LA-UR-02-5133

*Approved for public release; distribution is unlimited.*

*Title:* A Neural Network Based Approach for Tuning of SNS Feedback and Feedforward Controllers

*Author(s):* Sung-il Kwon
Amy Regan

*Submitted to:* LINAC 2002

# A NEURAL NETWORK BASED APPROACH FOR TUNING OF SNS FEEDBACK AND FEEDFORWARD CONTROLLERS *

Sung-il Kwon† and Amy Regan, LANL, Los Alamos, NM 87545, USA

*Abstract*

The primary controllers in the SNS low level RF system are proportional-integral (PI) feedback controllers. To obtain the best performance of the linac control systems, approximately 91 individual PI controller gains should be optimally tuned. Tuning is time consuming and requires automation. In this paper, a neural network is used for the controller gain tuning. A neural network can approximate any continuous mapping through learning. In a sense, the cavity loop PI controller is a continuous mapping of the tracking error and its one-sample-delay inputs to the controller output. Also, monotonic cavity output with respect to its input makes knowing the detailed parameters of the cavity unnecessary. Hence the PI controller is a prime candidate for approximation through a neural network. Using mean square error minimization to train the neural network along with a continuous mapping of appropriate weights, optimally tuned PI controller gains can be determined. The same neural network approximation property is also applied to enhance the adaptive feedforward controller performance. This is done by adjusting the feedforward controller gains, forgetting factor, and learning ratio. Lastly, the automation of the tuning procedure-data measurement, neural network training, tuning and loading the controller gain to the DSP is addressed.

## 1 INTRODUCTION

Neural networks are composed of massively connected neurons [1]. With their structures resembling more or less their biological counterparts, artificial neural networks are representational and computational models composed of interconnected simple processing elements called artificial neurons. In processing informations, processing elements works concurrently and collectively in parallel and distributed fashion. Neural network research stemmed from McCullen and Pitts' pioneering work [1] a half-century ago. Since then, numerous neural networks have been developed and extended their applications from pattern recognition, optimization, to control, dynamic system identification, prediction.

Neural networks have very close ties with optimization. Many learning algorithms have been developed based on optimization techniques such as least mean square algorithm and steepest decent algorithm. Neural networks learn from examples rather than having to be programmed in a conventional sense. In these senses, neural networks resemble the adaptive control/signal processing. When the system's complexity increase, the controller parameter tuning are burdensome and time consuming job. In SNS, there are 81 SRF cavities and each cavity has its own controller. For simplicity and effectiveness, PID controller is selected as the feedback controller. Each cavity has different RF parameters and in order to achieve the satisfactory feedback control, each controller has to be optimally tuned. A classical tuning such as Ziegler-Nichols method can be applied but for that, some data are to be measured beforehand. Also, adaptive PID controller can be used where the PID controller gains are adaptively tuned.

In this paper, a PID controller tuning method based on neural network is investigated. The difference of neural network tuning PID controller from adaptive PID controller is that as the system's complexity increases, the computational complexity of the former does not increase much. In order to verify the effectiveness and accelerate the real implementation, MATLAB/SIMULINK model is developed.

## 2 NEURAL NETWORK TUNING PID CONTROLLER

The neural network tuning discrete time PID controller configuration is shown in figure 1. The discrete time PID controller is given by

$$u(t)=u(t-1)+k_p(e(t)-e(t-1))+k_ie(t)$$
$$+k_d(e(t)-2e(t-1)+e(t-2)) \quad (1)$$
$$e(t)=r(t)-y(t) \quad (2)$$

where $t$ is the sample number. The tuning algorithm of the discrete time PID controller is obtained by minimizing the cost function given by

$$E=\frac{1}{2}e^2(t+1). \quad (3)$$

For PID gain tuning, three layered backpropagation neural network is considered. Input layer has three nodes which correspond to error $e(t)$ and its delays $e(t-1)$ and $e(t-2)$ Hidden layer has $N_H$ nodes and the transfer function of each node is Sigmoid function $f(x)=\dfrac{1.0}{1.0+e^{-x}}$ and its derivative with respect to $x$ is $f'(x)=f(x)(1-f(x))$. Output layer has 3 nodes, each corresponds to $k_p$, $k_i$, and $k_d$, and the transfer function of each node is Sigmoid function.

* support by Department of Energy of USA
† swkon@lanl.gov

## 4 REFERENCES

[1] G. W. Ng, *Application of Neural Networks to Adaptive Control of Nonlinear Systems*, John Wiley & Sons INC. , New York, 1997.

[2] W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Math. Biophysics*, vol 5, pp.115-143, 1943.


