S12-63
34072
p.13

1995113221

N95-19637

NEURAL NET CONTROLLER FOR INLET PRESSURE CONTROL OF ROCKET ENGINE TESTING

Luis C. Trevino
NASA-MSFC, Propulsion Laboratory
Huntsville, Alabama

ABSTRACT

Many dynamic systems operate in select operating regions, each exhibiting characteristic modes of behavior. It is traditional to employ standard adjustable gain PID loops in such systems where no apriori model information is available. However, for controlling inlet pressure for rocket engine testing, problems in fine tuning, disturbance accommodation, and control gains for new profile operating regions (for R&D) are typically encountered [2]. Because of the capability of capturing i/o peculiarities, using NETS, a back propagation trained neural network controller is specified. For select operating regions, the neural network controller is simulated to be as robust as the PID controller. For a comparative analysis, the Higher Order Moment Neural Array (HOMNA) method [1] is used to specify a second neural controller by extracting critical exemplars from the i/o data set. Furthermore, using the critical exemplars from the HOMNA method, a third neural controller is developed using NETS back propagation algorithm. All controllers are benchmarked against each other.

I. INTRODUCTION

An actual propellant run tank pressurization system is shown in Figure 1.1 for liquid oxygen (LOX). The plant is the 23000 gallon LOX run tank. The primary controlling element is an electro-hydraulic (servo) valve labeled as EHV-1024. The minor loop is represented by a valve position feedback transducer (LVDT). The major or outer loop is represented by a pressure transducer (0-200 psig). The current controller is a standard PID servo controller. The reference pressure setpoint is provided by a G.E. Programmable Logic Controller. The linearized state equations for the system are shown below:

$$\dot{x}_1=x_2-(0.8kg+c)x_1 \tag{1.1}$$

$$\dot{x}_2=5kg\ au-(0.8kg\ c+d)x_1+x_3 \tag{1.2}$$

$$\dot{x}_3=5abkg\ u-(0.8kg\ d+f)x_1+x_4 \tag{1.3}$$

$$\dot{x}_4=-0.8kg\ fx_1 \tag{1.4}$$

where kg=1, servo valve minimum gain. Based on previous SSME test firings, the average operating values for each state

