

# Neuro-Predictive Process Control Using On-Line Controller Adaptation

Alexander G. Parlos, *Senior Member, IEEE*, Sanjay Parthasarathy, and Amir F. Atiya, *Senior Member, IEEE*

**Abstract**—A novel architecture for integrating neural networks with industrial controllers is proposed, for use in predictive control of complex process systems. In the proposed method, a conventional controller, e.g., a proportional-integral (PI) controller, is used to control the process. In addition, a recurrent neural network is used in the form of a multistep-ahead predictor (MSP) to model the process dynamics. The parameters of the PI controller are tuned by a backpropagation-through-time (BTT)-like approach using “parallel learning” to achieve acceptable regulation and stabilization of the controlled process. The advantage of such a formulation is the effective on-line adaptation of the controller parameters while the process is in operation, and the tracking of the different process operating regimes and variations. The proposed method is used in the stabilization and transient control of u-tube steam generator (UTSG) water level. Currently, available constant-gain PI controllers are unable to stabilize the UTSG at low operating powers, necessitating manual operator control. The proposed predictive controller stabilizes the process and improves its transient performance over the entire operating range. The adaptive PI controller can handle severe operational transients in the presence of significant actuator noise and process parameter drifts that could result from aging and other wear-and-tear effects.

**Index Terms**—Predictive process control, recurrent neural networks, self-tuning proportional-integral (PI) control.

## I. INTRODUCTION

THE majority of industrial controllers employ fixed parameter, linear controller structures for controlling complex nonlinear systems [15]. More specifically, an overwhelmingly large fraction of these industrial controllers utilize standard classical structures, such as proportional-integral (PI) control. However, the set of control gains required for desired system performance at different operating conditions might be different. The use of constant-gain controllers is not always desirable because many process parameters drift throughout the system life-time as a result of wear-and-tear. Such behavior suggests the need for on-line adaptation, or self-tuning, of controller parameters to achieve desired process response. Over the past a couple of

decades, many methods have been developed for self-tuning of controller parameters, and currently many control vendors offer such solutions [3].

Model predictive control (MPC) refers to the direct use of an explicit and separately identifiable model for controlling a process [5], [14]. The core of all MPC algorithms is the moving horizon approach, also known as the open-loop optimal feedback approach, proposed by Propoi in 1963. The MPC designs yield control systems capable of operating without expert intervention for extended periods of time. An identified process model predicts the future response and then, the control action is determined so as to obtain the desired performance over a finite time horizon. The control problem that must be solved is an on-line optimization of the manipulated variables to satisfy multiple, changing performance criteria in the face of changing process characteristics, including constraints [7]. The MPC technique is a dynamic optimization approach to control problems. The flexible constraint-handling capabilities of MPC make it most suitable for process control problems.

Neural networks can be used to determine controller parameters, because of their well-known ability to solve complex problems by learning relationships directly from data. In this decade, certain neural networks have generated a lot of interest for use in nonlinear system identification and control [16], [17], [33]. They have been found to approximate arbitrary nonlinear mappings and they have been effectively used for the control of complex dynamic systems. Neural networks provide a framework for deriving analytic expressions for the modeling error gradients with respect to modeling parameters. The majority of neural network control applications have centered around the use of a neural network as a controller, which is often combined with a neural network identifier. In many industrial control applications, however, the controllers in place continue to be conventional linear controllers, such as PI or proportional-integral-derivative (PID) controllers. Large investments would be required to install new controller structures, such as neural-network controllers, in place of the simpler, conventional controllers. A less costly alternative is to develop approaches that would adapt conventional controller gains using neural-network algorithms, so as to successfully track the different process operating regions, and hence enhance control performance.

For a systematic classification of neural networks-based control systems, the reader is referred to a recent survey by Agarwal [1]. Neurocontrol has been proposed for various areas such as process control, control of robots, vehicles and structures, manufacturing applications and teleoperations [11], [12], [26], [27]. Bersini and Gorrini have recently developed an optimal neurocontrol algorithm based on a simplification of the

Manuscript received July 6, 2000. Manuscript received in final form November 7, 2000. Recommended by Associate Editor S. Nair. This work was supported by the U.S Department of Energy under grant *DE-FG07-89ER12893* to Texas A&M University. The work of A. F. Atiya was supported by the NSF Engineering Research Center at CalTech.

A. G. Parlos is with the Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: a-parlos@tamu.edu).

S. Parthasarathy is with the Honeywell Technology Center, Minneapolis, MN 55418 USA (e-mail: sanjay@htc.honeywell.com).

A. F. Atiya is with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: amir@work.caltech.edu).

Publisher Item Identifier S 1063-6536(01)03359-0.

backpropagation-through-time method [4]. Neurocontrol has tremendous potential in the process industries. Process plants are usually custom-built, and first-principle models developed for one process plant cannot be necessarily applied to other processes. The lack of portability of such models results in more time being spent on modeling and identification, and thus increases the operational costs of process plants. Neural networks offer a generic framework for modeling complex systems, and are cost-effective and easy to use.

Recently, parameterized neurocontrollers for process systems have been investigated by Samad and Su with encouraging results [29]. Wang *et al.* have investigated the use of neural networks for the adaptive generalized predictive control of nonlinear systems [31]. Omatu and Yoshioka applied genetic algorithms and self-tuning for neuro-PID control [18]. Prasad *et al.* applied a neural-network model for the long-range control of a thermal power plant [24]. More specific to the process studied in this paper, Kothare *et al.* proposed a level controller for the steam generator of nuclear power plants using extensions of linear MPC methods [10].

The main contribution of this paper is the development of a control method using the multistep-ahead predictive (MSP) capabilities of neural networks and the minimization of the prediction horizon error, to tune the available industrial controller parameters. The focus of this paper is on the development and demonstration of a control architecture for real-world process systems using minimal underlying assumptions. As such, theoretical stability and algorithmic convergence issues, typically addressed in linear control studies, are difficult to ascertain and are not dealt with here. The proposed control method uses a recurrent neural network to model the open-loop process dynamics. In this approach the average prediction error calculated throughout the prediction horizon is minimized. Using the neural-network model, analytical expressions can be derived for the error gradients of the controller parameters, and hence an adaptation rule for these gains can be derived. Toward this end we use MPC concepts that, in general, have proven successful when used in combination with a neural-network component [13], [25], [28], [34].

The proposed method is demonstrated by stabilizing and controlling the transient response of a U-tube steam generator (UTSG), a critical component in many nuclear power plants. The UTSG is an open-loop unstable process, and conventional PID controllers with constant parameters are unable to stabilize it at low operating powers. Furthermore, at low operating power regimes the process exhibits nonminimum phase and highly nonlinear behavior. The obtained results demonstrate the stabilization of the process throughout its operating envelope and its superior transient performance when compared with the currently utilized constant-gain PID controllers. The benefits of our proposed approach are demonstrated through extensive simulations on a plant-validated UTSG simulator.

The remainder of this paper is organized as follows: in Section II, the controller architecture proposed in this study is described in some detail. Section III presents the UTSG water level control problem and the controller development. In Section IV, the developed water level controller performance evaluation is summarized by presenting a number of simulation experiments.

The paper concludes in Section V with a discussion and the conclusions drawn from this research.

## II. PROPOSED CONTROLLER CONFIGURATION

In this section the proposed controller and its components are described, without the specifics of the water level control problem analyzed later in the paper.

### A. Overall Description

The proposed control method integrates concepts from MPC and from developments in neural networks for adaptive, on-line tuning of linear controller parameters. The proposed control system configuration is shown in Fig. 1. The figure depicts an existing controller with tunable gains in a closed-loop configuration and the proposed controller, shown in Fig. 1 as “proposed augmentation.” An integral component of the proposed controller is a recurrent neural-network operating in parallel with the process to be controlled. The neural network and the associated weight adaptation law comprise the predictor module. The gain adaptation module comprises of a simulated version of the existing controller, and the gain adaptation law. In the proposed system configuration, there are two feedback loops, the controller feedback loop, process and controller, and the prediction loop, the predictor and the gain adaptation law with the simulated controller. The two components of the prediction loop are briefly described below.

1) *Predictor Module:* A neural-network predictor is used to simulate the MSP response of the process. The predictor can be trained off-line to accurately estimate the process response to manipulated variables (or the control inputs) and future assumed values of the disturbances. With the recent development of faster and more accurate recurrent network learning algorithms some parts of this task can also be performed on-line [2], [21]. A recurrent network architecture has been found to “learn” the process dynamics to considerable accuracy and it is chosen as the empirical model structure for the identification [19]. The case-study presented in this research uses a dynamic learning algorithm for off-line and on-line supervised training of the network [19]. The persistent excitation necessary for the closed-loop identification is provided via the disturbances to the process. Off-line training using the closed-loop process response is performed until the network learns the process dynamics to sufficient accuracy. The trained network is then incorporated in the prediction loop. Further on-line tuning of the neural network is performed as necessary, to account for the drifts in process parameters and thereby enhance the identification accuracy.