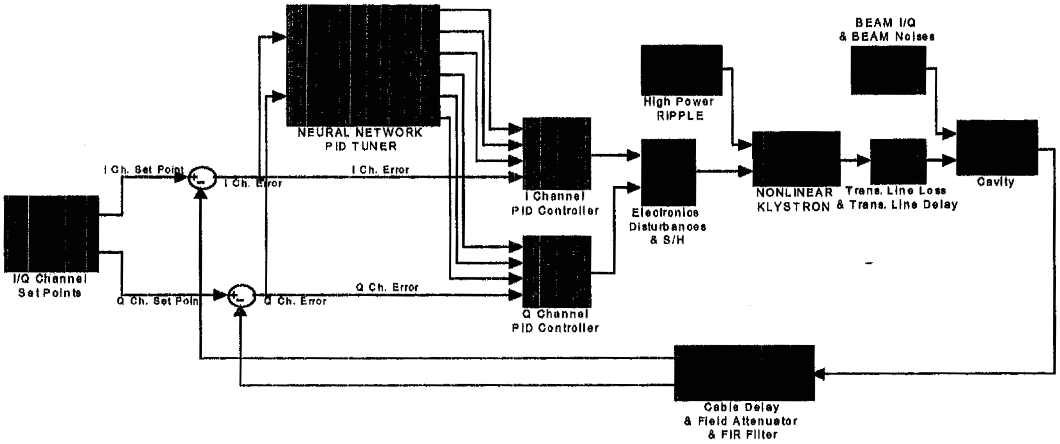


**Figure 2**: Neural Network Tuning PID Control System.

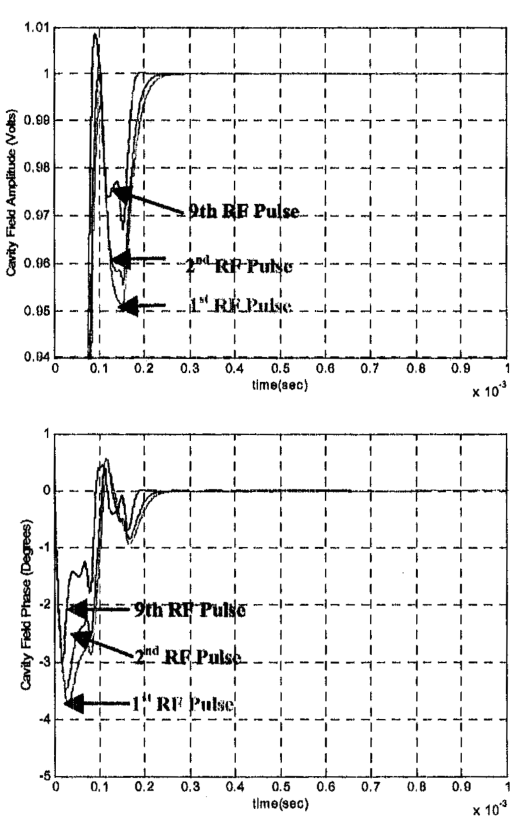


**Figure 3**: Cavity Field Amplitude (upper) and Cavity Field Phase (Bottom). Note that as the RF pulse number increases, the performance is improved.

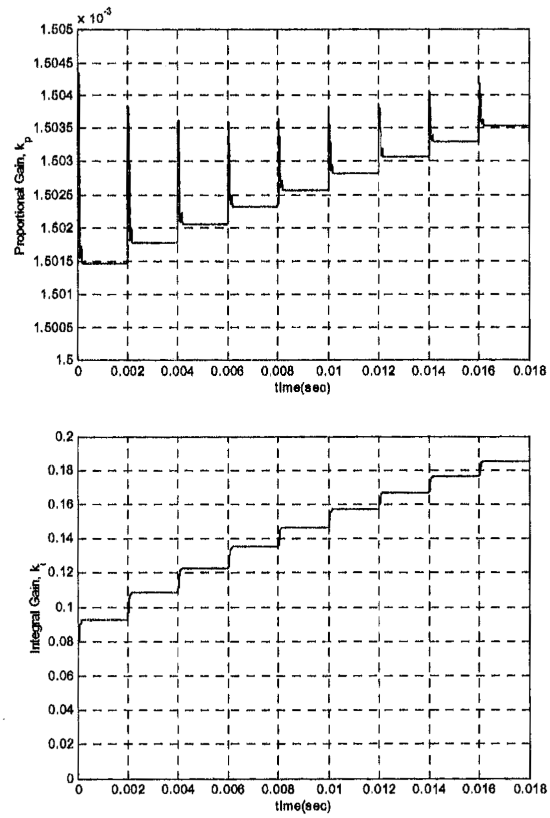


**Figure4**: Proportional gain(upper) and Integral gain (Bottom) of Neural Network Tuning PI Controller.

**Input Layer Nodes:** $n_i(t) = e(t-i)$ (4)

**Hidden Layer Nodes:** $n_j(t) = f(\sum_{i=0}^{2} w_{j,i}(t) \cdot n_i(t))$ (5)

**Output Layer Nodes:** $n_k(t) = f(\sum_{j=0}^{N_H-1} w_{k,j}(t) \cdot n_j(t))$ (6)

$$i = 0,1,2, \quad j = 0,1,\cdots,N_H - 1, \quad k = 0,1,2.$$

Hence, based on the steepest descent algorithm, the output layer weightings $w_{k,j}(t)$ are updated as follows.