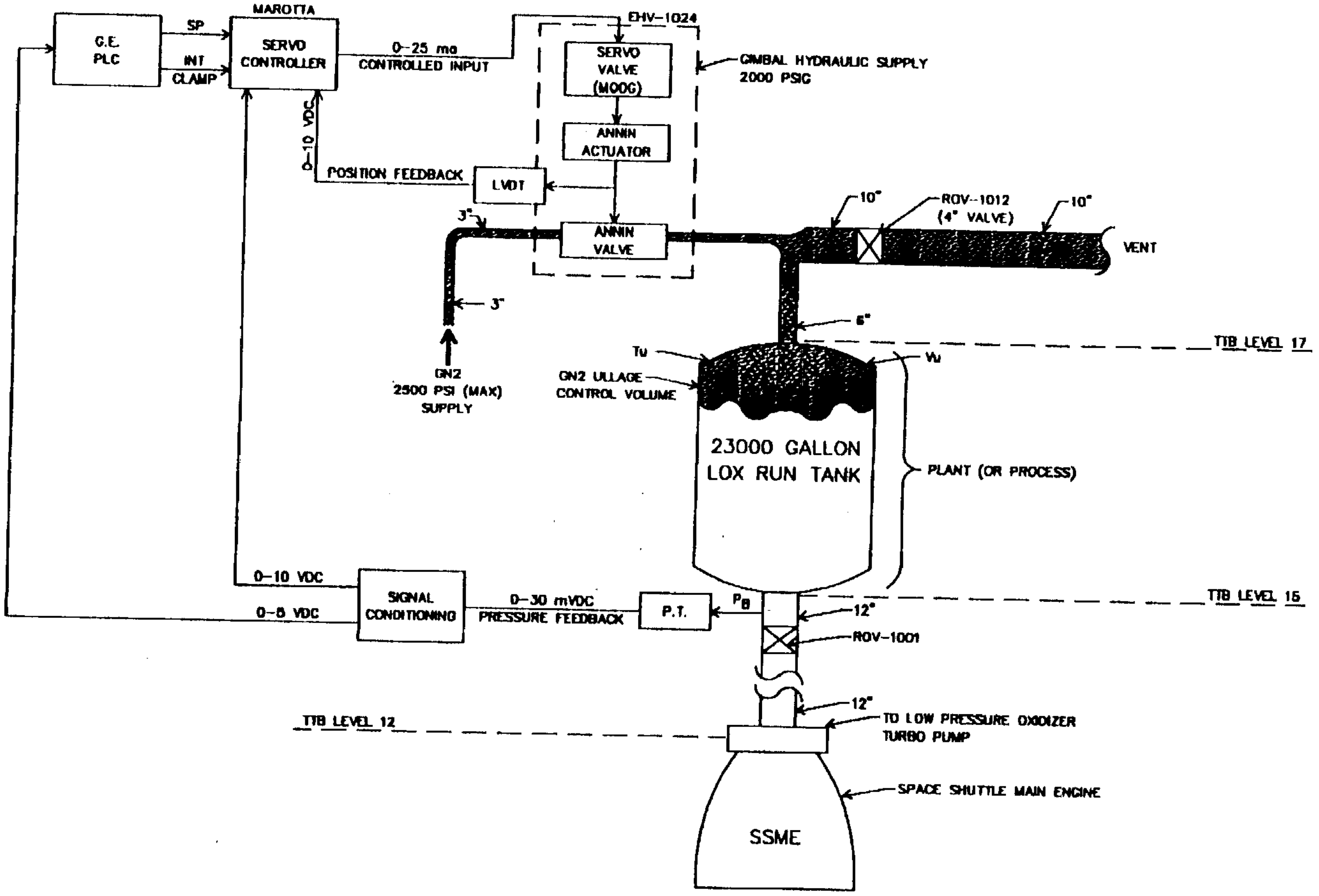

Figure 1.1 Technology Test Bed Propellant Lox Run Tank Pressurization System

variable are determined to be

x_1 = P_B:0-76 psig x_3 = V_u:250-350 ft^3

x_2 = T_u:150-300 R^o x_4 = L:0-1 inch

where P_B = bottom tank pressure

T_u = ullage temperature

V_u = ullage volume

L = valve stem stroke length

Using those ranges, the following average coefficients are algebraically determined:

a = 120.05 d = 5995.44

b = 89.19 f = 14.70

c = 214.30

II. Methodology

1. Using a developed PID-system routine from [2], an i/o histogram is established in the required format per [1] for a select cardinality of 300. Figure 2.1 portrays the scheme. A ramp control setpoint signal (from 0-120 psig) served as the reference trajectory. The input portion of the histogram is selected to be a five dimensional (300x5) matrix, four successive delayed samples and the current sample. The output portion is a one dimensional (300x1) vector. Therefore, the i/o histogram is simply represented by a 300x6 matrix.

2. Using the captured i/o data set and NETS back propagation algorithm, a neural network is next established with a 5-10-10-1 architecture. The trained network is next simulated as the controller for the system. Figure 2.2 illustrates the simulation scheme.

3. Using a developed HOMNA (KERNELS) algorithm [1], a reduced training i/o set is specified. The input portion of the set, "S", will provide the mapping of real time system inputs to the neural net controller (NNC). The output segment of the set is represented by the last column vector of the i/o set.

