2. CONTROL ENGINE CONCEPT

The vision of this control-system design process is described schematically in Figure 1. Our final goal is to develop methods to implement reliable control systems that automatically satisfy all the design requirements. As indicated by the arrow in Figure 1, this is accomplished by automating the design process to arrive at a plant implementation directly from the design requirements.

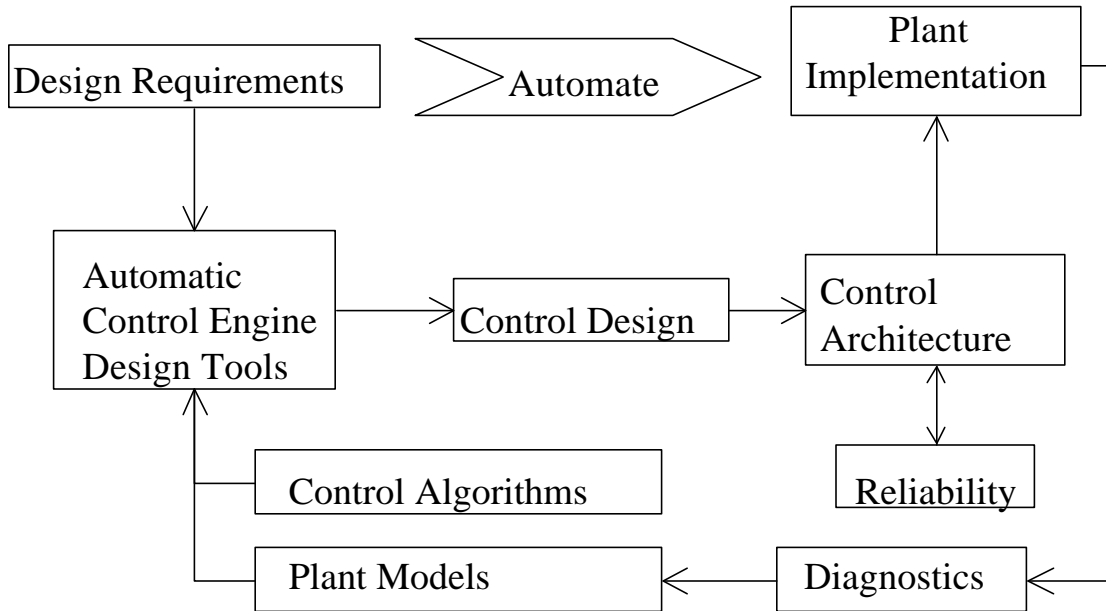


Figure 1. Schematic diagram of the automated control design process

The implementation of the methodology takes the bottom path of Fig. 1. For this implementation, the design requirements are fed into a control engine, which uses a library of control algorithms and validated plant models to arrive at the control design using an iterative optimization process. The control design is then implemented using validated control architectures, which are tested automatically to guarantee that the reliability requirements are met. Finally, during the lifetime of the plant, the plant model is maintained up to date (e.g, updated with component failures or mode changes) by an on-line diagnostics system.

3. CONTROL ENGINE IMPLEMENTATION

The automation of this design methodology requires the following steps:

3.1 Selection of Design Requirements Related to Control System Performance

The first step is to review the design requirements for applicability to the problem at hand. The automated design system will deal only with those design requirements that relate directly to the “performance” of the control system. The discarded requirements will have to be addressed separately in a manual fashion.

An example requirement is to specify that all the equipment must operate using a power supply of 110 Volts and 60 Hz. This is clearly a valid requirement, which must be specified when purchasing the control hardware, but is not directly related to control performance; thus, these types of requirements are not included in the automated control engine requirement set.

An example of a valid requirement is to specify that the control system must control the reactor without scram for a particular transient. For example, we could specify that a feedwater controller should automatically handle a loss of a single heater without resulting

in a reactor scram. Clearly, these types of requirements are directly related to control performance and must be included in the automated control engine requirement set.

The selection of “proper” design requirements is clearly one of the more important steps in the design process. It is also clear that some requirements will not be as clear cut as the examples above. In such cases, there must be an iterative process between the control engineers and the requirement “providers” to resolve differences. The clear advantage of our proposed automated control design process is that it frees the control engineers to concentrate on the high level goals and lets the computer do the tedious implementation work.

3.2 Implementation of Requirements in Mathematical Form

Once proper design requirements are identified, each must be documented in the form of a “subroutine.” This subroutine takes as input the results of a computer simulation and returns a value that indicates whether the particular requirement is satisfied. This documentation then becomes a mathematical description of the requirement, which is precise and reviewable. The set of subroutines that document all requirements then become a complete description of the control problem.

This step becomes in effect a second screening of the design requirements. If a design requirement cannot be written in the form a subroutine, it is probably not a proper control design requirement. The basic idea is that the requirement must be “testable.” In other words, given the results of a computer simulation, the control engineer must be able to determine unequivocally whether the requirement is satisfied or not; thus, a proper requirement should always be able to be documented in the form of a subroutine.

Clearly, to complete this second step, there must be an additional iterative process between the control engineers and the requirement “providers” until all requirements are documented in this mathematical form. This provides for yet another opportunity to review the completeness and appropriateness of the requirement set.

3.3 Development of a Control Algorithm Library

To automate the control system development, the control engine must have available a library of control algorithms. This library includes standard control algorithms such as proportional integral or advanced multivariate control schemes. The library may also include problem-specific algorithms; for example to control a steam-generator level, it would be logical to include a three-element controller in the library.

The control library can be as inclusive and sophisticated as desired. It may include problem-specific model-based controllers, neural networks, fuzzy controllers ... The only requirement is that those algorithms be parameterized; the control engine will select the optimal parameters for each algorithm and then pick the optimal algorithm so that the final design satisfies all requirements.

3.4 Development and Validation of Plant Models

Validated plant models are required to perform the computer simulations required for the automated control design. The plant models must be validated and must be kept up to date by the diagnostics system for the life of the plant.

3.5 Automated Control Design Development

The core of this research effort is the automated development of control designs that satisfy all the requirements. This is accomplished by using standard constrained minimization methods, where the requirements are the mathematical constraints, and the control engine minimizes an objective cost function that the designer specifies.

3.6 Development of Control Architectures

Our initial research addressing the control and information system architecture for future nuclear installations involves the development of functional requirements. The plant architecture that we envision can serve to establish an integration platform for functional capabilities and a distributed communications framework to support operations, maintenance, and engineering personnel at a 21st century nuclear power plant. The distributed network can provide the backbone to convey information from the data acquisition sources to the users and applications that process the information. The issues of architecture being addressed in this research are: (1) Provide a common, consistent interface to I&C systems; (2) Enable uniform, transparent access to distributed data sources; (3) Establish a computing environment that facilitates the integration of information and applications (e.g., diagnostics and control); (4) Define a system architecture that permits flexibility in implementation and expandability of functional capabilities; and (5) Define an approach to application support that lays the foundation for standardizing functions and interface conventions for the nuclear power industry.

Simply establishing network links among the various systems and installing workstations as network nodes addresses only part of the desired support for plant personnel at the of 21st century plant. A key goal of this research is to provide common functionality throughout the control and information system architecture. In this way, the efficiency of plant personnel in performing their tasks can be enhanced and the possibility of user error while interacting with multiple systems and data sources can be reduced. The proposed architecture concept is based on a layered approach with the capabilities and services of the application environment supporting the functionality of applications (e.g., diagnostics or controls).

3.7 Control Design Implementation

Once the automated control engine has identified a control design that meets all the requirements, the control design is implemented in the real-life plant using the validated control architectures.

3.8 Development of Diagnostics Methods to Update the Plant Model

Since the goal of this research is to maintain the design within its requirements for the life of the plant, it is imperative that the plant model be kept up to date. Thus, the implementation of these techniques requires the development of diagnostics methods that can identify component failures or degradations. Those component degradations should then be reflected in the plant model. The automated control engine is then implemented on-line with the updated models to guarantee that the original design requirements are still satisfied with the new plant configuration.

4. APPLICATION TO A SIMPLIFIED STEAM GENERATOR MODEL

As a demonstration of proof of principle, we have implemented this methodology using a standard off-the-shelf minimization algorithm. For this example, we have used the simplified steam generator model described schematically in Figure 2. This model is complex enough to be non-trivial, but simple enough to allow for a large number of fast calculations to define the proper strategies. The equations describing the dynamics of the steam generator are nonlinear:

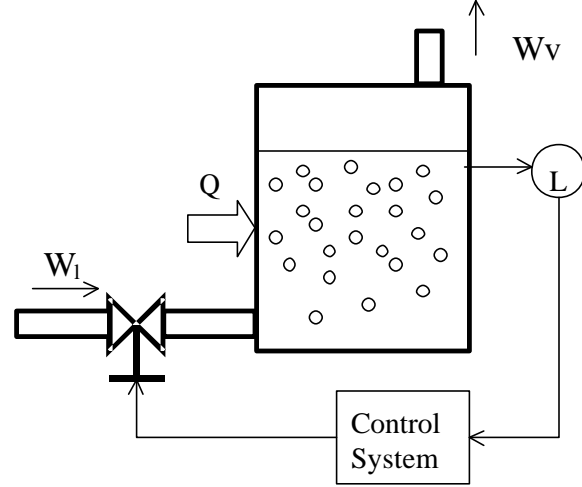


Figure 2. Schematic diagram of simplified steam generator model

$$\frac{dM}{dt} = W_l - W_v$$

$$\frac{dH}{dt} = Q + h_l W_l - h_v W_v$$

With the closure relations

$$\mathbf{a} = \frac{H - ALr_l h_l}{AL(\mathbf{r}_v h_v - \mathbf{r}_l h_l)} ; \quad \mathbf{r} = \mathbf{r}_l (1 - \mathbf{a}) + \mathbf{r}_v \mathbf{a} ; \quad L = \frac{M}{\mathbf{r}A} ; \quad W_v = \frac{W_{l0}}{\mathbf{a}_0} \mathbf{a}$$

For this proof-of-principle demonstration, the library of control algorithms includes only simple proportional or proportional-integral controllers. The control system can be described by the following equations

$$u = K_p (L_{set} - L) + K_i \int_0^t (L_{set} - L) dt$$

$$W_l = u \quad \text{if } 0 < u < u_{\max}$$

The maximum flow through the valve is limited by the full-open stem position, which introduces a hard nonlinearity. As design requirements, we selected the maximum 90% rise time for a step response, a maximum overshoot, and a maximum steady-state error. As objective cost function, we minimize the control effort, which is defined as

$$\text{Control Effort} = \int_0^t \left(\frac{du}{dt} \right)^2 dt$$

The control effort variable is proportional to the amount of work that the controller must perform. Typically, by minimizing this variable we obtain a minimum wear and tear of all control components and overall smoother performance.