$$w_{k,j}(t+1) = w_{k,j}(t) + \Delta w_{k,j}(t+1) \quad (7)$$

$$\Delta w_{k,j}(t+1) = -\eta \frac{\partial E}{\partial w_{k,j}(t)} + \alpha \Delta w_{k,j}(t) \quad (8)$$

where $\eta$ defines the learning rate and $\alpha$ is the momentum which improves the convergence speed and helps the network from being trapped in a local minimum. Similarly, the hidden layer weightings are updated. It can be easily verified that

$$\Delta w_{k,j}(t+1) = \eta \delta_k(t) n_j(t) + \alpha \Delta w_{k,j}(t) \quad (9)$$

$$\delta_k(t) = e(t+1) \frac{\partial y(t+1)}{\partial u(t)} n_k(t)(1 - n_k(t)) \frac{\partial u(t)}{\partial n_k(t)} \quad (10)$$

$$\Delta w_{j,i}(t+1) = \eta \delta_j(t) n_i(t) + \alpha \Delta w_{j,i}(t) \quad (11)$$

$$\delta_j(t) = \sum_{k=0}^{2} \delta_k w_{k,j}(t) n_j(t)(1 - n_j(t)). \quad (12)$$

In order to calculate $\delta_k(t)$, the system Jacobian is necessary. One can implement neural network emulator or system identifier to obtain the Jacobian or one can use the sign of the Jacobian and adjust the step size $\eta$ than to implement the emulator or identifier. In the case of our SNS LLRF system, the sign of the Jacobian of the system, the cascade of the klystron, cavity, and attenuator can be easily figured out and hence, the second approach is chosen.

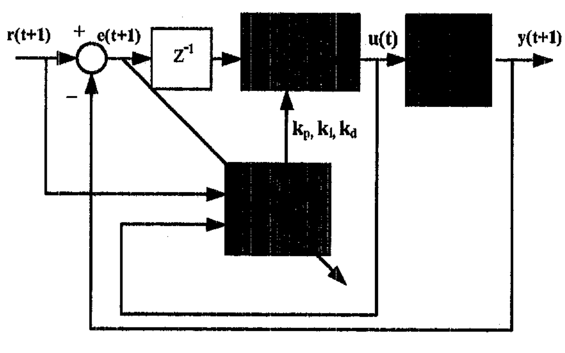


**Figure 1**: Neural Network tuning PID Controller.

When a PI controller gains are necessary to be tuned, then the same neural network tuner is applied. The discrete time PI controller is given by

$$u(t) = u(t-1) + k_p(e(t) - e(t-1)) + k_i e(t) \quad (13)$$

$$e(t) = r(t) - y(t)$$

and the neural network which tunes the PI gains, $k_p$ and $k_i$, is expressed as

**Input Layer Nodes:** $n_i(t) = e(t-i)$ (14)

**Hidden Layer Nodes:** $n_j(t) = f(\sum_{i=0}^{1} w_{j,i}(t) \cdot n_i(t))$ (15)

**Output Layer Nodes:** $n_k(t) = f(\sum_{j=0}^{N_H-1} w_{k,j}(t) \cdot n_j(t))$ (16)

$$i = 0,1, \quad j = 0,1,\cdots,N_H - 1, \quad k = 0,1.$$

## 3 MODELING AND SIMULATION

The neural network tuning PID/PI controller based low level RF control system is modelled with MATLAB/SIMULINLK. Figure 2 shows the model. Since the low level RF control system is developed in In-phase (I) and Quadrature (Q) coordinates, for each channel, neural network tuning PID controller is developed respectively. The advantageous fact is that the structure I and Q channels are symmetric, the same structures of the tuning neural network is applied and hence, the same step size $\eta$, and the same momentum $\alpha$ are used.

Since the neural networks are on-line, the neural network tuning PID/PI controller has the property of the adaptive control which has the potential of self-calibration against the system parameter changes due to aging, ambient temperature change, etc. Furthermore, it has the potential of the learning feedforward control against the beam loading transient since the backpropagation neural network is the feedforward network.

Two types of learning approaches are investigated for SNS pulsed RF system. The first learning approach is that the learning is performed at every sample of data. The second learning approach is that the learning is performed once at each RF pulse. In case of the first learning approach, the tuning PID/PI controller improves the performance against the beam loading as the RF pulse number increases. However, it needs computational burden because the update should be completed within one sampling time. One remedy is to decimate the data sampling. In case of the second learning approach, it improves the steady state performance of each RF pulse as the RF pulse number increases. Its computational burden is minimal because the average of certain number of samples in each RF pulse or a single sample in each RF pulse is used for the learning. However, its performance against the beam loading is worse than the first learning approach even though the performance improvement is expected.