4. After configuring the reduced i/o set into the needed formats, using MATLAB, the gain equation (2.1) is executed.

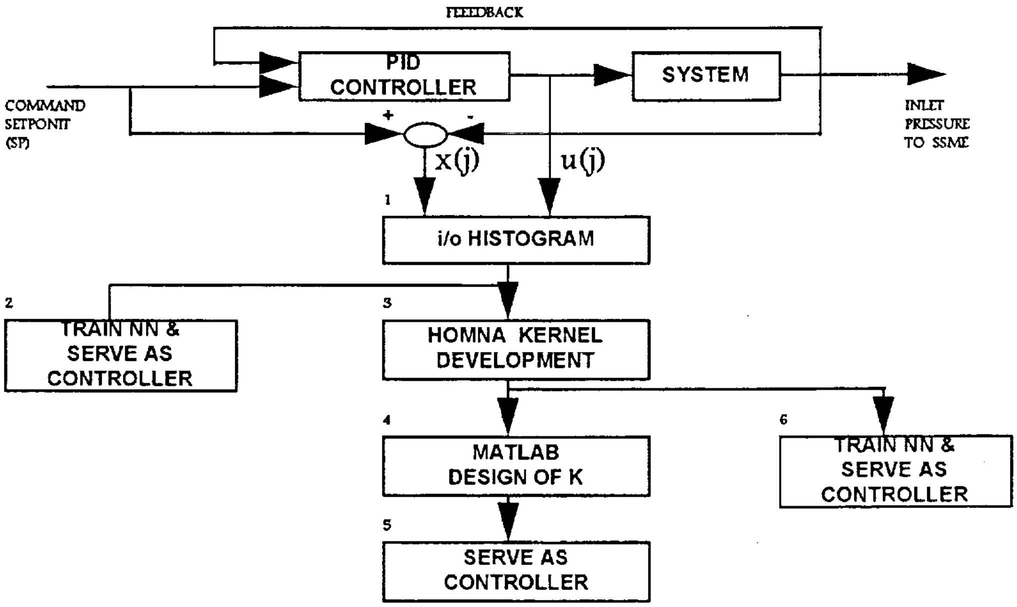

Figure 2.1 Scheme For Building i/o HIstogram and Training Set

$$K = YG^{-1} = Y\Psi(SS^*) \tag{2.1}$$

where K = neural gains (row vector) for single neural layer
Y = NNC controller output signature row vector
S = established matrix set of step 3
Ψ = any (decoupled) operation: exponential, etc.

For this project, Ψ was identical with that used in the literature of [1], namely the exponential function. "K" serves as a mapping function of the input, by way of "S", to the NNC output, u(j). Here, u(j) serves as control input to the system and is determined by equation (2.2) [1].

$$u(j) = K\Psi(Sx(j)) \tag{2.2}$$

where x(j) is the vector input. In accordance with the dimensions of the i/o histogram, a five dimensional input is used and is accomplished using successive delays. Namely, a typical input for any given sample is represented by

$$[x(j)\ x(j-1)\ x(j-2)\ x(j-3)\ x(j-4)]$$

The overall HOMNA scheme is embedded in the neural controller block of Figure 2.2 as a 5-5-1 architecture (single hidden layer).

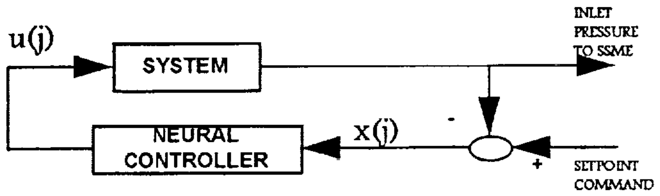

Figure 2.2. Simulation Scheme for NNC and System

5. For select cases to be presented, integral control for the HOMNA system was presented according to the following scheme of [1].

$$\Delta = \frac{1}{N}\sum[y(j)-\bar{Y}(j)] \tag{2.3}$$

where N = window (sampling) size
$y(j)$ = current system output
$\bar{y}(j)$ = desired output, or command setpoint, sp

6. Using the training i/o set of part 3, a separate neural network controller is established, again using NETS. The simulation scheme is similar to that of part 2.

7. PID system response plots are generated for a further comparitive analysis.

III. **RESULTS**

Table I. Case Summary

Case	System	Setpoint Command	Noise	Integral Control
1	Neural	Ramp	no	n/a
2	HOMNA	Ramp	no	None
3	HOMNA	Ramp	no	None
4	PID	Ramp	no	Present
5	Neural	Ramp	yes	n/a
6	HOMNA	Ramp	yes	Present
7	PID	Ramp	yes	Present
8	Neural	Profile	no	n/a
9	HOMNA	Profile	no	None
10	HOMNA	Profile	no	Present
11	PID	Profile	yes	Present
12	Neural	Profile	yes	n/a
13	HOMNA	Profile	yes	None
14	HOMNA	Profile	yes	Present
15	Neural	Ramp	no	n/a