The control engine immediately selected a proportional-integral controller with the optimized gain parameters that minimize the control actions while still satisfying the requirements. The control engine determined in all cases that a simple proportional control algorithm would not be able to satisfy the steady-state error requirement without an integral-control component, as expected.

Figure 3 shows an example response of the system for three different requirements. In all three cases, we have requested that the engine maintain an overshoot less than 2% for a 10% step demand in the level setpoint. The requirements are changed from case to case so that the response time is 5, 3, or 2 seconds respectively. The control engine automatically computes the optimal controller gains to satisfy those requirements and minimize the control effort. As it can be seen in Figure 3, the faster the system, the larger the control effort required.

Since the simplified steam generator model that we used is non-linear, it comes as no surprise that the optimized gain parameters are function of the operating conditions and Plant State. This feature has been used to demonstrate the adaptive nature of our control-design methodology. By running the control-engine and the diagnostic system in parallel with the real system, we are able to feed it the current operating conditions and the status of the failed components. Thus, the control-engine is able to determine on-line whether the original design requirements are met under the current operating conditions and Plant State. If the design requirements are not met, alternate control algorithms or parameters that meet those requirements are calculated. The alternate algorithms can be implemented on-line, or suggested to an operator, who makes the final decision about implementation.

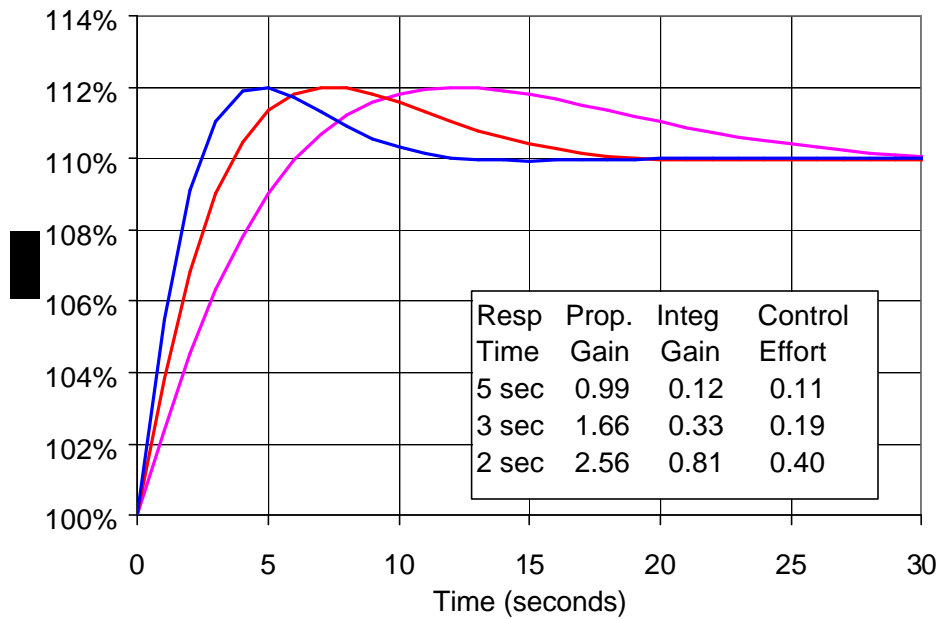


Figure 3. The control engine automatically determines the optimal control algorithm and parameters that satisfy the time response requirements.

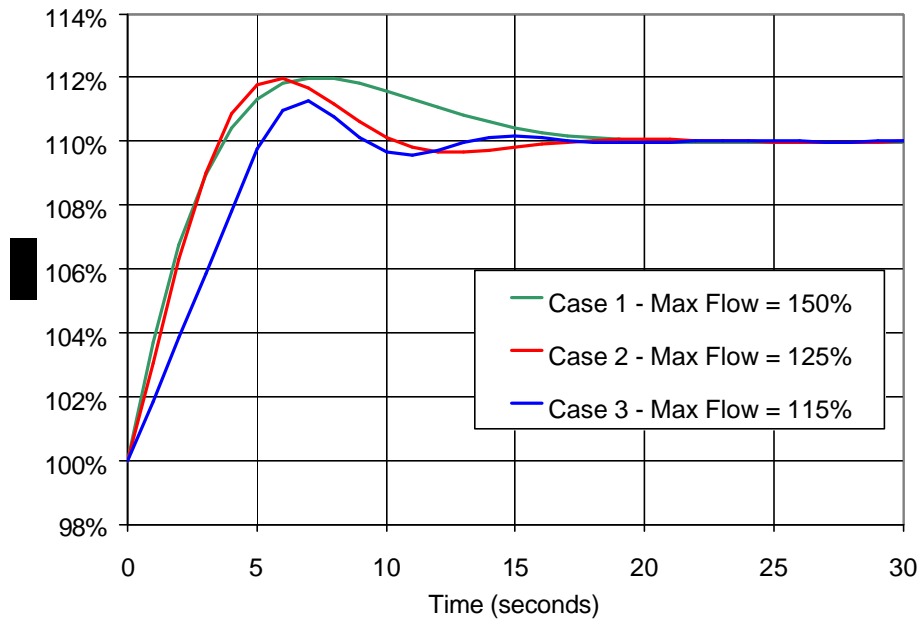


Figure 4. Simulation of a degraded condition. Inlet flow is limited to 150%, 125%, or 115%. Control engine diagnoses the condition and adjusts control parameters

Figure 4 presents an example of the use of the control engine methodology in an adaptive form during the life of the plant. For this example, we have assumed that some time after implementation, the available flow through the input valve is limited. An example of this limitation would be caused by cavitation on the valve, which would impose a new requirement to limit the flow. Another example of limited flow would be the loss of one of the upstream feed pumps. For the example in Fig. 4, we have assumed three cases, the original, a case where the maximum inlet flow is limited to 125% of nominal, and a third case where the maximum inlet flow is limited to 115%. For all this cases, we require an overshoot lower than 2% and a time response lower than 3 seconds.

The control engine is assumed to be running in diagnostics mode. For case 2 (125% maximum flow,) the control engine detects that the original time-response requirements are not met, and it calculates new optimal parameters that satisfy the requirements. For case 3 (115% maximum flow,) the control engine determines that the requirements cannot be satisfied because the minimum response time achievable is ~4 seconds. Even though the control system is capable of maintaining the level at 100%, it cannot satisfy the requirements under the specified 10% step response transient. For this case, the operator would be notified and there are two options: either (1) the requirements would need to be relaxed, or (2) the plant would need to be fixed to allow for higher inlet flow.

4.1 Nomenclature for Simplified Steam Generator Model

M	total mass
L	two-phase level
A	flow area
ρ	density
W	mass flow
α	void fraction
h	specific enthalpy
K	controller gain

4.2 Subscripts

l	liquid
v	vapor
max	maximum
set	setpoint
p	proportional
i	integral
0	initial condition

5. APPLICATION TO A FULL-SCOPE PWR SIMULATOR

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 developed in Task 3.1. 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 Figure 5 and Figure 6; 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 unusual 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

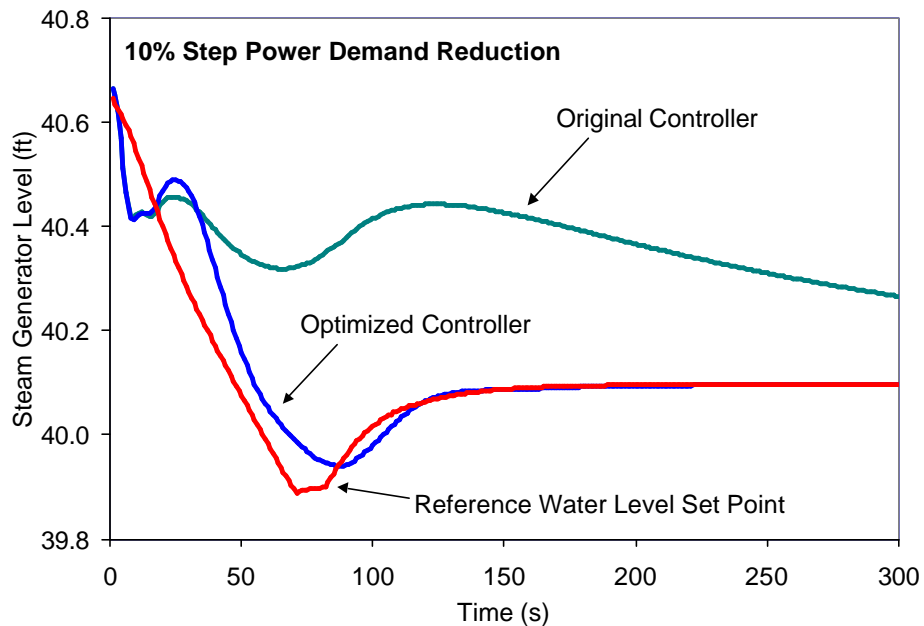


Figure 5. The Control Engine automatically calculates the level control strategy that satisfy all of the performance requirements

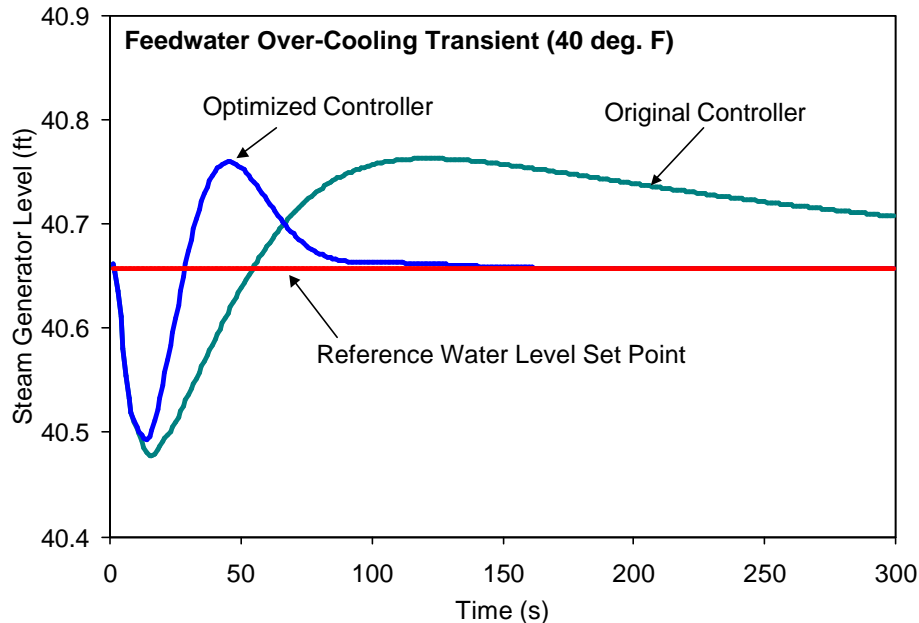


Figure 6. The Control Engine calculates the optimal control strategy for multiple postulated transients and performance requirements

Using a procedure similar to the one outlined above, we can automatically develop an optimized controller for our U-tube steam generator. Figure 7 shows one such controller. This is a typical three-element controller, where the steam generator level is controlled by setting up a feedback loop on the measured water level plus the mismatch between the steam flow and feedwater flows. By adding this flow mismatch in the three-element logic, these types of controllers can handle the well-known shrink and swell phenomenon, where the two-phase water level temporarily drops when additional cold feedwater is injected because it collapses steam voids.