2) *Gain Adaptation Module:* The gain adaptation module comprises of the simulated controller, and the associated gain adaptation law. In this study, we use a traditional PI controller for illustration purposes. The proposed method allows use of any controller structure instead. Without gain adaptation, at every time instant the simulated controller has the same gains as the actual controller in the feedback loop. When gain adaptation is implemented, the two controllers are tuned in tandem. The gain adaptation law uses the predicted tracking error over the

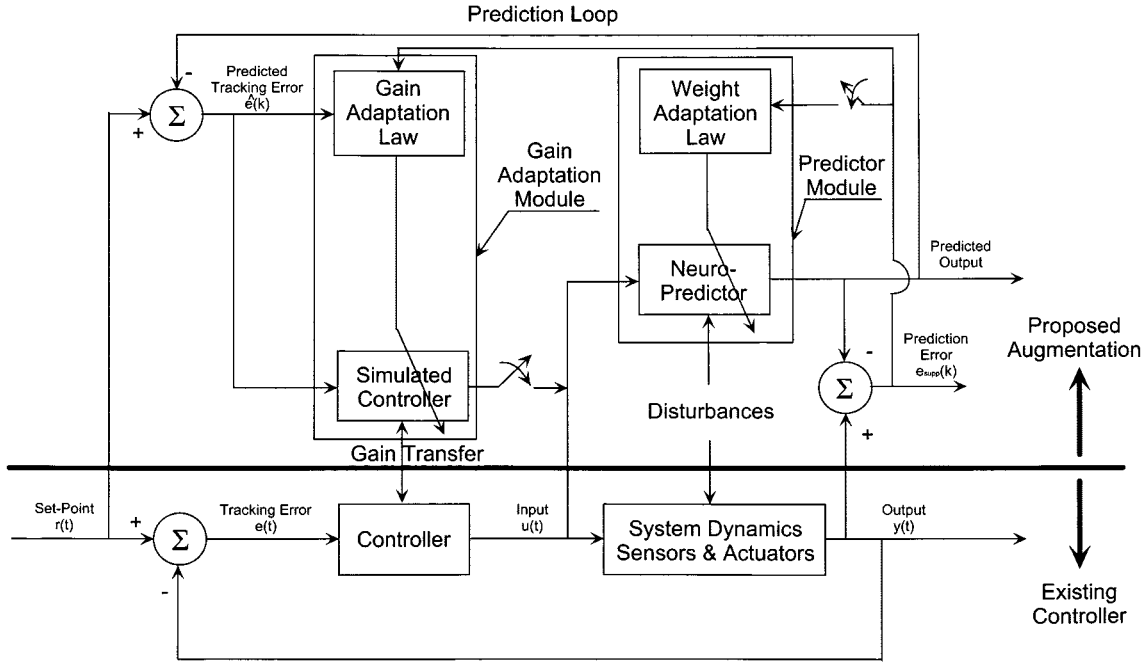


Fig. 1. The proposed control system configuration.

prediction horizon to compute changes in the controller gains. The simulated controller gains are then updated accordingly.

As seen from Fig. 1, there are three relevant error signals that must be defined. The tracking error,  $e(t)$ , is defined as the difference between the controller set-point and the measured process output that must be controlled.<sup>1</sup> This error signal is used in the actual feedback loop of the existing controller. The predicted tracking error,  $\hat{e}(k)$ , is defined as the difference between the same controller set-point, as before, and the process output predicted by the neuro-predictor. This is used in forming the error signal throughout the prediction horizon. Finally, the prediction error,  $e_{\text{supp}}(k)$ , is defined as the difference between the measured and predicted process outputs. The prediction error is used for controller gain adaptation and for improving the neuro-predictor via weight adaptation.

### B. Predictor Module

Feedforward multilayer perceptron neural networks have been the prevalent architecture used in the majority of neural-network applications [9]. These networks are good at approximating mainly static nonlinearities. In order to approximate dynamic systems with significant temporal variations, recurrent networks are preferred. Recurrent neural networks have links which feed the output of each node back to itself, either directly, or through other nodes (*crossstalk*). In this study the recurrent neural network described by Parlos *et al.* [19] is used to predict the MSP process dynamics in the prediction loop of the controller. The predictor has been trained with a dynamic learning algorithm, which has been found very effective in identifying nonlinear dynamical systems. More details regarding this learning algorithm can be found in [19].

<sup>1</sup>In this paper,  $t$  and  $k$  are used to denote continuous and discrete time, respectively.

The equations describing the  $i$ th node in the  $l$ th layer of the recurrent network used in this study are given as follows:

$$\begin{cases} z_{[l,i]}(k+1) = \sum_{j=1}^{N(l)} w_{[l,j][l,i]} x_{[l,j]}(k) \\ \quad + \sum_{j=1}^{N(l-1)} w_{[l-1,j][l,i]} x_{[l-1,j]}(k+1) + b_{[l,i]} \\ x_{[l,i]}(k) = F_{[l,i]}(z_{[l,i]}(k)) \end{cases} \quad (1)$$

where

$z_{[l,i]}(k)$  and  $x_{[l,i]}(k)$  state and output variables of the  $i$ th node in the  $l$ th layer, respectively;  
 $b_{[l,i]}$  bias to the node;  
 $w_{[l,j][l',i]}$  weight associated with the link between the  $j$ th node of the  $l$ th layer to the  $i$ th node of the  $l'$ th layer, with  $j = 1, \dots, N(l)$  (the number of nodes per layer) and  $l = 1, \dots, \mathcal{L}$  (number of layers).

The nonlinear discriminatory function,  $F_{[l,i]}(\cdot)$ , of the  $i$ th node in the  $l$ th layer is a squashing function. For the input and output layers,  $l = 1$  and  $l = \mathcal{L}$ , the discriminatory function is assumed linear.

### C. Gain Adaptation Module

Now the controller gain adaptation law is briefly described. Let the objective function to be minimized be given by

$$V(k) = \frac{1}{2} \sum_{i=1}^H \hat{e}^2 \left( k + \frac{i}{k} \right) \quad (2)$$

where  $\hat{e}(k + i/k)$  is the predicted tracking error at time step  $k+i$ , using input and output observations at time step  $k$ , where  $k$  represents the trailing edge of the prediction horizon. This is the error between the predictor output and the controller set-point.

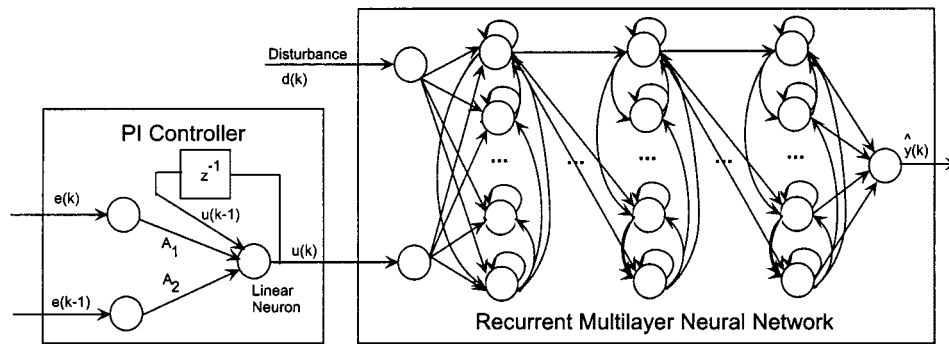


Fig. 2. The PI controller and neuro-predictor in series.

The predicted tracking error minimization is performed over a finite horizon  $H$ .

The predicted tracking error can be parameterized in terms of the controller gains and the neural-network weights and biases. The effect of controller gain tuning on the tracking error can be quantified by the gradient of the predicted tracking error with respect to the controller gains. An analytical expression for this gradient can be used to compute the desired change in the controller parameters. Such an analytical expression can be obtained by observing that the neural network predictor and the controller are connected in series, and that the chain rule can be applied to determine the desired gradients. This is the basis of the backpropagation algorithm for training a multilayer perceptron network, and we are extending this gradient-based optimization technique to the controller gains. The predicted tracking error gradients are backpropagated through the predictor to determine the change in the gains that minimize  $V(k)$ . As usual, the minimization can only be guaranteed to achieve a local minimum because of the nonlinearities in the neural-network model structure.

Without loss of generality consider a general constant-gain dynamic controller for a single-input single-output (SISO) system.<sup>2</sup> The control input can be expressed in terms of the tracking error as follows:

$$u(k) = \sum_{i=0}^m A_i e(k-i) + \sum_{i=0}^p B_i u(k-i-1) \quad (3)$$

where

- $u(k)$  control input at time instant  $k$ ;
- $e(k)$  tracking error at time instant  $k$ ;
- $A_i$  gain value associated with corresponding error  $e(k-i)$ ;
- $B_i$  gain value associated with corresponding control input  $u(k-i)$ ;
- $m$  and  $p$  number of zeros and poles of the controller, respectively.

Equation (3) depicts the controller model in discrete-time input-output form.

Let us specialize to the most commonly used industrial controller structure, that is a PI controller. A PI controller in the discrete-time domain can be expressed as

$$u(k) = A_1 e(k) + A_2 e(k-1) + u(k-1). \quad (4)$$

Let  $\mathbf{A}$  denote the gain vector and  $\mathbf{e}$  the PI controller error vector, as follows:

$$\mathbf{A} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \quad \mathbf{e}(k) = \begin{bmatrix} e(k) \\ e(k-1) \end{bmatrix}. \quad (5)$$

Then, (4) can be rewritten as

$$u(k) = \mathbf{A}^T \mathbf{e}(k) + u(k-1). \quad (6)$$

During the optimization process, the tracking error  $\mathbf{e}(k)$  in (6) is replaced with the predicted tracking error  $\hat{\mathbf{e}}(k)$ , while during controller operation (6) is used as presented.

As shown in Fig. 2, the gains of the controller can be viewed as the adjustable parameters which “weigh” the error signal. Therefore, the PI controller equations can be implemented in the prediction loop using a neural network consisting of a linear node. The linear node receives as its inputs the weighted error signals and the control input at the previous time instant. The tuning of the gains can then be interpreted as the on-line adjustment of the weights associated with the error signals which are input to the linear node.