* Case for procedural step 6

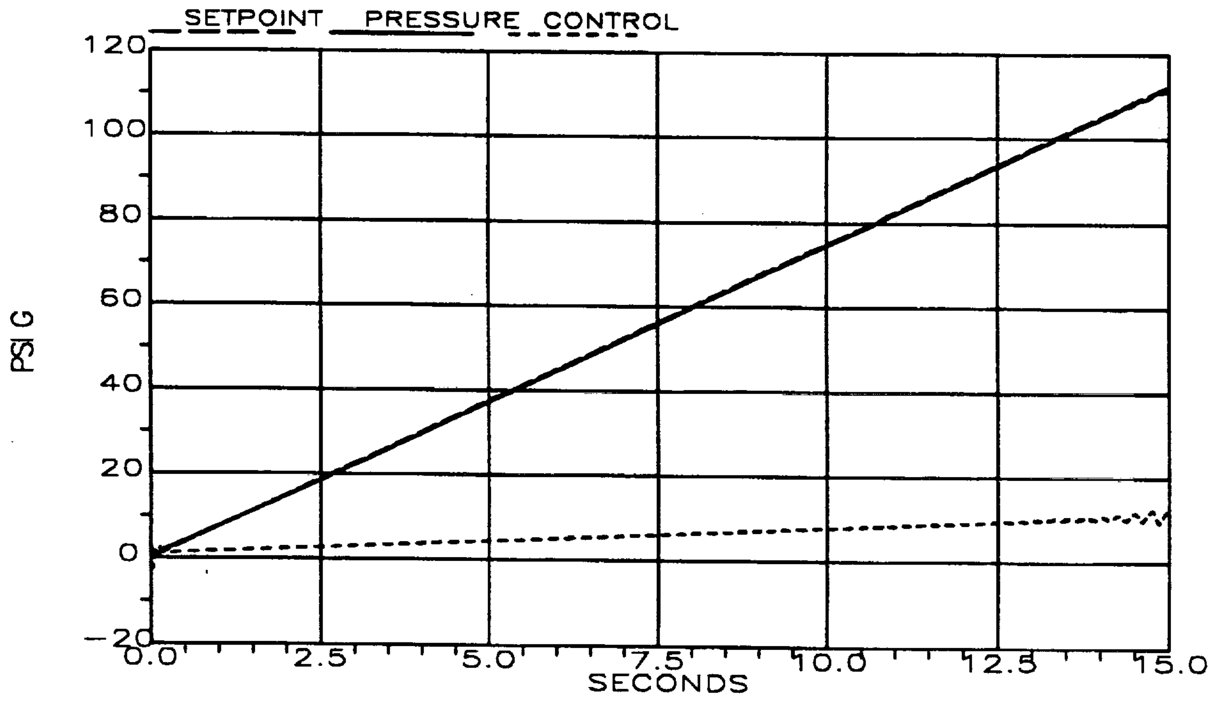

Figure 3.1 Case 1, 3, and 4 Simulation Results