Steam generator dynamics are non-linear and change significantly with operating power level as the void fraction changes. To compensate for those changes, the controller shown in Figure 7 was designed with power-dependent gains. In this way, the controller can be designed to be significantly more “aggressive” and have better performance at full power without losing stability at low power levels. As seen in Figure 7, our controller design provides a tradeoff between proportional, integral, and three-element gains as power changes. This results in greatly improved performance, especially at high powers where the steam generator operates most of the time.

What makes this controller unique, however, is not its performance characteristics, but the fact that it was designed completely automatically using the design engine of Task 1.4. In essence, we have captured the design requirements for this steam generator in the control engine and a new optimized controller can be calculated at any time in the life of the facility if conditions or equipment changes (e.g., component failures or replacements) occur.

$$\Delta FCV = k_p(P) \times \left[(L - L_{set}) + k_3(P) \times (w_l - w_s) \right] + k_i(P) \times \int (L - L_{set}) dt$$

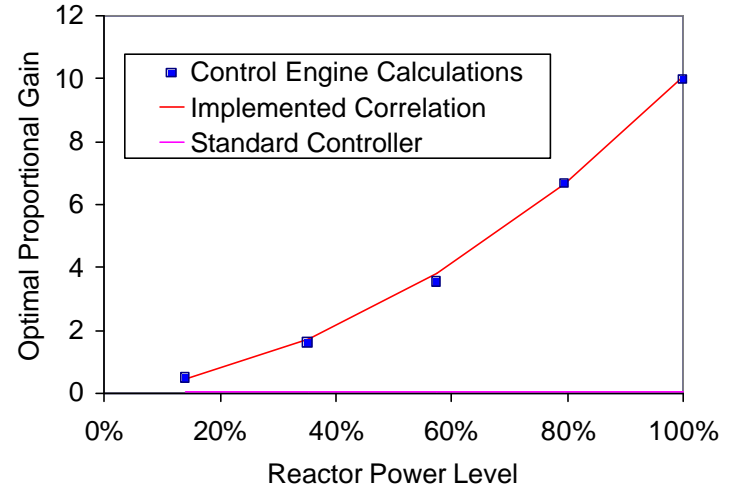
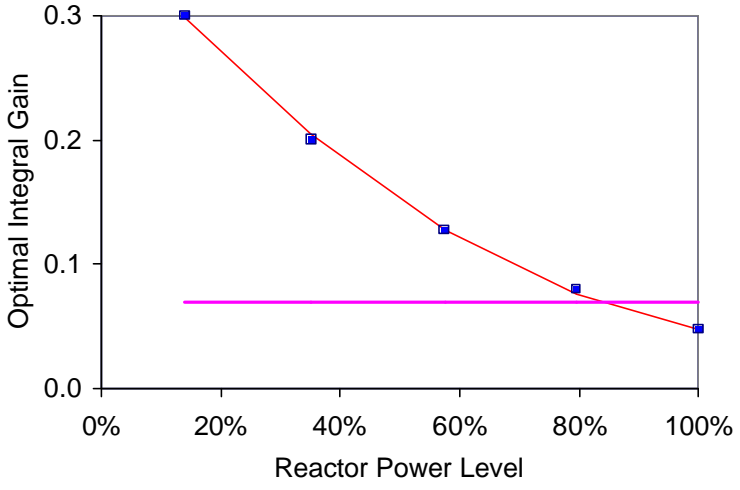
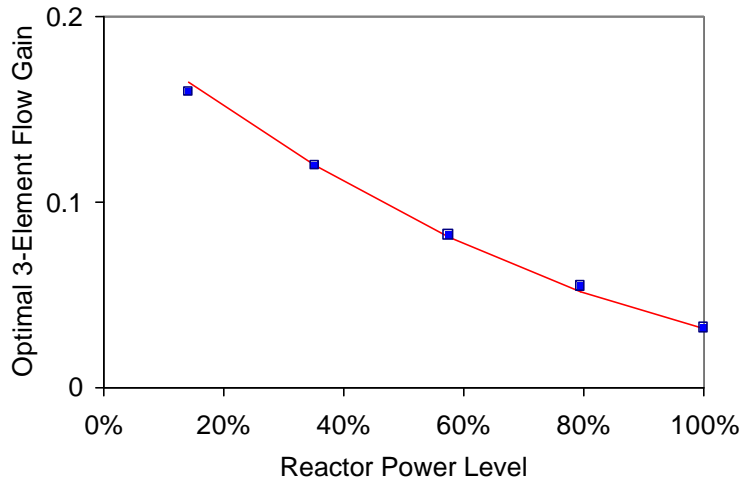


Figure 7. Power-Dependent Optimal Controller

Figure 8 and Figure 9 show examples of this feature. In the example of Figure 8, a feedwater pump is assumed failed, so that the maximum feedwater flow available to the controller is limited. The Control Engine automatically recognizes that, because of the maximum flow limitation, the original controller will produce a significant overshoot. The Control engine automatically suggests a new set of control parameters that are less aggressive and minimize the overshoot caused by the new plant condition. In the example of Figure 9, the diagnostics system has detected an incipient failure of a steam flow sensor. Then Control engine, then, evaluates the impact of this failure (by assuming completely failed) and concludes that if the sensor fails, the controller will become unstable. It then, automatically, re-computes optimized control parameters that do not use the steam flow sensor, and it suggests a re-optimized controller that will not be sensitive to the sensor failure should it occur. Note that this new controller is not as good as the original controller, but it is more fault-tolerant given the information we now know about the probability of failure of the sensor.

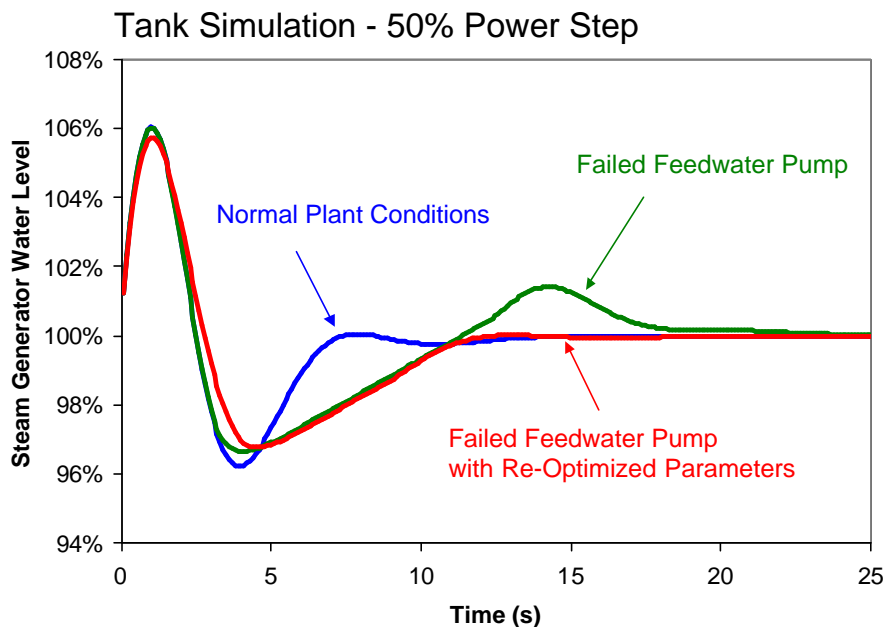


Figure 8. Example of Control Engine decision following a failed feedwater pump. Control parameters are re-optimized to minimize overshoot for a postulated power step transient

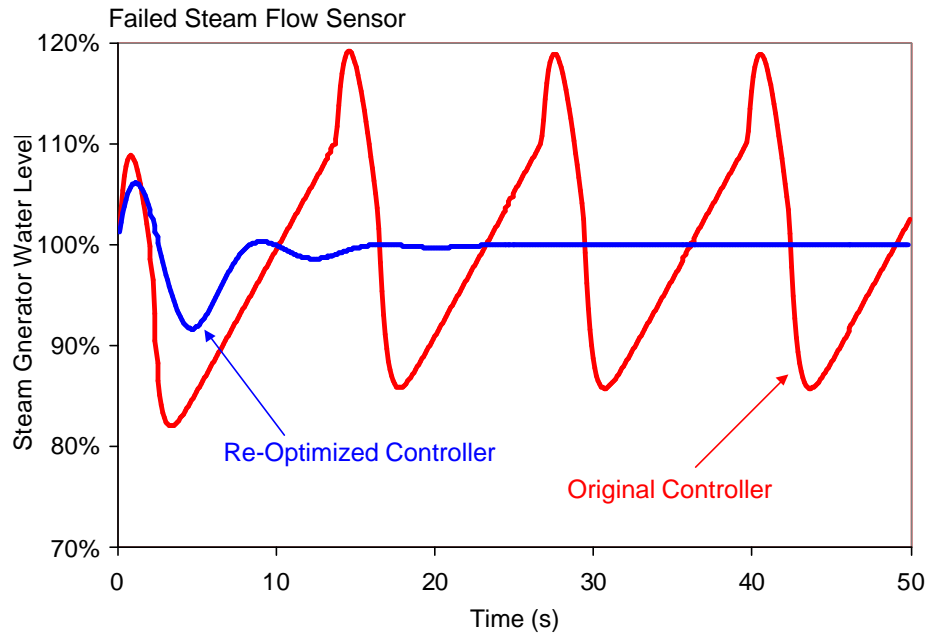


Figure 9. Example of Control Engine decision after diagnosis of an incipient failure of the steam flow sensor. Control parameters are re-optimized to prevent instability

6. LITERATURE REVIEW FOR REQUIREMENTS-DRIVEN CONTROL DESIGN

6.1 Introduction

Highly effective control systems can be realized by translating plant design requirements and specifications into specific control functions, behavior, and architecture. The proposition is to uniquely map design requirements to elements of the control system. Ultimately, every requirement can be precisely met by a thorough mapping. This control system design process can be automated, which leads to benefits of increased productivity and completeness.