The gain adaptation algorithm used in this study is based on the backpropagation-through-time (BTT) approach [23], [32]. The BTT algorithm is an extension of the conventional backpropagation algorithm to training recurrent networks. In this method the time-iterations of the recurrent network are unfolded into layers. This means that for every new time-iteration, effectively one layer is added. The gradients are then backpropagated on this equivalent network as in the conventional backpropagation algorithm. Thus, we are effectively backpropagating the gradients from the final time  $T$  (corresponding to the last layer of the equivalent network) backward in time till time  $t = 1$  (corresponding to layer 1 of the equivalent network).

In our specific problem, let  $\delta(l)$  be the gradient vector of the error with respect to the network state vector at step  $l$ . The distinction from the BTT algorithm is that the  $\delta$ s are propagated through the considered prediction horizon only, starting from time  $k+H$ , and going back to step  $k$ , where  $k$  is the current time step. This means that first  $\delta(k+H)$ , that is the  $\delta$  associated

<sup>2</sup>This development is also applicable to multivariable controllers.

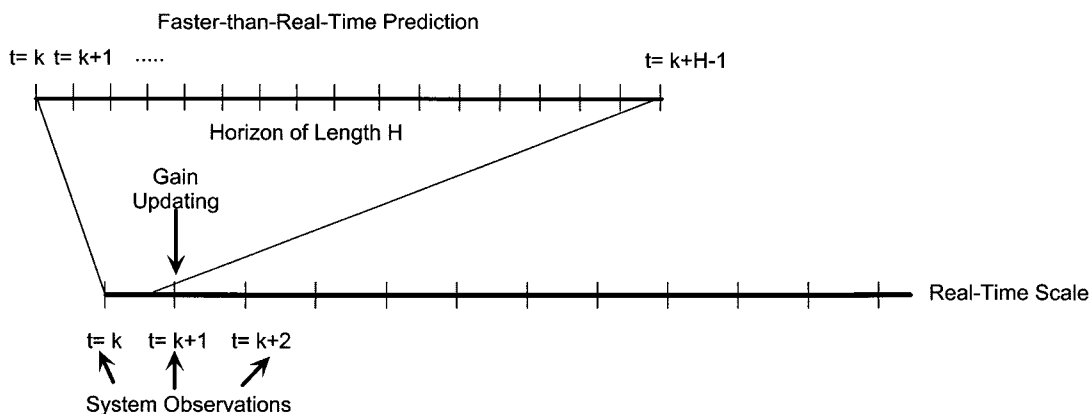


Fig. 3. Time scales involved in the controller implementation.

with the last period of the time horizon, is evaluated. Then backward propagation is performed, successively evaluating  $\delta(j)$  in terms of  $\delta(j+1)$ , until  $\delta(k)$  is obtained. The quantity  $\delta(k)$  represents the  $\delta$  value associated with the control input node in the first layer of the neural network during the first period of the prediction horizon. Note that whenever the control input (4) is encountered, it is considered as a “linear node” in the neural network.

Also note that this backpropagation procedure essentially follows from the successive application of the chain rule. Once  $\delta(k)$  has been obtained, the gradient of the predicted tracking error,  $V(k)$ , with respect to the controller gain vector is evaluated as

$$\frac{\partial V(k)}{\partial \mathbf{A}} = - \sum_{i=0}^{H-1} \delta(k+i) \mathbf{e}(k+i) \quad (7)$$

where  $\mathbf{e}(j)$  is given by (5), except that whenever  $\mathbf{e}(j)$  appears for  $j > k$  it is replaced by the predicted error  $\hat{\mathbf{e}}(j)$  rather than the actual error.

The neural network is thus critical to the overall gain adaptation algorithm, since it allows us to compute the sensitivities of the controller gains. The gradients are backpropagated through the neural network. It should be pointed out that in this study the “parallel learning architecture” is used instead of the commonly used “series-parallel learning architecture.” It is widely known in the literature that for dynamic backpropagation learning, the latter is preferred because of stability and convergence issues. In this study, however, the need to perform MSP necessitates the use of the “parallel learning” approach. No stability or other convergence problems have been encountered in this application.

For a given prediction time instant, the extent of the gain adaptation is computed using steepest descent, as follows:

$$\Delta \mathbf{A} = -\rho_{\mathbf{A}} \frac{\partial V(k)}{\partial \mathbf{A}} \quad (8)$$

where  $\rho_{\mathbf{A}}$  is the adaptation rate for updating the controller gains.

#### D. Controller Implementation Issues

The control action is arrived at by minimizing the predicted sum-squared tracking error over a finite time horizon. The pre-

diction over the finite horizon and the gain updates are performed faster-than-real-time by passing the control input computed by the simulated PI controller and the assumed disturbance sequence through the MSP neural network, i.e., by running the prediction loop.

As shown in Fig. 3, the future response of the neural network and the predicted tracking error are obtained at each time instant of the prediction horizon. This is accomplished by running the prediction loop faster-than-real-time, and by assuming a certain disturbance profile in the prediction horizon. The gradient term of (7) is summed-up over all future samples in the prediction horizon. All these computations are performed prior to the arrival of the next observation. Once a new observation arrives, it is used to compute the single-step prediction error using (7), and it is augmented with the MSP tracking error gradient sum obtained previously over the entire prediction horizon. Then, the updated controller parameters are calculated using a steepest descent rule using (8), and the results are implemented in the actual feedback controller, as well as in the simulated controller. This completes one algorithmic cycle.

At any given time, the new observations are used for two purposes. First, the single-step prediction error,  $e_{\text{supp}}(k)$ , is used to update the weights of the neural network and thereby enhance its predictive capability, if and when necessary. Second,  $e_{\text{supp}}(k)$  is also used to fine-tune the controller parameters using the generalized  $\delta$ -rule, (7). Therefore, the change in the controller parameters is a function of the predicted mean-squared tracking error over the prediction horizon, and the instantaneous tracking error between the observed and predicted responses. This minimizes the effects of model mismatch on the controller adaptation, and it also ensures that gain adaptation will continue even if the length of the prediction horizon is zero. Some inherent robustness is thus added to the control system and it is not necessary to continuously adapt the predictor for improved performance. The predictor weight adaptation can be performed intermittently, only when the predicted and actual responses diverge significantly.

#### E. Advantages of the Proposed Controller

The proposed algorithm encompasses some advantages of MPC, and some advantages of recurrent neural networks as they

relate to nonlinear system identification and derivation of the gain adaptation algorithm. The proposed method is characterized by the fact that tuning of the controller gains is carried out on-line using the identified process model, and without any expert supervision. The process need not be interrupted for tuning or resetting of the controller parameters. The controller parameters are adapted based on the predictive time-optimization of the system behavior, hence desirable process performance can be achieved over the entire operating envelope of the process and for long process operating periods. The proposed control architecture is suitable for processes with complex dynamics, necessitating identification prior to control. This is made possible by the neural MSP module, an excellent nonlinear mapping tool. The developed approach can be generalized to any linear or nonlinear controller structure, so long as the controller gradients can be computed. As opposed to other model-based control techniques, there is no need to explicitly invert the process model for computation of the desired control input. This is a distinct advantage since very often, a model may not be available in analytic form or it may not be well-behaved. Inherent robustness to drifts in process parameters and to modeling uncertainties is added to the conventional feedback loop by continuous on-line adaptation of the predictor, if necessary. Finally, the proposed retrofit, rather than the replacement of existing controllers, is a key advantage and it can be fully implemented in software.

### III. STEAM GENERATOR WATER LEVEL CONTROL

In this section, the physical control problem used to demonstrate the effectiveness of the proposed controller is described. The control problem chosen is that of a UTSG water level regulation in the presence of load variations and process drifts resulting from process tear-and-wear. This is a complex process control problem and proper water level control is critical during power plant operations. Current industrial practice is summarized, along with the process modeling approach used in this study. The section ends with a description of the water level controller development effort.

#### A. Control Problem Definition

A UTSG, a critical power plant component, is chosen as a representative process system for testing and evaluating the proposed neuro-predictive control algorithm. The UTSG is a highly complex process system, exhibiting severe nonlinearities and unstable, nonminimum phase open-loop response. In nuclear power plants, down-time can result in significant loss of revenue. It has been observed that a large number of nuclear power plant outages occur during plant start-up or at the low power range of operation, with about 60% originating from the feed-water, condensate and auxiliary feed-water systems. In addition, more than about half of these trips have directly resulted from the manual control of the UTSG water level [6]. As a result, serious efforts have been made to improve the closed-loop operational performance of the UTSG controller, in the low-power and start-up region. A schematic diagram of a UTSG with some of its characteristic physical dimensions is shown in Fig. 4.

The function of the UTSG water level control system is to maintain the water level in the section of the annulus surrounding the bottom of the separators at its programmed value, despite load changes. The downcomer water level is indicative of the water inventory in the secondary side of the UTSG, and this determines the amount of energy transferred from the primary to the secondary side of the UTSG. The water level cannot be allowed to rise too high since this would lead to excessive moisture carryover which could damage the turbine. The water level can neither be allowed to fall too much since this would uncover the feed-water ring and there could be the risk of a water hammer and associated hydrodynamic instabilities in the feed-water pipes. The UTSG water level control problem is complicated due to the reverse thermal dynamic effects of “*shrink*” and “*swell*”, which are more prominent at start-up and low power range of operation [6]. The water level, measured in the downcomer of the UTSG, temporarily reacts in a reverse manner in response to water inventory changes. This phenomenon is due to the two-phase mixture of steam and water present in the tube bundles. The only correct indication of the water inventory in the downcomer is the flow mismatch between the feed-water flow and the steam flow signals. However, at low operating powers (<20% of full power) and during start-up, the feed-water flow signal is highly corrupted by noise and it cannot be used in any automatic water level control system. Currently, during low power operations manual control is relied upon and the ensuing control action is prone to errors regardless of the operator’s skills and experience [6]. More recently, some power plants have added instrumentation, such as steam-dome pressure sensors, that are more indicative of the water inventory within the UTSG. Nevertheless, during manual control mode the operator very often over-estimates the effects of the shrink and swell, and the resulting control action is flawed. Other factors which make the control of the UTSG water level difficult are the constraints on the amount of feed-water flow dictated by the pump rating, that is actuator saturation, and the process parameter uncertainties and modeling inaccuracies which need to be considered at the controller design stage itself. Hence, an automatic control system is desired which could predict the nonminimum phase behavior and ensure satisfactory and desired process performance over the whole operating range of the UTSG, all this without any human interference.

#### B. Current Practice

The typical controller used in UTSG water level regulation is a PI controller, a subclass of the generic PID controllers. PID controllers are easy to implement and are adequate for controlling most process systems. Hence, they are widely used in a number of process industries. The integral control action reduces the steady-state error. The derivative control serves as an anticipatory action by taking remedial measures based on the gradient of the error signal. The derivative action is not desired in high noise environments, since it could cause the control action to be exaggerated by sensor noise. PID controllers can handle most control problems except in systems with significant dead-time, systems with significant oscillatory dynamics and systems with significant stochastic disturbances [8]. Some tuning of PID controller gains is carried out by experienced operators using practical rules and sometimes involves

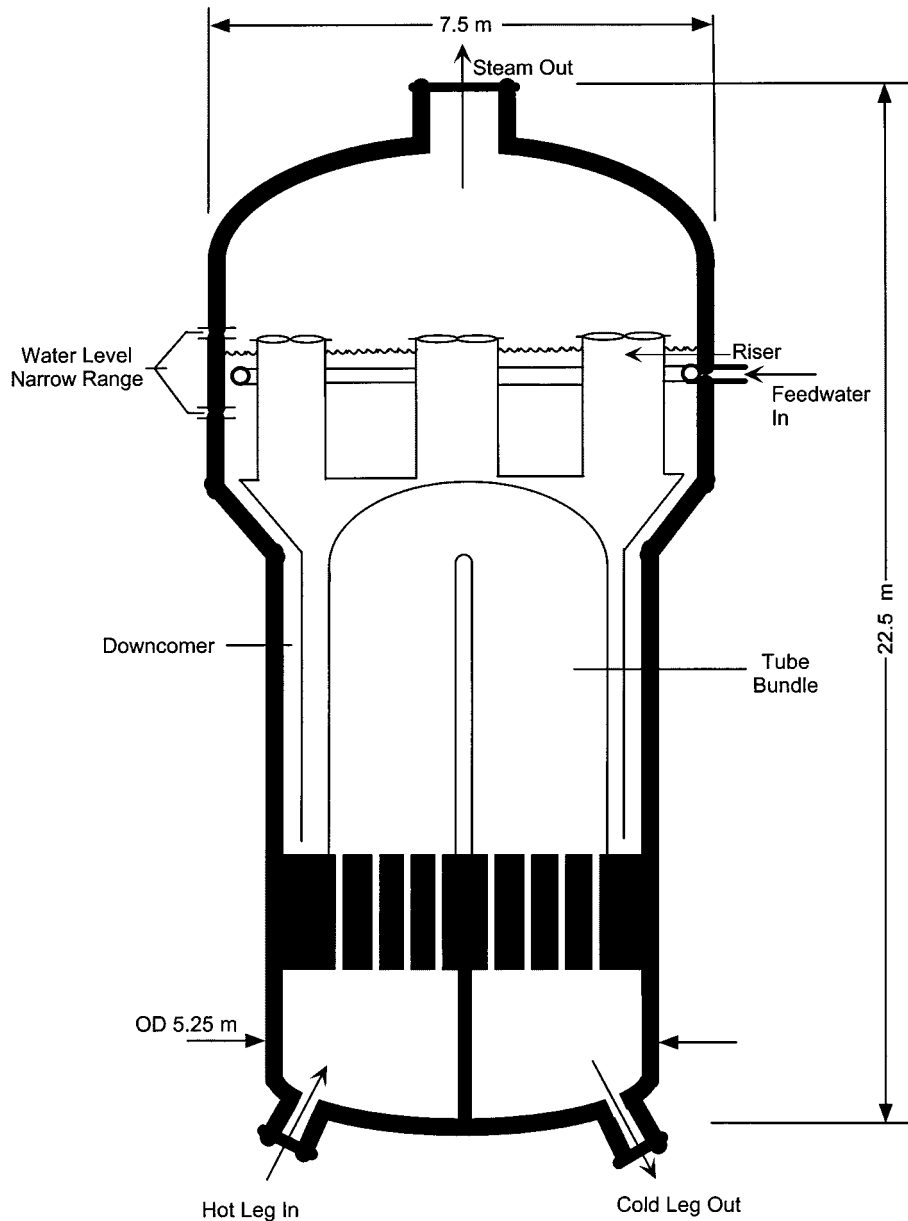


Fig. 4. U-tube steam generator schematic.

“trial-and-error” procedures [3]. Self-tuning PID controllers are also widely made available by numerous industrial control vendors, and they have found wide use in the process industries.

The so-called “three-element” controller, the PI controller used for UTSG water level regulation, determines the feed-water flow control signal, based on the three inputs it receives: the downcomer water level signal and the mismatch in the steam and feed-water flow signals. The feed-water flow is adjusted by varying the valve flow area in response to the control signal received from the controller. The steam flow rate is indicative of the operating power level. The analog equation for the control action of the “three-element” controller is

$$\begin{cases} \delta W_{fw}(t) = K_{pf} \left[ \epsilon_W(t) + \frac{1}{T_{if}} \int \epsilon_W(t) dt \right] \\ \quad + K_{pl} \left[ \epsilon_L(t) + \frac{1}{T_{il}} \int \epsilon_L(t) dt \right] \end{cases} \quad (9)$$

where

- $\delta W_{fw}(t)$  applied feed-water flow signal perturbation (kg/s);
- $\epsilon_W(t)$  mismatch in the feed-water and steam flow rate deviations;
- $\epsilon_L(t)$  water level deviation;
- $K_{pf}$  proportional flow gain;
- $K_{pl}$  proportional level gain;
- $T_{if}$  and  $T_{il}$  integral times for the flow mismatch and water level signals, respectively.

During low-power operation the flow mismatch signal is not available because of the high noise content of the feed-water flow rate. Water level regulation is then reverted to manual control, where most operator errors and process shut-downs are encountered.

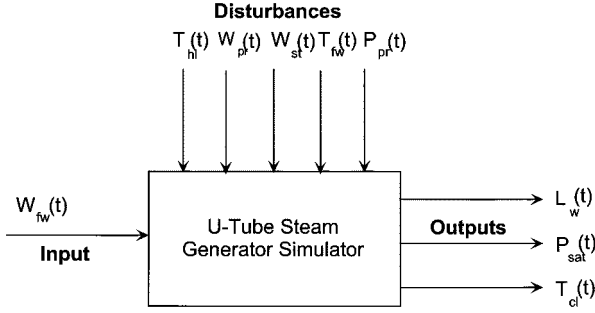


Fig. 5. U-tube steam generator block diagram.

### C. Process Modeling

In this study two models of the UTSG are utilized to demonstrate the controller performance. First, an accurate simulator is used to generate data characteristic of UTSG input and output measurements. The simulator is briefly described in this section. Second, a UTSG neural-network predictor is used as an integral part of the proposed controller; the predictor is briefly described in the next section.

Choi [6] modified an existing UTSG simulator developed by Strohmayer [30] to improve its low power simulation capabilities. The simulator was validated against actual process data and it was found to accurately simulate many of the severe operational transients encountered in the entire UTSG operating regime. The simulator was adopted for testing the proposed controller.

The UTSG simulator has three outputs: water level  $L_w(t)$ , secondary pressure  $P_{sat}(t)$ , and the cold-leg temperature  $T_{cl}(t)$ . The feed-water flow rate  $W_{fw}(t)$ , is the only control input to the UTSG. The five disturbances affecting the UTSG are the hot-leg temperature  $T_{hl}(t)$ , the primary flow rate  $W_{pr}(t)$  (always a constant to within small random variations), the steam flow rate  $W_s(t)$ , the feed-water temperature  $T_{fw}(t)$ , and the primary pressure  $P_{pr}(t)$  (almost always a constant). Changes in the power demand are translated to changes in the UTSG steam flow rate and this signal provides the persistent excitation needed for effective system identification. The hot-leg temperature and the feed-water temperature are usually expressed as functions of the operating power, and given the current operating power level they can all be calculated in a straightforward manner. Fig. 5 shows a block diagram of the UTSG simulator with the inputs, disturbances, and outputs shown.

The system of equations for the UTSG can be expressed using the following nonlinear state-space representation [20]:

$$\mathbf{E}(\mathbf{x}(t))\dot{\mathbf{x}}(t) = \mathbf{f}_{ps}(\mathbf{x}(t), u(t), \mathbf{w}(t)) \quad (10)$$

where,  $\mathbf{x}(t)$  is a twelve dimensional state vector, the controlled input  $u(t)$  is feed-water flow rate  $W_{fw}(t)$ , and the disturbance vector  $\mathbf{w}(t)$  consists of the five disturbances shown in Fig. 5 and an additional process noise term,  $\mathbf{v}(t)$ . The nonlinear matrix function  $\mathbf{E}(\cdot)$ , and the nonlinear vector function  $\mathbf{f}_{ps}(\cdot)$  contain very complex semiempirical expressions derived from the physics of the UTSG problem and other empirical correlations.