The design process can be accelerated and the final product improved by automating the capture processes and allowing human designers to make high-level decisions. Specifying control system behavior and its architecture then becomes a direct embodiment of the requirements specification. Such automation is possible given the developments in computer science and control engineering over the last decade.

The theme of automating the process and its attendant benefits can extend from capturing the design requirements through designing the control system and into the resulting functionality of the control system itself. The increased thoroughness in specifying control system functions that results from a complete capture of the design requirements permits the development of highly automated control systems. The control system that emerges from this design process can have automated features that give it a span of control over a wide range of plant modes and off-normal conditions. These automatic control features will be consistent with the initial requirements.

The operational time horizon of a given plant may be 40 to 60 years because of the high initial cost of a nuclear power plant as well as siting difficulties. In that period, many repairs, replacements, upgrades, and modifications will be made to plant components. Some alterations require technical knowledge related to the intent in the original specification.

Accurately mapping directly from design requirements and system specification to a tractable document for control design also becomes an invaluable reference, preserving details for future plant replacements, upgrades and modifications. The concept is to not only store original specifications and design information but also continuously update diagnostic and modification-related information so that the historical database is maintained.

The resulting control system and plant system combination must be shown stable and able to perform under a variety of extreme transients and component failures. Proving that the final system will respond appropriately and consistently under all conceivable conditions is required whether the control design was accomplished by pencil-and-paper or highly automated computer effort. The capture process, therefore, should also include testing and validation requirements besides the customary system performance requirements.

For Generation IV nuclear power plants, there exists an opportunity to increase the capabilities of plant control systems and simultaneously reduce plant staffing needs and the risk of inappropriate human operator actions. Control designers have the opportunity to consider multiple operations scenarios, develop optimum strategies for problem resolution, and include those strategies as part of the control system automation.

By its actions or inaction, the plant control system can cause or avert challenging conditions, which may require intervention by the reactor safety system or other plant safety features. There exists a need in the control system for fault tolerance at several levels: (1) redundancy in hardware to directly compensate for physical component failure and (2) diagnostic and compensatory actions through algorithmic methods to detect and overcome faults that are more insidious. The latter need is the motivation for ensuring that the design requirements are completely mapped to the functionality of the control system.

Most control systems are designed as a rigid structure that is not able to choose control or diagnostic algorithms to suit changing tasks and conditions. Further, a lack of formalism does not allow designers and reviewers to verify how well the control commands and architecture link to the requirements. Usually, systems that generate diagnostic information target human operators and are not connected to automatic control systems. The integration of diagnostic and prognostic capability with hybrid control represents a major improvement in automatic control systems. The thoroughness of this effort is related to how well the designers can map the requirements to final design. The remainder of this report discusses the issues of integrating these concepts.

6.2 The scope of automatic, system-wide control

A power plant is a hierarchy of systems and states. At a high enough level for the overall plant, its fundamental states become (1) start up, (2) shutdown, (3) normal power operation, and (4) operation in degraded state. At lower levels in the system hierarchy, individual plant subsystems have similar states with a few additions: (1) start up, (2) shutdown, (3) standby, (4) normal operation, (5) refurbish/regenerate, and (6) degraded state. In the current generation of nuclear power plants, human operators maneuver the plant and its subsystems between these states. For Generation IV plants, the mission is to increase the degree of automatic response. To do so, requires the implementation of hybrid control schemes that permit simultaneous control and coordination of continuous-time and discrete-event plant subsystems. A brief description of these control schemes follows.

6.2.1 Continuous-Time Control

Continuous systems, which are characterized by differential or partial differential equations, are controlled in a proportioning manner to achieve a set-point value through feedback from proportionally measuring sensors. This well-known concept has been discussed with examples in the previous report [ORNL/TM-2000/265]. Continuous feedback control is a simple automation since the output value is automatically maintained to a set point without constant human intervention. The greatest body of literature of theory and tools has been developed for control of continuous systems. As an example, maintaining neutron flux to a specific value, hence reactor power, is a continuous control task.

6.2.2 Discrete-Event (logic) Control

Discrete event systems, which are characterized by a finite collection of specific, distinct states and the transitions between them, are controlled by applying logic through combinational and sequential rules. These are not proportional systems. Often, the control of these systems is left to schemes based on heuristic rules inferred from practical plant operation. Industrial plants regularly use programmable logic controllers (PLC) to achieve event control. These commercially

available controllers, which are programmed in ladder logic, offer limited intelligence and do not integrate well with complex continuous control algorithms. Start-up sequencing of simple equipment that invokes discrete (binary) steps is well accomplished by PLCs; however, complex machinery, which may have the possibility of multiple start-up paths depending on internal and external conditions, go beyond the fixed programming of PLCs.

6.2.3 Hybrid control

Continuous control systems experience difficulty achieving their objectives in the face of changing dynamics, such as radical swings in subsystem parameters, component failures, or changes in equipment interconnectivity. In addition, some components are by nature finite state and therefore not controllable by traditional continuous-time methods. Hybrid control is the combination of continuous and discrete control to achieve automatic control over a wide range of system conditions, configurations, and desired outputs. At its simplest level, a hybrid controller can be envisioned as a switching mechanism between a collection of continuous controllers. This issue was discussed in the previous project report and examples were given. In more complex implementations, the hybrid controller would include capability to select modes and states of multiple subsystems to effect a coordinated movement to target goals even with malfunctioning equipment. Diagnostics plays an integral role in the advanced hybrid controller to accomplish that capability.

Combining continuous and discrete control techniques leads to a hierarchical structure with continuous controllers carrying out the tasks of regulation and tracking at lower levels while discrete controllers supervise their operation and make more abstract, strategic decisions. For the most part, the continuous parts and discrete-event parts are designed independently and then combined. Figure 10 below illustrates the hierarchical relationship between continuous and discrete control. Discrete-event activities dominate at the coordination and decision-making levels.

For small-scale systems, the design task of the logic (discrete) component can become complex. The extension to large-scale systems is difficult to scale. The body of literature is growing on design of hybrid control systems, most of which to date is applied to intelligent vehicle systems.

Many subsystems contain multiple control types so that discrete and continuous controls must work together at the equipment level. To control a flow loop for example may require continuous control of pump speed but also requires discrete control over electrical power to motors, oil lift pumps, and valve positioners. Figure 11 illustrates the combination of discrete and continuous control at the component level. (See ORNL/TM-9500 for more in depth descriptions of the automation of discrete and continuous control.).

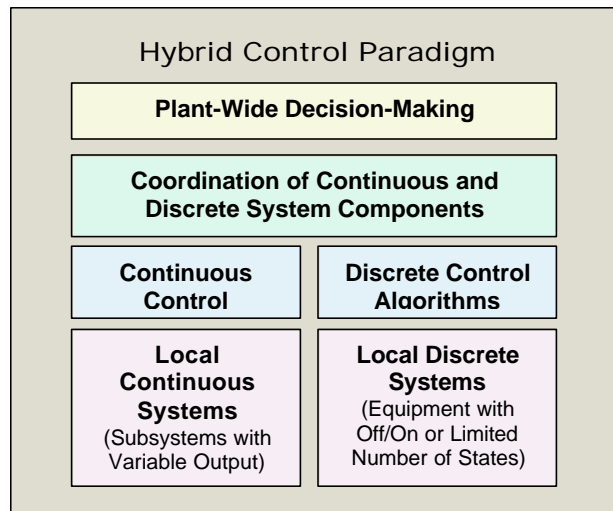


Figure 10. Hybrid control consists of a mixture of continuous and discrete (logic) control functions that coordinates operation of an entire system over a wide range of conditions and states.

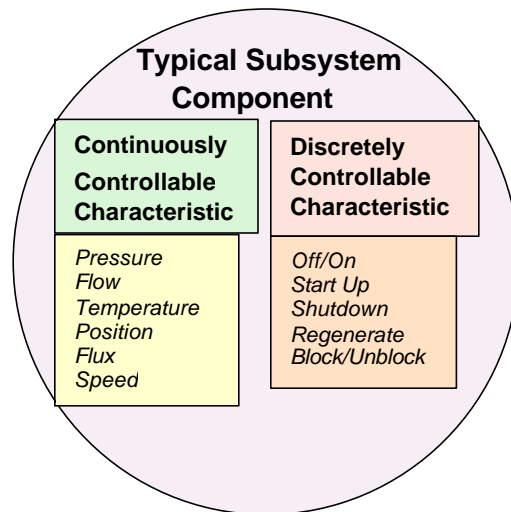


Figure 11. A typical subsystem component may contain both continuously variable and discretely addressable elements. An example might be a feedwater subsystem for which there can be continuous (feedback) control to valve position and pump speed to maintain pressure and flow and binary control over lubricating pumps, cooling, and circuit breaker state

6.2.4 Plant System Classification Method

A nuclear power plant is representative of a large-scale system, in which a number of smaller systems serving specific functions are interconnected and governed by interrelated goals and constraints. These systems are classified according to their functional relationship to the overall plant mission and according to the type of control required to govern them. Prime, support, and utility systems comprise the plant. These systems work together to make a self-sustaining system for electrical power production. A description follows.

Prime plant systems are those that contribute directly to the plant's ultimate output—megawatts. Prime systems, which are cascaded, constitute a flow path for the process. Energy is transferred through various material flows from the reactor core to the electrical energy on the transmission grid. Although the energy flow is unidirectional, material flows are looped (e.g., steam to feedwater). Individual prime systems are influenced by both the upstream and the downstream conditions of their neighbors. As an example, typical nuclear power plant systems are shown in the table below with examples of prime, support, and utility control subsystems. Some of the control is continuous and some discrete. Systems chosen are reactor, primary heat removal, steam generator, main steam transport, feedwater condenser, feedwater heaters and deaerator, turbine, and heat rejection.

Table 1. Major plant system classification by relation to plant output

Major Plant System	Prime Control System	Support Control System	Utility Control System
Reactor	Flux, temperature, rod position	Fuel leak detection, control rod drive motor cooling, cover gas pressure	<i>(Not necessarily associated uniquely with particular plant systems)</i> Plant electrical Fire protection Service water Gas supply Compressed air Cooling water Hydraulics HVAC Radioactive waste Auxiliary steam
Primary heat removal	Flow, pony motor	Motor lubrication, oil lift, purification	
Steam generator and steam transport	Level, blow down, turbine bypass, moisture separators & reheat	Steam isolation	
Feedwater condenser	Pressure, temperature	Demineralizer, pump lubrication, vacuum, gland seal, chemical make-up	
Feedwater heaters and deaerator	Auxiliary feedwater, condensate pump	Motor lubrication, gland seal	
Turbine	Steam flow, main breaker	Turning motor, hydrogen purity, seal, stator cooling, lubrication, block valves	
Heat rejection	Tower flow, fans	Water treatment, Motor-pump lubrication and seals, inventory	

Support systems supply necessary functions and services to the prime plant systems. Support systems control electrical power, cooling, lubrication, and other materials. Such services may be in support of equipment (e.g., motors, pumps, and valves), facilities (e.g., containment, tanks, and piping), and process materials (e.g., water, compressed air, and lubricating oil). Support systems, which are not usually interconnected in cascade form, operate independently with minimal influence on each other.

Utilities, which are a special case of support systems, supply bulk common materials and power to the prime and support systems. Example utilities are plant electrical distribution, fire protection, service water, gas supply, environmental conditioning, hydraulic supply, and auxiliary steam. In many cases, the support systems direct the flow of utility system products in support of the prime systems.

The major interfaces of an automated control system are shown in

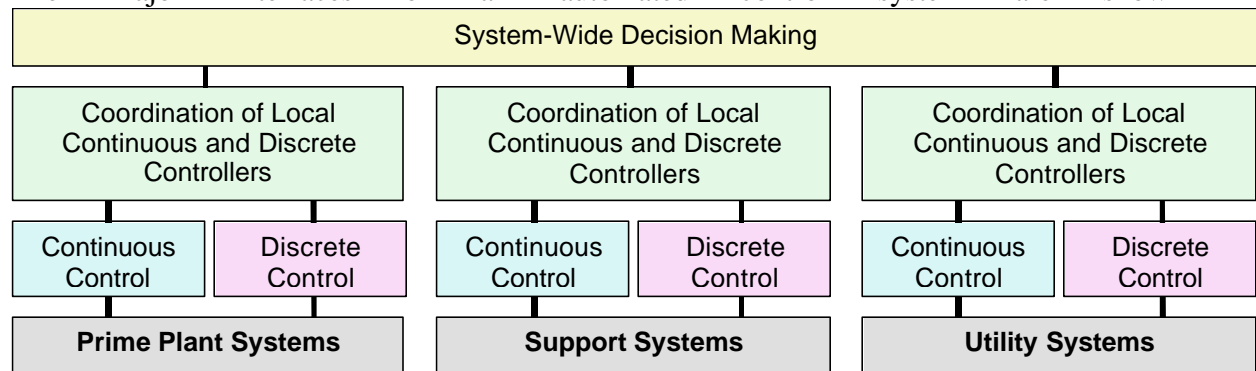


Figure 12. Notice that the control system’s interaction continues to include human operators and maintenance personnel. The figure also indicates the type of information that is passed between the subsystems.

As discussed previously, the automated control system consists of hybrid control with diagnostic capabilities whose functionality is derived directly from the design requirements. The general structure, which consists of local continuous and discrete control and higher-level decision-making and coordination, is shown in Figure 13. The requirements data must be captured and cataloged to permit the design task to carry on module by module.

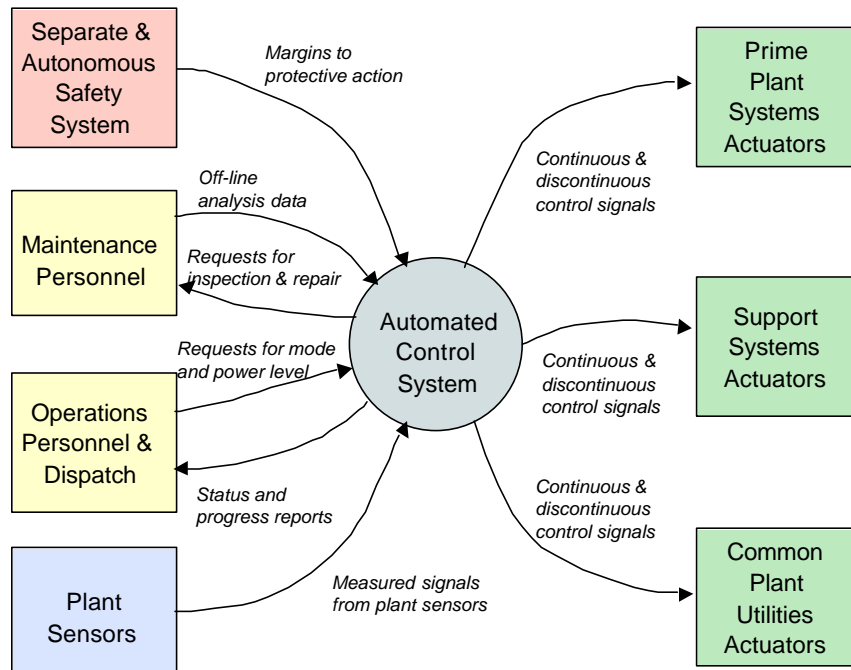


Figure 13. The automated plant control system has boundaries with all major plant subsystems including human operators and maintainers

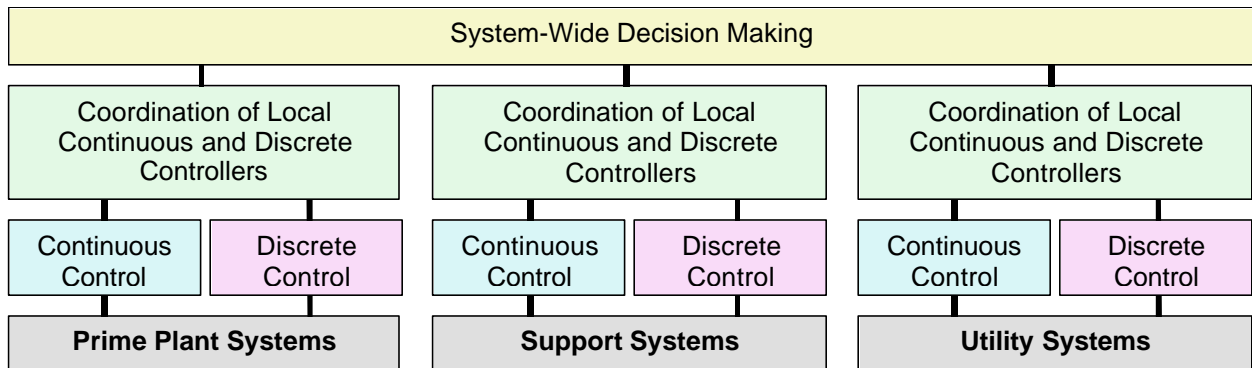


Figure 12. Hierarchical hybrid control applied to nuclear power plant systems

6.2.5 Control Methods that Apply to Automated Hybrid Control

A discussion of several classifications of continuous and discrete control is given in Table 2 below. These are for representative purposes and are not meant to be exhaustive since new methods are regularly reported in the literature. Further, various combinations of algorithmic methods are possible and may result in performance improvement. An automated design process would require an extensive library of algorithmic methods. The library would need to have annotated and cross-indexed information regarding each method's suitability to control specific types of subsystems and equipment. Since new methods and combinations are a source of constant research, the library would need to be easily updated.

Table 2 Listing of representative control methods for both continuous and discrete processes.

Continuous-Time Control Methods	Discrete-Event Control Methods
<p>Proportional-Integral-Derivative (PID) Control — compares the output of a process with a set point and generates an error signal to control the process. May be used in combination with feedback and feed-forward. Part of the error signal may be augmented by derivative or integral action to improve performance.</p> <p>Model-Based Control — contains dynamic models representing the system being controlled. Models are based on differential equations derived either from first principles or by system identification.</p> <p>Adaptive Control — adjusts to changing characteristics of the controlled plant to maintain satisfactory stability and performance.</p> <p>Intelligent Control — based on biological and cognitive models. Examples are expert systems, fuzzy systems, and neural networks.</p> <p>Nonlinear Control— contains nonlinear factors to compensate for plants whose dynamics contain significant nonlinear response, e.g., nonlinear-inverse-dynamic controllers.</p> <p>Optimal Control— minimizes or maximizes objective performance criteria. An example is the linear quadratic Gaussian (LQG) controller. Most applications of optimal control are based on linear state-space equations.</p> <p>Robust Control— provides satisfactory stability and performance for plants whose dynamic characteristics are uncertain or for which significant noise is in the measurement. Examples are LQG with Loop Transfer Recovery and H_{∞}.</p>	<p>If-Then-Else Rules — conditional decision making in which the outcome decision is based on the logical condition of input variables.</p> <p>State-Based Control — a more complex application of <i>if-then-else</i> rules in which the states of the system being controlled are the basis for the structure of the rules. The transition between states is initiated by multiple conditions, which in turn initiate specific actions that drive the system to the desired state. This method may take advantage of graphically oriented display. It is also suited to mathematical formalization.</p> <p>Data-Based Control — emphasizes internal and external activities. The flow of data and communications is well modeled with this method. Timing and sequence is not handled as well with this method.</p> <p>Intelligent Agent-Based Control — is a growing area of research. Several potential advantages are evident including the “mobility” property. Agents can move about in a system to apply their specialty as needed.</p> <p>Formal Methods — applies to a broad range of techniques that employ mathematically precise operators to represent states, modes, and actions.</p> <p>Object-Oriented Control — takes advantage of information hiding and inheritance properties.</p>

6.3 Need for automated requirements capture

For efficient and effective operation, the control system must be an outgrowth of the design requirements to the degree that designers can trace all elements of the final control design back to the initial design requirements. Automating the design-requirement capture process eliminates human errors of omission and commission and frees the designers to consider additional functionality and back-up strategies. The control system should be no less than what is needed to make the requirements operational but no more as well, i.e., extraneous functionality represents extra cost and opportunities for unintended action.

The degree of complexity and cost of failure associated with plant control system justifies the need for automated Requirements capture, which is one of the earliest tasks in control system design process. It is also open-ended in the sense that requirements evolve both during the design phase and throughout the life of the plant. The extent of automation that is intended in the control system determines the level of detail in the requirements captured. The critical steps to creating a workable process are shown in Table 3.

Table 3 The critical steps to creating a requirements-driven control design.

Step	Description
Decompose the plant and its components into individual modules	Define plant as a collection of modules to manage complexity in the subsequent design steps. Models can be developed for each of the modules.
Integrate informal requirements and specifications with formal specification methods	This step produces a logical (mathematical) representation that can be algorithmically manipulated to generate state representations and executable code.
Design control system modules	Once a logical representation has been created, elements of the control system can be designed directly from formal specification.
Integrate discrete and continuous controls with diagnostics to form a complete hybrid control system.	Hybrid control combines logical and continuous control types together to create a system capable of operability over a wide range of conditions that might be imposed from both external and internal influences.
Integrate human operations and maintenance	The goal of automation in the context of nuclear power systems includes effective use of a minimal complement of human operators and maintenance personnel.
Perform exhaustive simulation to validate the control system	Simulation must be exhaustive to cover the variety of transients and component failure scenarios. Models must also be validated.
Implement control system in real-time hardware/software environment	Because of the real-time nature of control, the hardware implementation is tightly connected with the software thus, the total system must be implemented together.