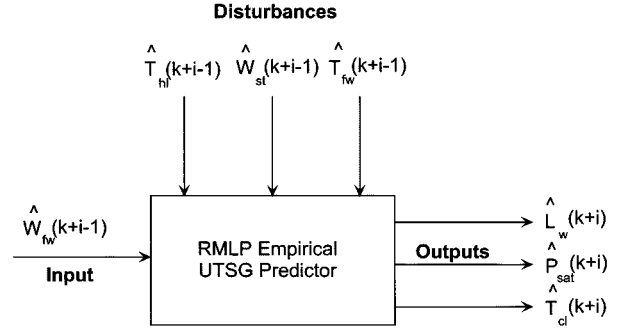


Fig. 6. UTSG FIR-type neuro-predictor block diagram.

The system of these nonlinear ordinary differential equations is solved to advance the transient simulation. The measured UTSG outputs are expressed as

$$\mathbf{y}(t) = \mathbf{h}_{ps}(\mathbf{x}(t), \mathbf{w}(t)) \quad (11)$$

with,

$$\mathbf{y}(t) = [L_w(t), T_{cl}(t), P_s(t)]^T \quad (12)$$

where  $\mathbf{h}_{ps}(\cdot)$  is a nonlinear vector function.

### D. Water Level Controller Development

The proposed controller consists of two major components: an existing PI controller and the adaptation part that consists of a predictor and a gain adaptation law, as shown in Fig. 1. In this study, the PI controller structure currently utilized in UTSG water level controllers is assumed. Any other controller structure could be incorporated with appropriate modifications to the gain adaptation algorithm.

As mentioned earlier, because of the high noise content of the feed-water flow sensor readings at low operating powers, the  $\epsilon_W$  shown in (9) cannot be utilized for effective water level control. Therefore, only the downcomer water level sensor reading is available for regulating the water level in the UTSG. The three-element controller can therefore be written in the discrete-time form as follows:

$$W_{fw}(k) = W_{fw}(k-1) + K \left[ 1 + \frac{\tau_s}{T_i} \right] \epsilon_L(k) - K \epsilon_L(k-1) \quad (13)$$

where

- $\tau_s$  sampling period;
- $K$  proportional controller gain;
- $T_i$  integral time constant.

Now, let us define the vector  $\mathbf{A}$  in terms of the PI controller parameters as

$$A_1 = K \left( 1 + \frac{\tau_s}{T_i} \right), \text{ and, } A_2 = K. \quad (14)$$

Then the change in the control action can be represented as a linear combination of the errors “weighted” by the gains  $A_1$  and  $A_2$ , as shown in Fig. 2.

The key component of the adaptation part of the proposed controller is the neuro-predictor module. It contains an empirical model, or a MSP predictor, which can be identified off-line

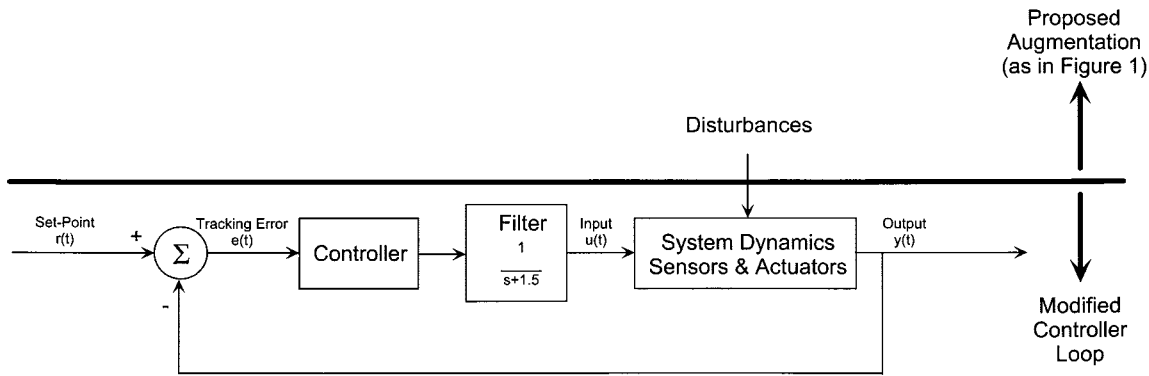


Fig. 7. Control loop modified with filter to eliminate feed-water flow and water level oscillations.

or on-line using only the measurements available to the controller. In this study, an empirical UTSG neuro-predictor developed by Parlos *et al.* [19] is utilized. This predictor uses a recurrent neural network in the form of a nonlinear finite impulse response (FIR) filter, and it was trained using a dynamic learning algorithm for tuning the weights and biases of the network. As reported by Parlos *et al.* [19], the developed neuro-predictor captured the UTSG model dynamics to sufficient accuracy over its entire operating envelope, i.e., from hot standby to full power operation. As an indication of the predictor accuracy, MSP simulations were performed and compared with the simulator response. Fig. 6 depicts a block diagram of the developed UTSG empirical model. In this figure time is expressed in the discrete-time domain because of the discrete-time nature of the predictor, and the predictor outputs are estimates of the UTSG outputs at times  $k + i$ .

The three elements of the proposed controller architecture, including the assumed controller structure, the identified predictor and the recursive weight update law, and the gain adaptation law have all been described and are now used in testing the controller effectiveness on the UTSG simulator. Initial values for the PI gain, the integral time constant and sampling period must be selected to allow UTSG operation prior to adaptation. The selection of the initial values is described in the next section.

#### IV. WATER LEVEL CONTROLLER PERFORMANCE EVALUATION

The proposed water level control system was tested using a process validated UTSG simulator previously discussed. The neuro-predictor as identified by Parlos *et al.* [19] was integrated with the gain adaptation module and the PI control loop as shown in Fig. 1. A baseline PI controller was developed for testing purposes. The selected PI gains were found to result in good performance for a number of simulation scenarios at high operating powers. At low operating powers, the constant gain PI controller was unable to provide good performance or even to stabilize the process, despite considerable amount of tuning over a wide range of gains. In this research we seek acceptable system transient performance over all the operating regime of the UTSG, via on-line gain adaptation of the standard PI controller parameters.

As previously defined, let  $\tau_s$  be the controller sampling interval and  $\rho_A$  be the controller gain adaptation rate. Further, let

$H$  denote the neuro-predictor horizon (consisting of a number of 10-s neural-network sampling intervals), and let  $\eta$  be the learning rate for the predictor weight update rule. Since the sampling interval used in the training phase of the predictor was 10 s, given the network states at time  $k$  and the network inputs at time  $k + 10$ , the predictor provides an estimate of the process output at time  $k + 10$ . The sampling interval of 10 s was arrived at by striking a balance between the computational burden for training the network, and the accuracy desired in capturing the UTSG dynamics. From a water level control point of view, a sampling interval  $\tau_s$  of 10 s results in satisfactory controller performance at high operating powers. However, at low operating powers a smaller  $\tau_s$  is required to stabilize the UTSG water level and to ensure good transient performance. For sampling intervals less than 10 s, the neuro-predictor states and the resulting outputs are computed using linear interpolation of the states and outputs computed at 10-s intervals. It should be noted that the predictor is a recurrent network and therefore has both dynamic states and outputs.

It has been our experience that the water level sensor exhibits minimal noise which can be removed via filtering. Therefore, during controller performance evaluation, it is assumed that the water level sensor noise does not significantly impact the control effort. Hence, water level sensor noise is not considered in the simulations presented. Also, unless otherwise specified, on-line adaptation of the neuro-predictor is not used. Therefore, the predictor is not updated during the transients simulated. In order to filter out the mild oscillations captured by the water level sensor, which are thought to be caused by the fluctuations in the feed-water flow rate, a low-pass filter of the form

$$F(s) = \frac{1}{s + 1.5} \quad (15)$$

was incorporated at the UTSG input, as shown in Fig. 7.

Extensive simulation studies have been performed with and without actuator noise. Since the former cases are more severe and more representative of the operating environment of the UTSG, only these results are described below. In the summary table presented in later parts of this paper, results without actuator noise are also included for comparison purposes. A more extensive set of simulations for the proposed controller can be found in [22]. As expected, the developed adaptive PI water level controller consistently yields better system performance

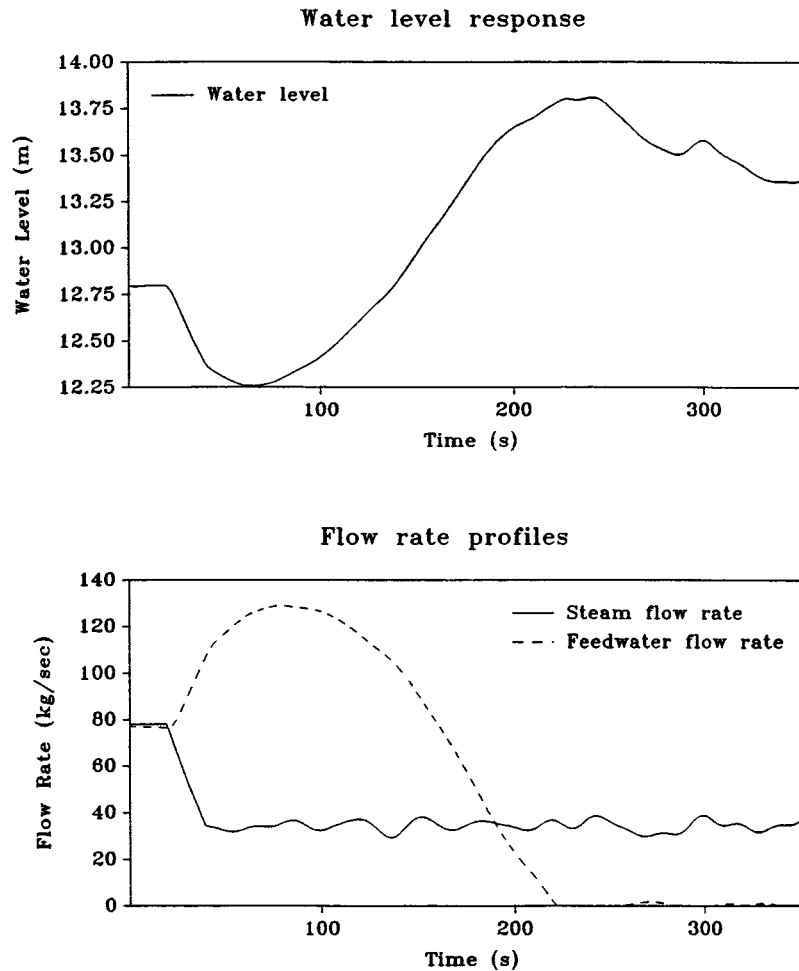


Fig. 8. Constant-gain controller response to step decrease in power demand from 20% to 10% of full power.