6.3.1 Capture and Design Process

The flow of the entire requirements capture and design process is shown in Figure 14. The first activity shown is the conversion of design requirement and specifications to a logical and mathematical form. Efforts by an international team are discussed below.

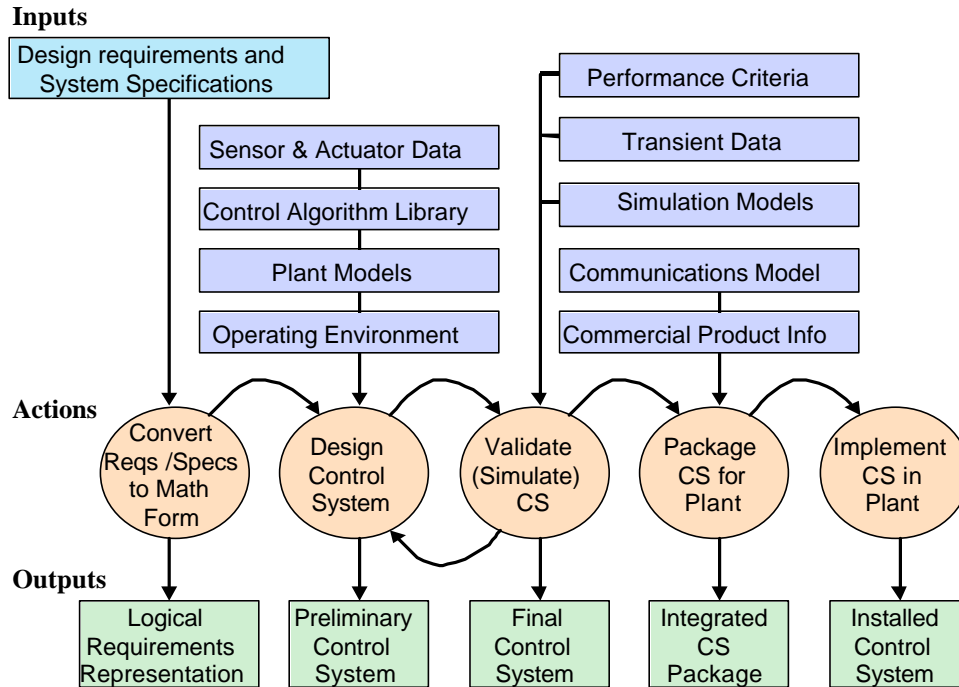


Figure 14. Capturing design requirements and converting them into a final working control system is a part of a larger multi-step process for producing a comprehensive automatic control system.

The outputs of the conversion activity provide both historical information and immediate input to the design activity shown next in the flow. The design activity maps the formal requirements with the control algorithm library, sensor and actuator databases, plant subsystem models, and the characteristic of their operating environment. A preliminary control system is developed for the major prime, support, and utility subsystems. The control incorporates continuous and discrete controls as well as coordination and high-level decision-making. It is at this step that diagnostics and prognostic systems are also developed to be a part of the complete automated control system. It is unclear how much of the design process (as distinct from the capture process) can be automated.

Following the design activity, validation is performed. This activity consists of simulating numerous transient and failed equipment scenarios. Required for this work are performance criteria, transient data, and models for performing simulation runs. This activity iterates with the design activity as simulation uncovers incremental improvements.

The remaining activities shown in Figure 5 place the control system in the context of the plant addressing communication issues, power supply, redundancy, and installation.

6.3.2 Potential mechanisms for capturing requirements

There is a relationship between how you capture and represent the design requirements and the control system architecture and functionality that emerges. It is clear that several academic disciplines must come together to effect an automated design requirements capture effort. Table 4 illustrates six major disciplines that contribute methods and perspective to the capture and design problem. Some of the important topics are discussed below.

Table 4. Several distinct disciplines contribute to automated control requirements capture and subsequent conversion to control system.

Discipline	Typical Areas of Significance
Computer science	Formal logic methods Object oriented programming Intelligent agents
Diagnostics/Prognostics	Statistical analysis Failure modes analysis Predictive modeling Sensor and command signal analysis
System modeling	Plant simulation Continuous systems modeling Discrete systems modeling
Data storage	Data mining Integrated access and access control Data archival and retrieval
Control engineering	Continuous systems control Discrete systems control Hybrid systems control
Human-machine engineering	Interface interaction Cognitive studies

6.3.2.1 Intelligent Agents

The literature indicates that intelligent agents have been growing in their use in computer science over the last decade. Great possibilities exist for using agent capabilities to capture the control system design requirements and create a structure for indicating their relationship. Multiple agents can be assigned to look for and capture according to specific requirement categories. Other agents can scan for inconsistencies and errors as the process evolves. Still other agents can compile resource requirements for the subsequent design steps.

All software agents are programs, but not all programs are agents. The salient differences between traditional computer programs and intelligent agents are compared in the table below. There are activities for which intelligent agents offer potential advantages, e.g., activities that require teaming and may cross several computer platforms. Agents are adaptive and autonomous, which may be advantageous for some applications; however, more traditional (static) programming may be appropriate for some precision calculational tasks.

Table 5. Comparison of traditional computer programming with intelligent agents.

Traditional Computer Program	Intelligent Agent
Static	Dynamic
Direct manipulation: user initiates every action.	Indirect manipulation -- autonomous. Actions may be initiated by either the user or the agent system.
Non-interactive. Dialogs are fully scripted.	Interacts with user and with other agents.
Never changes, unless changed by a human or an error in the program.	Adapts, learns.
Runs one time, and then stops to be run again when called.	Persistent. Continues to run over time.
Predictable--does what you tell it to, even if you didn't mean what you said.	Interprets what you mean, not what you say. In the best of circumstances, actions are based on rules, but they may change over time, or in reaction to different circumstances.
Follows instructions.	May initiate actions as well as respond to instructions.
Stays in one place.	May be mobile, traveling to other servers.

6.3.2.2 Formal specification methods

An international collaborative effort was mounted in 1996 to apply formal specification methods to generating a steam boiler control system from a specification document. The effort by Abrial et al. produced 33 solution methods from numerous students based on numerous formal languages and constructs [Abrial, J.R., Borger, E., and Langmaack, H. (Eds.) "Formal Methods for Industrial Applications: Specifying and Programming the Steam Boiler Control," Springer-Verlag 1996.]. The conclusion is that there are several viable ways to approach the capture and design problem. It is beyond the scope of this report to catalog and critique each of the methods. Also since the publication of the results others have continued the spirit of the development with continued improvement. For example, Petre et al. have examined combining formal and informal design methods to permit better integration with software practice [Petre, L., Qvist, M., and Sere, K., "Distributed Object-Based Control Systems," TUCS Report N. 241, Turku Centre for Computer Science, February 1999].

6.3.3 Categories of information that constitute control system requirements

Specific categories of information are necessary to complete the set of design requirements. The requirements capture process is based on gathering sufficient information for the design process to produce a control system robust under many operating conditions. It is reasonable to expect the capture process to ask for information in specific categories. An example list of categories distilled from the literature is given in Table 6.

It is clear that various checks and verifications are needed for the resulting system designs. Many of these take the form of simulation testing against normal and off normal operating conditions. Simulations should be performed using an exhaustive catalog of simulation scenarios and performance criteria. A discussion of the verification, validation, and simulation testing needs are beyond the scope of this report.

Table 6. Categories of information for the design requirement capture process.

Requirement Category	Description
Plant modes and configuration	Highest level requirements for the operation of the overall plant
Definitions of system functions and roles	Purpose, functional descriptions, behavioral characteristics, and interactions of component subsystems
Operational objectives of subsystems and equipment	Tolerances, performance objectives
Operating limits	Maximum and minimum limits both dynamic and static that lead to damage and abnormal operation. Includes periods that they can be sustained.
Indicators of off-normal conditions	Parameters and combinations of parameters to watch for that indicate abnormal operation.
Minimum essential control parameters	Minimum sensor and actuator sets that allow controlled operation
Description of physical environment surrounding control system components (including stress factors)	Environmental extremes that physical equipment must withstand and for what duration
Consequences of control failure	Analysis of the modes and behaviors that result from damaged and failed control system components
Significant equipment combinations	Description of preferred and detrimental relationships and connections between equipment
Preferred Control algorithms	Description of successful control algorithms and the conditions and parameters associated with good performance
Models of components, sub-systems sensors, and actuators	Models for design of control algorithms and models for use in simulations for control performance validation

6.4 References

6.4.1 General

- Abrial, J.R., Borger, E., and Langmaack, H. (Eds.) "Formal Methods for Industrial Applications: Specifying and Programming the Steam Boiler Control," Springer-Verlag 1996.
- Berkan, R. C., Upadhyayu, B. R., Tsoukalas, L. H., Kisner, R. A., and Bywater, R. L., "Advanced Automation Concepts for Large-Scale Systems," IEEE Control Systems, Vol. 11, No. 6, October 1991, pp. 4-12.
- Bock, C., "Goal-Driven Modeling," Jour. of Object-Oriented Prog., Vol. 13, No. 5, September 2000.
- Feldman, S., "Intelligent Agents: A Primer," Searcher Vol. 7, No. 9, <http://www.infotoday.com/searcher/oct99/feldman+yu.htm>, October 1999.
- Godbole, D., Lygeros J., and Sastry S., "Hierarchical Hybrid Control: A Case Study," Proc. of the 34th IEEE CDC., <http://citeseer.nj.nec.com/godbole94hierarchical.html>, 1994, pp. 166-190.
- Johnson, G. Schrader, D., and Yamamoto, R., "Review Templates for Computer-Based Reactor Protection Systems," NUREG/CR-6680, May 12, 2000.
- Kisner, R. A. and Raju, G. V. S. "Automating Large-Scale Power Plant Systems: A Perspective and Philosophy," ORNL/TM-9500, December 1984.
- Kisner, R. A., "A Framework for Selecting Suitable Control Technologies for Nuclear Power Plant Systems," Control and Instrumentation, Vol. 32, No. 4, October-December 1991, pp. 511-520.
- Na, M. G., "Design of a Receding Horizon Control System for Nuclear Reactor Power Distribution," Nucl. Sci. & Engr., 138 (3): 305-314 July 2001.
- Petre, L., Qvist, M., and Sere, K., "Distributed Object-Based Control Systems," TUCS Report N. 241, Turku Centre for Computer Science, February 1999.
- Thomson, M. and Zayegh, A. "Process Control Concepts in relation to On Line Analyzers," <http://www.measurement.com.au/tp-2.htm>, accessed date September 19, 2001.
- Torrissen, B. C., "Intelligent Agents and Conceptual Modeling," Norwegian Univ. Trondheim, <http://www.pvv.org/~bct/sprithesis/iathesis.html>, May 1996.
- Wooldridge, M and Jennings, N. R., "Intelligent Agents: Theory and Practice," Knowledge Engineering Review, Cambridge Univ. Press, New York, 10:2, June 1995, p 115.