than the constant-gain PI controller currently used in UTSG's. A constant-gain PI controller was quite adequate for controlling the water level for most scenarios at high operating powers (that is operating power  $>20\%$  of full power). However, as it will be discussed during the presentation of the performance summary table, there are some high operating power transients for which the constant-gain PI controller is unable to stabilize the water level. At high operating power, the difference between simulations with and without actuator and process noise is quite insignificant. This is because the signal-to-noise ratio (SNR) at high operating powers is significant, i.e., the noise content in the signal is a very small fraction of the signal strength. As expected, at low operating powers the presence of actuator signal noise adds to the instability of the UTSG. At low operating powers, the constant gain PI controller was able to stabilize the UTSG for increases in power demand. However, the constant-gain controller transient performance was not considered satisfactory. For decreases in power demands in this operating range, the constant-gain PI controller was not able to stabilize the process at all. As discussed in the following paragraphs, the adaptive PI controller was able to stabilize the UTSG with quite acceptable transient performance throughout the low, as well as the high power operating regime [22]. The simulation results of the constant-gain controller quite often consist of unstable transients and the resulting plots are outside the desired scale.

The figures presented in this study exclude significant portions of these instabilities for ease in comparing transient responses. Whenever appropriate, water level oscillations have been included to demonstrate the desired argument. However, as stated in the following paragraphs, in some transients the wild variations in water level response have been eliminated.

Fig. 8 shows the UTSG response to a step decrease in power demand from 20% to 10% of full power with the PI controller parameters set at  $K = 110$  and  $T_i = 110$ . These parameters were chosen as the best parameters that lead to the stabilization of the largest region in the high power range. The feed-water flow is unable to recover from its initial saturation, occurring at about 270 s, for a significant amount of time and this causes the UTSG water level to behave erratically. It has been determined that it is critical for the feed-water flow to recover soon after the initial saturation to zero, else the controller fails in stabilizing the process. In this scenario the water level becomes unstable causing "trip" of the UTSG resulting in plant shut-down. In Fig. 8, the water level response beyond 350 s increases rapidly outside the plotting scale and it has been eliminated.

Starting with the same initial controller parameters as in Fig. 8, and following suitable adaptation of the controller gains using the proposed adaptation method, the water level response is vastly improved for the same simulation experiment. The horizon length is set at 100 s, and the control action is com-

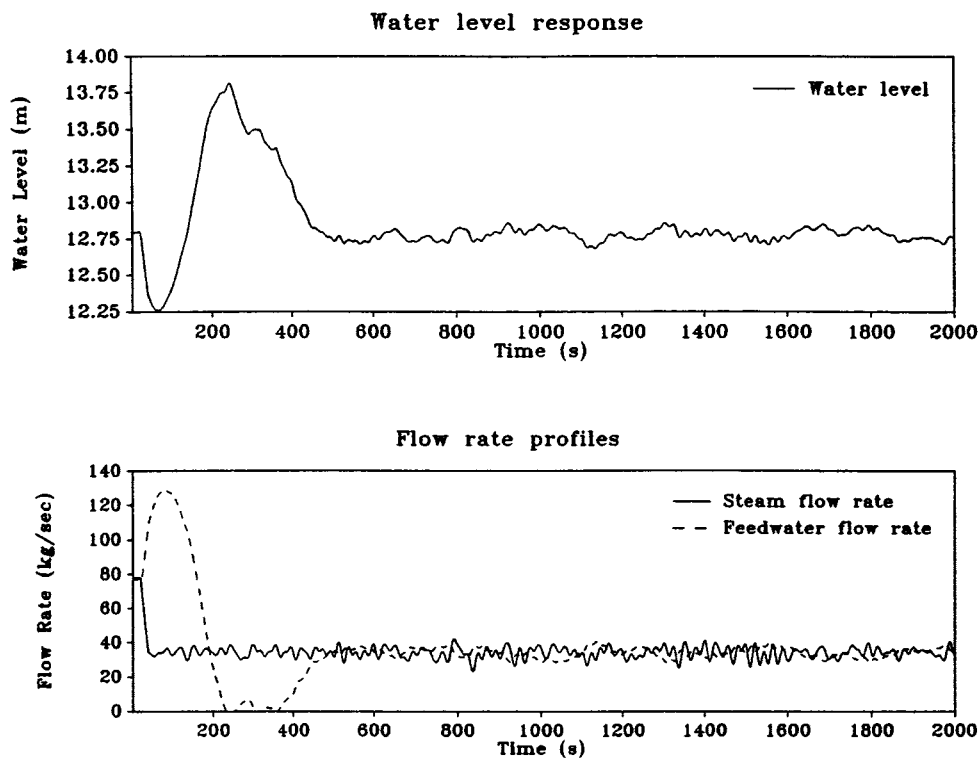


Fig. 9. Adaptive-gain controller response to step decrease in power demand from 20% to 10% of full power.

puted every 0.1 s. The process settles down in 436 s and the peak-to-peak values of the water level excursions are well within the “trip” set-points. The response of the closed-loop system for this case study is shown in Fig. 9.

Next, a ramp-down in power from 95% to 5% of full power was simulated. The ramp-down rate is 6% per minute from 95% to 20% of full power, and 1% per minute down to 5% of full power. The change in the ramp-down rate was implemented to reflect the limitations imposed by plant operational safety considerations. It was determined that the constant-gain PI controller could not stabilize the process for this frequently encountered operational scenario. The water level response and the flow rate profiles using the constant-gain PI controller are depicted in Fig. 10. As seen in this figure, even following termination of the power demand change, which occurs at approximately 2000 s, the water level oscillations are very significant. In fact, following 4000 s the water level response becomes completely unstable. Using suitable gain adaptation for the above transient, it was determined that the adaptive PI controller stabilized the UTSG, and the process settled down in approximately 2577 s. The water level response and the flow rate profiles using the adaptive PI controller are shown in Fig. 11. The observed mild oscillations in the water level response of the adaptive PI controller are within the range of the steady-state oscillations of the actual plant data at low power operation and they do not usually result in planned shut-down.

A final simulation experiment is mentioned that intends to explore the robustness of the developed controller to model parameter uncertainties. However, detailed transient simulation responses are not presented. To test the robustness of the developed controller to process parameter drifts, a modeling uncer-

tainty was introduced into the UTSG simulator in the form of a perturbation in the value of the tube fouling factor. This variable characterizes the ability of the UTSG tubes to efficiently transfer thermal energy. As the UTSG ages, the fouling factor drifts. The simulation experiment consisted of a ramp-up in power from 10% to 25%. The ramp-up rate is 1% per minute, and the value of the tube fouling factor is perturbed by about 4%, a perturbation well within expected changes in the lifetime of the UTSG. The constant-gain PI controller failed in stabilizing the UTSG. With on-line predictor identification enabled, the adaptive PI controller was able to stabilize the UTSG in 1408 s. The controller was deemed to yield robust performance in view of process parameter drifts.

A summary of the adaptive PI controller performance is presented in Table I. Some of the transients included in the table were previously described in detail, whereas others were not. For the first transient tabulated, the performance of the adaptive controller has been compared with that of the constant-gain PI controller at high operating power. The constant PI gains were obtained via coarse tuning of the controller parameters but the controller was unable to stabilize the process. Both simulations included actuator noise. The next two transients are for the low operating power range. Transients without actuator noise are presented, followed by transient scenarios with colored actuator noise. The last transient tabulated is a ramp decrease from 95% to 5% of full power with actuator noise. The results indicate that the adaptive PI controller, as expected, consistently outperforms the constant-gain PI controller. In fact, in all of the simulations presented in Table I, the constant-gain PI controller resulted in unstable closed-loop response. The gain adaptation can ensure stable response and good process performance over