6.4.2 References From ORNL/TM9500

- Atary, J. and Shah, M. 1972. "Modeling and Analytical Control System Design of A Complete Nuclear Reactor," in *Proc. IFAC 5th World Congress*.
- Ball, S. J. et al. June 1982. "Initial Dynamic Simulation of an HTGR Sensible Energy Transport and Storage Plant," ORNL/TM-8226.
- Ball, S. J. January 1964. -Approximate Models for Distributed Parameter Heat Transfer Systems," in *Trans. ISA* 3(1), 38.
- Bell, D. J., Cook, P. A., and Munro, N. 1982. *Design of Modern Control Systems*, Peter Peregrinus Ltd., IEE.
- Birdwell, J. D., Ed. 1984. "Issues in the Design of a Computer-Aided Systems and Control Analysis and Design Environment (CASCADE)," ORNL/TM-9038.
- Bjorlo, T. J., et al. 1970. "Digital Control of Halden Boiling Water Reactor by a Concept Based on Modern Control Theory," *Nucl. Sci. Eng.* 39.
- Blomsnes, B., et al. 1973. "A Computer Control Concept for Load Following Control of Nuclear Plant," HPR-164, Halden Reactor Project.
- Broadwater, R. P. November 1984. Tennessee Technological University, personal communication.
- Broadwater, R. P. October 1984. *Nuclear Power Plant Feedwater Controller Design*, Babcock and Wilcox Patent, U.S. Pat. 4,478,783.
- Broadwater, R. P. February 1983. "A Design Approach for a Power Plant Feedwater Control System," *Control Systems Magazine*, 3 (1).
- Broadwater, R. P. 1977. "Dynamic Modeling and Simulation of a Nuclear Power Plant," PhD Dissertation, VPI&SU, Blacksburg, Virginia.
- Chen, A. T. 1976. "A Digital Simulation for Nuclear Once Through Steam Generators," PhD Dissertation, The University of Tennessee, Knoxville.
- Chu, Y. Y. and Rouse, W. B. December 1979. "Adaptive Allocation of Decision-Making Responsibility Between Human and Computer in Multi-Task Situations," *IEEE Trans. Systems, Man, and Cybernetics* SMC-9(12), 769-78.
- Colley, R. W. August 1983. "Alternative Strategies for the Procedure Prompting System" Hanford Engineering Development Laboratory, HEDL-PC-2398.
- Colley, R. W., et al. August 1982. "Procedure Prompting System," HEDL-TC-2242.
- Colley, R. W. April 1982. "Sequencing Established States: A Methodology for Automatic Procedure Generation and Rule-Based System Control," HEDL-TC-2210.

- Cummins, J. D., et al. 1973. "Application of Modern Control Theory in Nuclear Power," *Proc. 1973 IAEA Conf. Nuclear Power Plant Control and Instrumentation*, Prague, Yugoslavia.
- Daniel, W. R. July 1984. "Large-Scale Prototype Breeder, Plant Control and Protection System Design Review," Sunnyvale, California.
- Davison, E. J. 1966. "A Method of Simplifying Linear Dynamic Systems," *IEEE Trans. Autom. Control* AC-11(1).
- DeMarco, T. 1979. *Structured Analysis and System Specification*, Yourdon Press.
- Demore, L. A. and Matta, J. A. January 1975. "Initial Reactor Control System Design for CRBRP," Westinghouse.
- Eykhoff, P. 1974. *System Identification, Parameter, and State Estimation*, J. Wiley & Sons, New York.
- Feigenbaum, M. J. 1979. "The Universal Metric Properties of Nonlinear Transformations," *J. Statistical Physics* 21(6).
- Findeisen, W., et al. 1980. *Control and Coordination in Hierarchical Systems*, Wiley.
- Frogner, B. and Grossman, L. M. 1975. "Estimation and Optimal Feedback Control Theory Applied to Boiling Water Reactor," *Nucl. Sci. & Eng* 58.
- Gane, C. and Sarson, T. 1979. *Structured Systems Analysis: Tools and Techniques*, Prentice-Hall.
- Huynh, Q. A. June 1978. "Cost Benefit Analysis for Dynamic Testing in FFTF," M.S. Thesis, The University of Tennessee, Knoxville.
- Jamshidi, M. 1983. *Large-Scale Systems*, North Holland.
- Johnson, C. D., 1976. "Theory of Disturbance - Accommodating Controllers," in *Control and Dynamic Systems: Advances in Theory and Applications*, 12, Academic Press, pp. 387-489.
- Joyner, Luther, 1984. Joyner Engineers, Inc., Forest, Virginia, personal communication.
- Kerlin, T. W., et al. November 1977. "Identification of Nuclear Systems," *Nucl. Technol.* 36.
- Kerlin, T. W. and Katz, E. M. May 1983. "Pressurized Water Reactor Modeling for Long-Term Power System Dynamics Simulations," EPRI, EL-3087.
- Kisner, R. A. and Frey, P. R. 1982. "Functions and Operations of Nuclear Power Plant Crews," NUREG/CR-2587, ORNL/TM-8237.
- Lakely, N. R. April 1982. "Why Structured Techniques in Software Development," Power/Industrial Simulator Users Conference, Calverton, Maryland.

- Lipinski, W. C. and Vacroux, A. G. 1970. "Optimal Digital Computer Control of Nuclear Reactors," *IEEE Trans. Nucl. ScL* **NS-17**.
- Mesarovic, M. D., et al. 1970. *Theory of Hierarchical Multilevel Systems*, Academic Press.
- Morrow, I. and Robinson, W. B. October 1983. "Software Engineering for Real-Time Systems," *Advances in Instrumentation* 38, Part 1, Proc. ISA International Conference and Exhibit, Houston, Tex.
- Oguri, K. and Ebizuka, Y. July 1975. "Synthesis of Digital Control Systems for Nuclear Systems," *J. Nucl. ScL Technol.*
- Page-Jones, M. 1980. *The Practical Guide to Structured Systems Design*, Yourdon Press.
- Pulliam, R., et al. 1983. "A Methodology for Allocating Nuclear Power Plant Control Functions to Human or Automatic Control," NUREG/CR-3331, ORNL/TM-8781.
- Rouse, W. B., et al. 1984. "A Method for Analytical Evaluation of Computer-Based Decision Aids," NUREG/CR-3655, ORNL/TM-9068.
- Schultz, M. A. 1961. *Control of Nuclear Reactors and Power Plants*, McGraw Hill.
- Smith, D. E. May 1984. "The Application of Automated Reasoning Software to the Operation of LMFBR's," HEDL-TC-2568.
- Ward, P. and Campbell, D. October 1983. "Requirements Definition for Process Control Systems," *Advances in Instrumentation* 38, Part 1, Proc. ISA International Conference and Exhibit, Houston, Tex.
- Weaver, L. E. 1967. *Reactor Dynamics and Control*, Elsevier.
- Weaver, T. L. September 1983. "The Design Method for a Control Process for Multiple Source Election Cyclotron Resonance Heating for the ELMO Bumpy Torus-Proof of Principle Device," *Nuclear Technology/Fusion*, Special Issue.
- Yourdon, E. and Constantine, L. L. 1979. *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design*, Prentice-Hall.
- Yourdon Course, January 1984. "Structured Analysis and Design for Real-Time Systems," Yourdon, Inc., 1133 Avenue of the Americas, New York, N.Y.
- Zaborszky, J. "Digital Control of the Large Electric Power System in Normal and Emergency State Operation by Decision and Control," Partial Collection of Publications from Research Conducted at Department of Systems Science and Mathematics, Washington University, St. Louis, Mo. (no date available).
- Zektomaki, N. A., Sandell, N. R., and Athans, M. February 1981. "Robustness Results in Linear-Quadratic Gaussian Based Multivariable Control Design," in *IEEE Transactions on Automatic Control*, AC-26, (1) pp. 75-93.

Zhiwei, L. and Kerlin, T. W. September 1983. "Modeling and Low Cost Simulation of Whole PWR Power Plant," Dept. of Nuclear Eng., The University of Tennessee, Knoxville.

6.4.3 General Reference for Hybrid Control and Intelligent Agents

"What's new: hybrid control system targets batch processes," *Chemical Engineering Progress* 93, (July 1997).

Altman, E. and Gaitsgory, V., "Asymptotic optimization of a nonlinear hybrid system governed by a Markov decision process," *SIAM Journal on Control and Optimization* 35 (6), 2070-2085 November 1997.

Altman, E. and Gaitsgory, V., "Control of a hybrid stochastic systems," *Systems & Control Letters* 20(4), April 1993.

Anon (Ed.), *Proceeding of the 1997 IEEE International Symposium on Intelligent Control*, p. 436, Istanbul, Turkey, July 16-18, 1997.

Au, B. K., "Interface considerations for a hybrid control system using programmable controllers and distributed control systems," 1991 TAPP/ISA PUPID Process Control Conference, Lake Buena Vista, FL, March 3-7, 1991.

Aubin, J. P. and Haddad, G., "Cadenced runs of impulse and hybrid control systems," *International Journal of Robust and Nonlinear Control* 11(5), April 20, 2001.

Bencze, W. J. and Franklin, G. E., "Separation principle for hybrid control system design," *IEEE Control Systems Magazine* 15(2), 1995.

Bieleck, T. R., "Linear-quadratic control problem revisited," *SIAM Journal on Control and Optimization* 33(5), September 1995.

Branicky, M. S., Borkar, V. S., and Mitter, S. K., "Unified framework for hybrid control: model and optimal control theory," pp. 31-45 in *IEEE Transactions on Automatic Control*, Piscataway, NJ, January 1998.

Chen, J. and Howell, J., "A self-validating control system based approach to plant fault detection and diagnosis," *Computers & Chemical Engineering* 25(2-3), March 15, 2001.

Choi, Y. K., Lee, M. J., Kim, S., and Kay, Y. C., "Design and implementation of an adaptive neural-network compensator for control systems," *Engineering, Computing & Technology*.

Ferreira, J.C.E. and Sysk, R. A., "An investigation of the influence of alternative process plans on equipment control," *Journal of Manufacturing Systems* 19(6), 2001.

Fierro, R., Lewis, F. L., and Liu, K., "Hybrid control system design using a fuzzy logic interface," *Circuits, Systems, and Signal Processing* 17(3), 1998.

Gao, J. and Chen, Z., "Formal verification of hybrid system," *Journal of Beijing University of Aeronautics and Astronautics*, 25(2), 1999.

- Getmanov, V. G., Zhirov, M. V., and Shakhovskoi, A. V., "An identification algorithm for a linear discrete dynamic control system," *Automation and Remote Control* 62(4), April 2001.
- Grasselli, O. M., Menini, L., and Valigi, P., "Continuous-time asymptotic tracking and regulation for parameter-dependent multi-rate sampled-data systems," pp. 3626-3630 in *Proceeding of the 1997 American Control Conference*, Albuquerque, NM, June 4-6, 1997.
- Hajji, M. S., Bass, J. M., Browne, A. R., Schroder, P., Cross, P. R., and Fleming, P. J., "Development framework: work in progress towards a real-time control system design environment." Pp. 4/1-4/3 in *Proceedings of the 1996 IEE Colloquium on Advances in Computer-Aided Control System Design, London, UK, March 14, 1996*.
- Holmes, M. and Ray, A., "Fuzzy damage-mitigating control of a fossil power plant," pp. 140-147 in *IEEE Transactions on Control Systems Technology*, University Park, PA, January 2001.
- Inoue, T., Taniguchi, H., and Ikeguchi, Y., "A model of fossil fueled plant with once-through boiler for power system frequency simulation studies," pp. 1322-1328 in *IEEE Transactions on Power Systems*, Tokyo, Japan, November 2000.
- Isaka, S., "Fuzzy temperature controller and its applications," pp. 59-65 in *Proceedings of SPIE-The International Society for Optical Engineering*, Boston, MA, September 8-10, 1993.
- Kamwa, I., Grondin, R., and Hebert, Y., "Wide-area measurement based stabilizing control of large power systems-A decentralized/hierarchical approach," pp. 136-153 in *IEEE Transactions of Power Systems*, Quebec, Canada, February 2001.
- Kanoun, K. and Ortalo-Borrel, M., "Fault-tolerant system dependability-Explicit modeling of hardware and software component-interactions," pp. 363-376 in *IEEE Transactions on Reliability*, Toulouse, France, December 2000.
- Kim, J. T., Kwn, K. C., Hwang, I. K., Lee, D. Y., Park, W. M, Kim, J. S., and Lee, S. J., "Development of advanced I&C in nuclear power plants: ADIOS and ASICS," *Nuclear Engineering Design* 207(1), Jaejon, South Korea, July 2001.
- Ko, B. S. and Edgar, T. F., "Performance assessment of multivariable feedback control systems," *Automatica* 37(6), June 2001.
- Larsen, M. H., Langer, G., and Kirkby, P., "A modeling approach for improved implementation of information technology in manufacturing systems," *Intelligent Automation and Soft Computing* 7(1).
- Lemmon, M. and Bett, C., "Hybrid control system design using robust linear control agents," pp. 2688-2693 in *Proceeding of the 1995 34th IEEE Conference on Decision and Control*, New Orleans, LA, December 13-15, 1995.
- Keller, J. P. and Anderson, B.D.O., "H-infinity optimal controller discretization," pp. 1781-1785 in *Proceeding of the 29th IEEE Conference on Decision and Control*, Honolulu, HI, December 5-7, 1990.

- Lemmon, M. and Antsaklis, P., "Computationally efficient framework for hybrid controller design," pp. 333-338 in *Proceeding of the IEEE/IFAC Joint Symposium on Computer-Aided Control System Design, Tucson, AZ, March 7-9, 1994*.
- Lu, T. P. and Yih, Y. W., "An agent-based production control framework for multiple-line collaborative manufacturing," *International Journal of Production Research* 39(10), July 2001.
- Mackriell, L. E., Kwok, K.C.S., and Samali, B., *Engineering Structures*, 19(10), Elsevier Science Ltd, Oxford England, October 1997.
- Malmborg, J. and Eker, J., "Hybrid control of a double tank system," pp. 133-138 in *Proceeding of the 1997 IEEE International Conference on Control Applications*, Hartford, CT, October 5-7, 1997.
- Molchanov, A. P and Bauer, P. H., "Robust stability of digital feedback control systems with floating point arithmetic," pp. 4251-4258 in *Proceedings of the 1995 34th IEE Conference on Decision and Control*, New Orleans, LA, December 13-15, 1995.
- Moughamir, Szaytoon, J., and Afilal, L., "Modelling and analysis of an industrial hybrid control system," pp. 851-856 in *Proceeding of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, San Diego, CA, October 11-14, 1998.
- Na, M. G., "Design of a receding horizon control system for nuclear reactor power distribution," *Nuclear Science and Engineering* 138(3), Kwangju, South Korea, July 2001.
- Namba, R., Yamamoto, T., and Kaneda, M., "Design and experimental evaluation of a self-tuning controller supplementing a robust PID controller," *JSME International Journal Series C-Mechanical Systems Machine Elements and Manufacturing* 44(1), Hiroshima, Japan, March 2001.
- Orski, D., "Bubnicki method for decision making in a system with hybrid knowledge representation," pp. 230-237 in *Proceedings of the 1998 13th International Conference on Systems Science*, Wroclaw, Poland, September 15-18, 1998.
- Parisini, T. and Sacone, S., "Stable hybrid control based on discrete-event automata and receding-horizon neural regulators," *Automatica* 37(8), August 2001.
- Piccoi, B., "Necessary conditions for hybrid optimization," pp. 410-415 in *38th IEEE Conference on Decision and Control*, Phoenix, AZ, December 7-10, 1999.
- Punal, A., Melloni, P., Roca, E., Rozzi, A., and Lema, J. M., "Automatic start-up of UASB reactors," *Journal of Environmental Engineering-ASCE* 127(5), Santiago, Spain, May 2001.
- Raisch, J. and O'Young, S. D., "Time-driven supervisory control of hybrid dynamical systems," pp. 716-721 in *Proceedings of the 1996 UKACC International Conference on Control*, Exeter, UK, September 2-5, 1996.

- Robinson, D., Chen, R., McAvoy, T., and Schnelle, P.D., "An optimal control based approach to designing plantwide control system architectures," *Journal of Process Control* 11(2), College Park, MD, April 2001.
- Skiadas, P. and Koussoulas, N. T., "Semi-quantitative models in hybrid control systems," pp. 590-592 in *Proceedings of the 1996 3rd IEEE International Conference on Electronics, Circuits, and Systems*, Rodos, Greece, October 13-16, 1996.
- Smirnov, N. V. and Smirnova, T. Y., "The synthesis of multi-programme controls in bilinear systems," *PMM Journal of Applied Mathematics and Mechanics* 64(6), 2000.
- Smith, D. J., "New control system improves Cogen flexibility," *Power Engineering* 105(3), March 2001.
- Stiver, J. A., Koutsoukos, X. D., and Antsaklis, P. J., "An invariant-based approach to the design of hybrid control systems," *International Journal of Robust and Nonlinear Control*, 11(5), April 2001.
- Stiver, J. A., Antsaklis, P. J., and Lemmon, M. D., "Hybrid control system design based on natural invariants," pp. 1455-1460 in *Proceedings of the 1995 34th IEE Conference on Decision and Control*, New Orleans, LA, December 13-15, 1995.
- Stiver, J. A. and Antsaklis, P. J., "Extracting discrete event system models from hybrid control systems," pp. 298-301 in *Proceeding of the 1993 IEEE International Symposium on Intelligent Control*, Chicago, IL, August 25-27, 1993.
- Stiver, J. A., and Antsaklis, P. J., "State space partitioning for hybrid control systems," pp. 2303-2303 in *Proceedings of the 1993 American Control Conference*, San Francisco, CA, June 2-4, 1993.
- Sussmann, H. J., "Maximum principle for hybrid optimal control problems," pp. 425-430 in *38th IEEE Conference on Decision and Control*, Phoenix, AZ, December 7-10, 1999.
- Tani, T., Sasada, T., Utashiro, M., Umamo, M. and Tanaka, K., "Neuro-fuzzy hybrid control system of nonlinear process in petroleum plant," pp. 2501-2506 in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Perth, Austria, November-December 1, 1995.
- Tani, T., Utashiro, M., Umamo, M., and Tanaka, K., "Application of practical fuzzy-PID hybrid control system to petrochemical plant," pp. 1211-1216 in *Proceedings of the 3rd IEEE Conference on Fuzzy Systems*, Orlando, FL, June 26-29, 1994.
- Tani, T., Murakoshi, S., Sato, T., Umamo, M., and Tanaka, K., "Application of neuro-fuzzy hybrid control system to tank level control," pp. 618-623, *Second IEEE International Conference on Fuzzy Systems*, San Francisco, CA, March 28-April 1, 1993.
- Tronci, E., "Optimal finite state supervisory control," pp. 2237-2242 in *Proceedings of the 1996 35th IEEE Conference on Decision and Controls*, Kobe, Japan, December 11-13, 1996.

- Turner, P., Boudewijnse, G. J., and Donderi, D., "Process control problems in the pulp and paper industry-The human factor ought to be considered," *Pulp & Paper-Canada* 102(6), Montreal, Canada, June 2001.
- Ul'yanov, V. S. and Yazenin, A.V., "A soft computing-based mathematical model of an intelligent control system for a complex, globally unstable object," *Journal of Computer and Systems Sciences International* 40(3), Tver, Russia, May-June 2001.
- Vadigepalli, R., Gatzke, E. P., and Doyle, F. J., "Robust control of a multivariable experimental four-tank system," *Industrial & Engineering Chemistry Research* 40(8), Newark, DE, April 18, 2001.
- Van Bremen, A.J.N and de Vries, T.J.A., "Design and implementation of a room thermostat using an agent-based approach," *Control Engineering Practice* 9(3), March 2001.
- Van Schuppen, J. H., "Coordination control of parallel operations in a hybrid control system," pp. 2095-2100 in *Proceedings of the 1998 37th IEEE Conference on Decision and Control*, Piscataway, NJ, 1998.
- Wright, D. T. and Williams, D. J., "Object-oriented software design techniques for process control," *Transactions of the Institute of Measurement Control*, 16(1), London, England, 1994.
- Yang, H. and Choi, J., "Design of a hybrid control system using neural networks," pp. 1255-1260 in *Proceedings of the 1995 IEEE International Conference on Systems, Man and Cybernetics*, Vancouver, BC, October 2-25, 1994.
- Young, K. W., Piggin, R., and Rachitrangsan, P., "An object-approach to an agile manufacturing control systems design," *International Journal of Advanced Manufacturing Technology* 17(1), 2001.