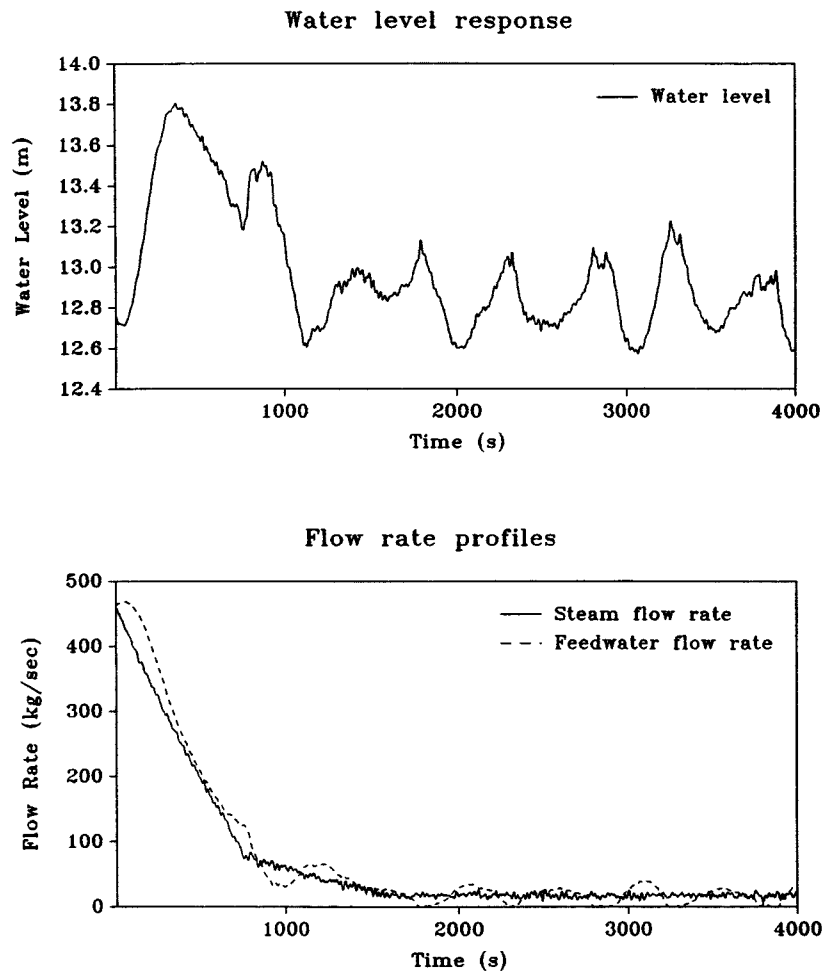


Fig. 10. Constant-gain controller response to ramp decrease in power demand from 95% to 5% of full power, at 6% per minute.

the entire operating range of the process. We should point out to the reader that comparing a constant-gain PI controller with an adaptive PI controller containing a complex adaptation scheme may not seem very reasonable. However, in this study we are attempting to solve a real industrial control problem, and we are comparing our proposed solution with current industrial practice. Other, more sophisticated control methods have been applied to this problem. Such solutions might not be easily embraced by industrial practitioners [20].

## V. DISCUSSION AND CONCLUSION

In this research, an algorithm has been presented for process control via the continuous on-line adaptation of linear controller parameters based on the concept of neuro-prediction. Recurrent neural networks are used to develop an MSP for the system to be controlled. Furthermore, a steepest descent controller gain adaptation law is derived using the “parallel learning” architecture, necessitated by need to perform MSP. The algorithm is tested on a complex process system simulator, namely a UTSG, and the performance of the adaptive PI controller is extensively studied. Through the use of simulations, it is determined that the adaptive PI controller can successfully control the system via gain adaptation. At every sampling instant, the gains of the PI controller are tuned in the negative direction of the prediction

error gradients, akin to the *training* of neural networks. In most high power transients, a marginal decrease in the peak-to-peak excursions of the water level is observed, though in some instances the adaptive PI controller stabilized otherwise unstable transients even in the high operating regime. However, the main benefit of the gain adaptation algorithm has been its ability to consistently stabilize the process at low operating powers, and to ensure satisfactory transient system performance over the entire operating envelope of the process.

The effectiveness of the adaptive PI controller to stabilize and improve the performance of the UTSG at low operating powers deserves some further insight. There are two main reasons for the instabilities caused at low operating powers by the three-element controllers currently utilized in UTSGs. First, as the low power region is approached the feed-water flow rate signal becomes excessively noisy resulting in a noisy flow mismatch signal. The flow mismatch signal is the instantaneous difference between the feed-water and steam flow rates, and it is the only signal available to the three-element controller indicative of the true water mass inventory in the UTSG. The second reason for the instabilities is the nonminimum phase behavior of the measured water level signal which is exceedingly pronounced in the low power operating region.

The standard three-element controller cannot make use of the flow mismatch signal during low power operation because of its

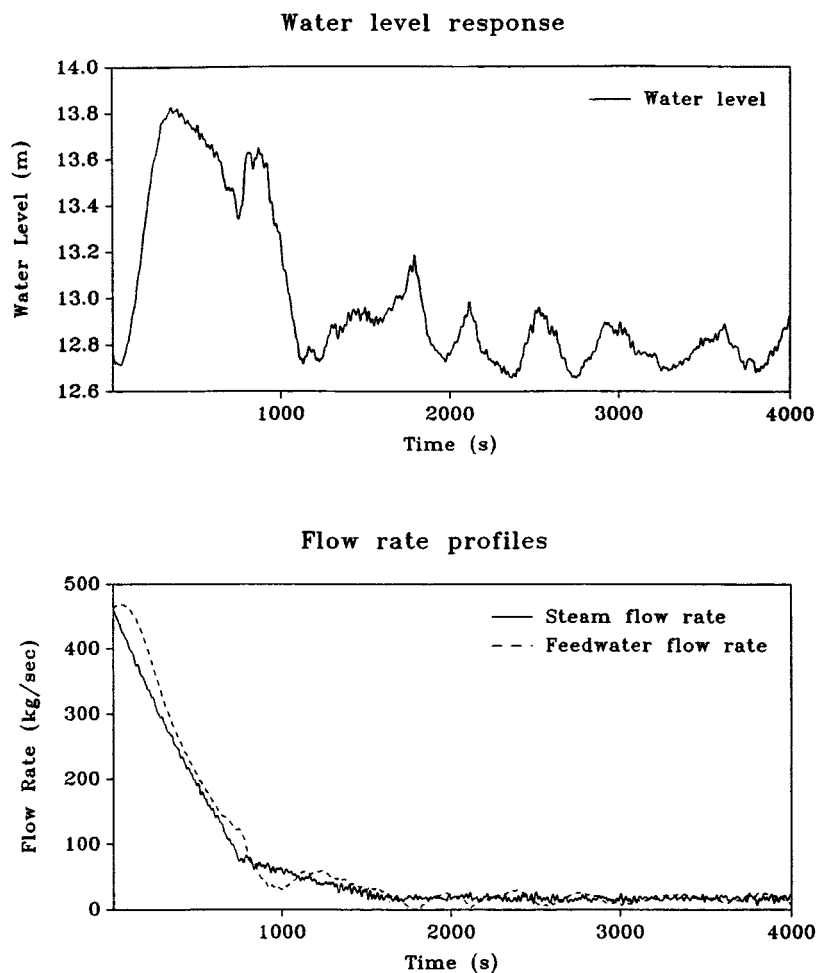


Fig. 11. Adaptive-gain controller response to ramp decrease in power demand from 95% to 5% of full power, at 6% per minute.

TABLE I  
PERFORMANCE FOR PROPOSED ADAPTIVE PI CONTROLLER

Transient <sup>3</sup>	Noise <sup>4</sup>	Peak-to-Peak (m)		Settling Time (s)
		max	min	
Case I <sup>5</sup>	Y	12.88	11.84	1025
Case II <sup>6</sup>	N	13.81	12.26	448
	Y	13.81	12.26	436
Case III <sup>7</sup>	N	13.36	12.54	833
	Y	13.35	12.54	487
Case IV <sup>8</sup>	Y	13.83	12.66	2577

<sup>3</sup> Simulations without adaptation result in unstable responses.

<sup>4</sup> This represents actuator noise.

<sup>5</sup> 60 % to 70 % of Full Power Ramp @ 9 %/min.

<sup>6</sup> 20 % to 10 % of Full Power Step.

<sup>7</sup> 10 % to 5 % of Full Power Ramp @ 6 %/min.

<sup>8</sup> 95 % to 5 % of Full Power Ramp @ 6 %/min.

noise content. As a result, the severe nonminimum phase response of the water level measurement prevents the three-el-

ement controller from stabilizing the system. The controller, with only proportional and integral water level measurement terms, lacks the predictive capability to anticipate the reverse dynamics of the water level, resulting in instabilities. The proposed gain adaptation of the three-element controller utilizes a process model for predicting the presence and severity of the nonminimum phase water level response through accurate MSP. As a result of this anticipatory component of the controller, the gains are appropriately adjusted to minimize the predicted water level regulation error, compensating for its nonminimum phase behavior. The nonminimum phase water level response cannot be eliminated, but the water level oscillations it causes are damped, preventing instabilities. One final observation is related to the presence of severe nonlinearities in the UTSG dynamics, as obtained from a linearization error analysis. The presence of such nonlinearities compounds the complexity of the UTSG water level control at low powers. The availability of an on-line process model capable of accurate MSP becomes a necessity for effective water level control at low powers.

The algorithm presented in this research allows for on-line predictor tuning, improving the MSP capabilities of the developed neuro-predictor. The accuracy of the predictor can be improved as new observations become available, thereby lending robustness to the controller architecture. Thus, drifts in process parameters due to aging, corrosion, material defects, etc. can

be easily handled by fine-tuning the predictor, if and when required. One of the important outcomes of this study has been the successful use of controller sampling intervals that are less than the predictor sampling interval. The neuro-predictor sampling interval is arrived at by considering the size of the training set, the CPU time required to train the network, and the minimum number of data required to identify the process dynamics to within a desired accuracy. If the control sampling interval is smaller than the neuro-predictor sampling interval, then a linear interpolation of the states of the network, between the network sampling intervals, is used for prediction and for gain adaptation. Because of the slow process dynamics of the UTSG involved, this interpolation resulted in quite acceptable results.

The proposed control system architecture comprises of an auxiliary sub-system which can be easily augmented to existing process controller feedback loops. The sub-system could be implemented on microprocessors, microcontrollers or even on dedicated DSP cards. Considering the slow dynamics of the UTSG, the gain adaptation algorithm comprises of simple algebraic calculations which are not computationally very intensive. A study on the time requirements showed that the algorithm can easily be incorporated within a real-time environment with reasonable prediction horizon lengths, even for very small control sampling intervals. As with any other on-line optimization control algorithm, the major limitation of the presented control approach is that the process dynamics must vary slow enough, so as to allow for faster-than-real-time prediction of a horizon with reasonable length. At the same time, the increasing speed of industrial computing platforms will enable the application of the proposed algorithm to processes with faster dynamics.

## REFERENCES

- [1] M. Agarwal, "A systematic classification of neural-network-based control," *IEEE Contr. Syst. Mag.*, pp. 75–93, April 1997.
- [2] A. F. Atiya and A. G. Parlos, "New results on recurrent network training: Unifying the algorithms and accelerating convergence," *IEEE Trans. Neural Networks*, vol. 11, pp. 697–709, May 2000.
- [3] K. J. Åström and B. Wittenmark, *Adaptive Control*, 2nd ed. Reading, MA: Addison-Wesley, 1995.
- [4] H. Bersini and V. Gorrini, "A simplification of the backpropagation-through-time algorithm for optimal neurocontrol," *IEEE Trans. Neural Networks*, vol. 8, pp. 437–441, Mar. 1997.
- [5] N. Bhat and T. J. McAvoy, "Use of neural networks for dynamic modeling and control of chemical process systems," *Comput. Chem. Eng.*, vol. 14, no. 4/5, pp. 573–583, 1990.
- [6] J. J. Choi, "Nonlinear digital computer control for the steam generator system in a pressurized water reactor plant," Ph.D., MIT, Cambridge, MA, 1987.
- [7] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [8] T. Häggglund and K. J. Åström, "Industrial adaptive controllers based on frequency response techniques," *Automatica*, vol. 27, no. 4, pp. 599–609, 1991.
- [9] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.
- [10] M. V. Kothare, B. Mettler, M. Morari, P. Bendotti, and C.-M. Falinower, "Level control in the steam generator of a nuclear power plant," *IEEE Trans. Contr. Syst. Technol.*, vol. 8, pp. 55–69, Jan. 2000.
- [11] D. M. McDowell, G. W. Irwin, G. Lightbody, and G. McConnell, "Hybrid neural adaptive control for bank-to-turn missiles," *IEEE Trans. Contr. Syst. Technol.*, vol. 5, pp. 297–308, May 1997.
- [12] T. W. Miller, R. S. Sutton, and P. J. Werbos, *Neural Networks for Control*. Cambridge, MA: MIT Press, 1990.
- [13] P. Mills, A. Zomaya, and M. Trade, "Adaptive model-based control using neural networks," *Int. Journal Control*, vol. 60, no. 6, pp. 1163–1192, 1994.
- [14] M. Morari, *Robust Process Control*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [15] K. F. Muller and W. F. Schaefer, "Self-tuning control technology for nuclear power plants: Final Rep.," EPRI TR-101 650, 1993.
- [16] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4–28, Mar. 1990.
- [17] K. S. Narendra, "Neural networks for control: Theory and applications," *Proc. IEEE*, vol. 84, pp. 1385–1406, Oct. 1996.
- [18] S. Omatu and M. Yoshioka, "Self-tuning neuro-PID control and applications," in *1997 IEEE Int. Conf. Syst., Man, Cybern.*, vol. 3, 1997, pp. 1985–1989.
- [19] A. G. Parlos, K. T. Chong, and A. F. Atiya, "Application of the recurrent multilayer perceptron in modeling complex process dynamics," *IEEE Trans. Neural Networks*, vol. 5, pp. 255–266, Mar. 1994.
- [20] A. G. Parlos and O. T. Rais, "Nonlinear control of U-tube steam generators via  $H_\infty$  Control," *Contr. Eng. Practice*, vol. 8, pp. 921–936, 2000.
- [21] A. G. Parlos, O. Rais, and A. F. Atiya, "Multistep-ahead prediction in complex systems using dynamic recurrent neural networks," *Neural Networks*, 2000, to be published.
- [22] S. Parthasarathy, "Model predictive adaptive control of process systems using recurrent neural networks," M.S. thesis, Texas A&M Univ., College Station, TX, 1993.
- [23] B. Pearlmutter, "Learning state-space trajectories in recurrent neural networks," *Neural Comput.*, vol. 1, pp. 263–269, 1989.
- [24] G. Prasad, E. Swidenbank, and B. W. Hogg, "A neural net model-based multivariable long-range predictive control strategy applied in thermal power plant control," *IEEE Trans. Energy Conv.*, vol. 13, no. 2, pp. 176–182, June 1998.
- [25] D. Psychogios and L. Ungar, "Direct and indirect model based control using artificial neural networks," *Ind. Eng. Chem. Res.*, vol. 30, pp. 2564–2573, 1991.
- [26] G. V. Puskorius and L. A. Feldkamp, "Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks," *IEEE Trans. Neural Networks*, vol. 5, pp. 279–293, Mar. 1994.
- [27] G. V. Puskorius, L. A. Feldkamp, and L. I. Davis Jr, "Dynamic neural-network methods applied to on-vehicle idle speed control," *Proc. IEEE*, vol. 84, no. 10, pp. 1407–1420, Oct. 1996.
- [28] J. Saintdonat, N. Bhat, and T. McAvoy, "Neural-net-based model predictive control," *Int. J. Contr.*, vol. 54, no. 6, pp. 1453–1468, 1991.
- [29] T. Samad and H.-T. Su, "On the optimization aspects of parametrized neurocontrol (PNC) design," *IEEE Trans. Comp., Packag., Manufact. Technol. C*, vol. 19, pp. 27–36, Jan. 1996.
- [30] W. H. Strohmayer, "Dynamic modeling of vertical U-tube steam generators for operational safety systems," Ph.D. dissertation, MIT, Cambridge, MA, 1982.
- [31] F. Wang, L. Mingzhong, and J. Wang, "Adaptive Generalized Predictive Control for Nonlinear Systems Using Neural Networks," in *Proc. IEEE Int. Conf. Ind. Technol.*, 1996, pp. 806–810.
- [32] P. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, Oct. 1990.
- [33] "Brain-like intelligent control: What it does and How to do it," *Proc. 36th Conf. Decision Contr.*, pp. 3902–3904, Dec. 1997.
- [34] J. Zhan and M. Ishida, "The multistep predictive control of nonlinear SIS processes with a neural model-predictive control (NMPC) method," *Comp. Chem. Eng.*, vol. 21, no. 2, pp. 201–210, 1997.



**Alexander G. Parlos** (S'81–M'86–SM'92) received the B.S. degree in nuclear engineering from Texas A&M University, College Station, in 1983, and then the S.M. degree in nuclear engineering, the S.M. degree in mechanical engineering, and the Sc.D. degree in automatic control and systems engineering, all from Massachusetts Institute of Technology, Cambridge, in 1985, 1985, and 1986, respectively.

He has been on the faculty at Texas A&M University since 1987, where he is currently an Associate Professor of Mechanical Engineering, with joint appointments in the Department of Nuclear Engineering and (by courtesy) the Department of Electrical Engineering. His research interests include the development of data-driven methods and algorithms for life-cycle health and performance management of dynamic systems. He has applied these concepts to

electro-mechanical systems and more recently to computer networks. His theoretical research interests involve the development of learning algorithms for recurrent neural networks and their use in nonlinear estimation and control. His research has resulted in one patent, two pending patents, and 16 invention disclosures. He has more than 110 publications in journals and conferences, and he is serving as an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS since 1994.

Dr. Parlos is a Senior Member of AIAA, a member of ASME, INNS, and a registered professional engineer in the State of Texas. He has been an Associate Editor of the *Journal of Control, Automation and Systems* since 1999.



**Sanjay Parthasarathy** received the B.Eng. degree in mechanical engineering from the University of Bombay, India, in 1990 and the M.S. degree in nuclear engineering from Texas A&M University, College Station, in 1993. He is currently pursuing the doctoral degree with the Mechanical Engineering Department, University of Minnesota, Minneapolis.

Since 1993, he has been employed at the Honeywell Technology Center's Systems and Controls Laboratory, where he has specialized in advanced controls research for manufacturing technologies and for industrial process automation. He has been the technical and program lead for many materials processing projects funded by governmental agencies. His research interests include nonlinear systems, adaptive control, data-centric technologies, real-time control, and software development.



**Amir F. Atiya** (S'86-M'90-SM'97) received the B.S. degree in 1982 from Cairo University, Cairo, Egypt, and the M.S. and Ph.D. degrees in 1986 and 1991 from California Institute of Technology, Pasadena, all in electrical engineering.

From September 1990 to July 1991, he was a Research Associate with Texas A&M University, College Station. From July 1991 to February 1993, he was a Senior Research Scientist with QANTXX, Houston, TX, a financial modeling firm.

In March 1993, he joined the Computer Engineering Department at Cairo University as an Assistant Professor. Since March 1997, he has been a Visiting Associate in the Department of Electrical Engineering at Caltech. His research interests include neural networks, learning theory, pattern recognition, Monte Carlo methods, data compression, and optimization theory. His most recent interests are the applications of learning theory and computational methods to finance. He has been an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS since 1998.

Dr. Atiya received the *Egyptian State Prize for Best Research in Science and Engineering*, in 1994. He also received the *Young Investigator Award* from the *International Neural Network Society*, in 1996. He is a Co-Guest Editor of the forthcoming special issue of IEEE TRANSACTIONS ON NEURAL NETWORKS on Neural Networks in Financial Engineering. He served on the organizing and program committees of several conferences.