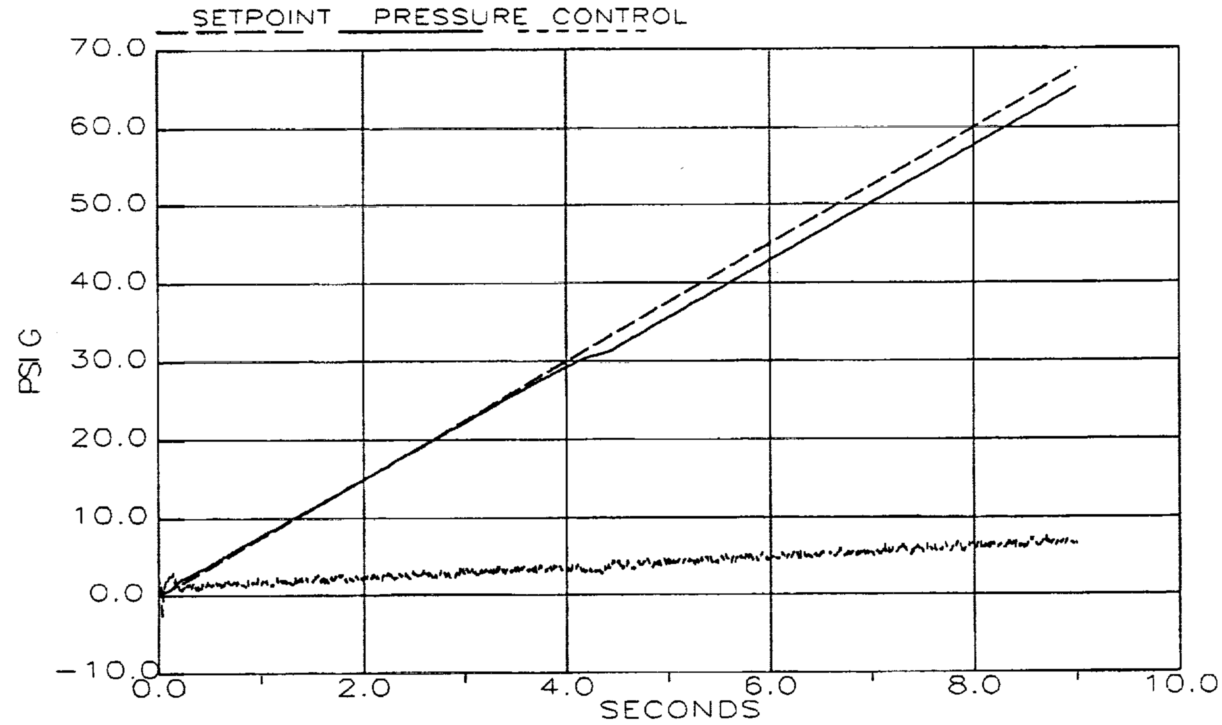

Figure 3.2 Case 2 Simulation Results

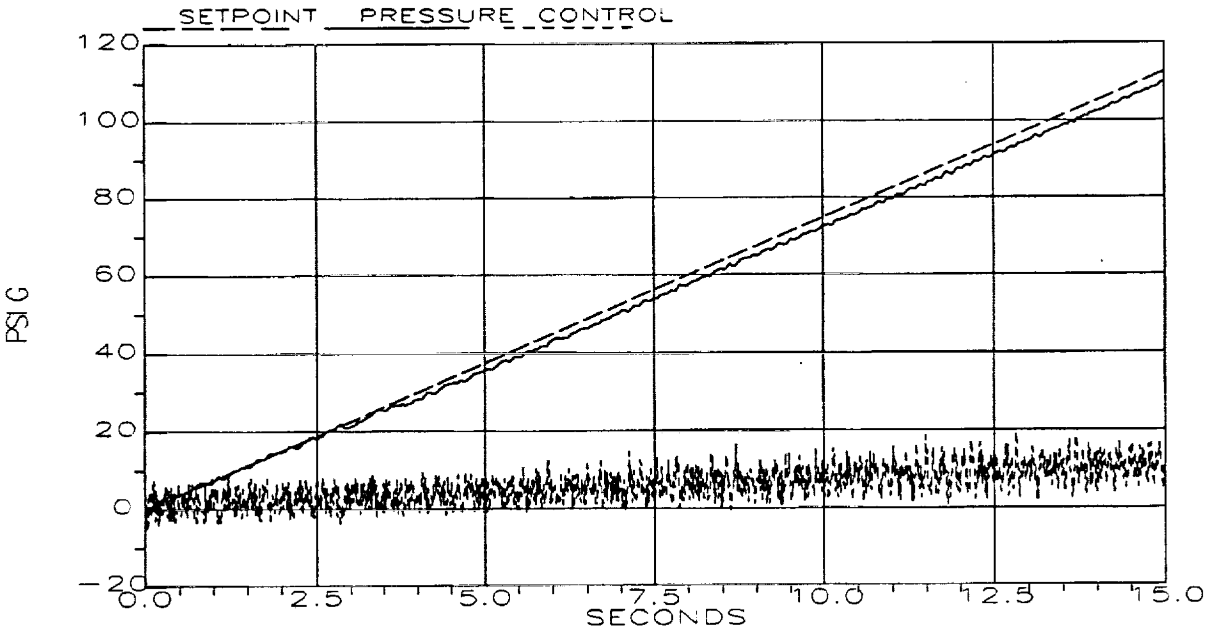

Figure 3.3 Case 5 Simulation Results

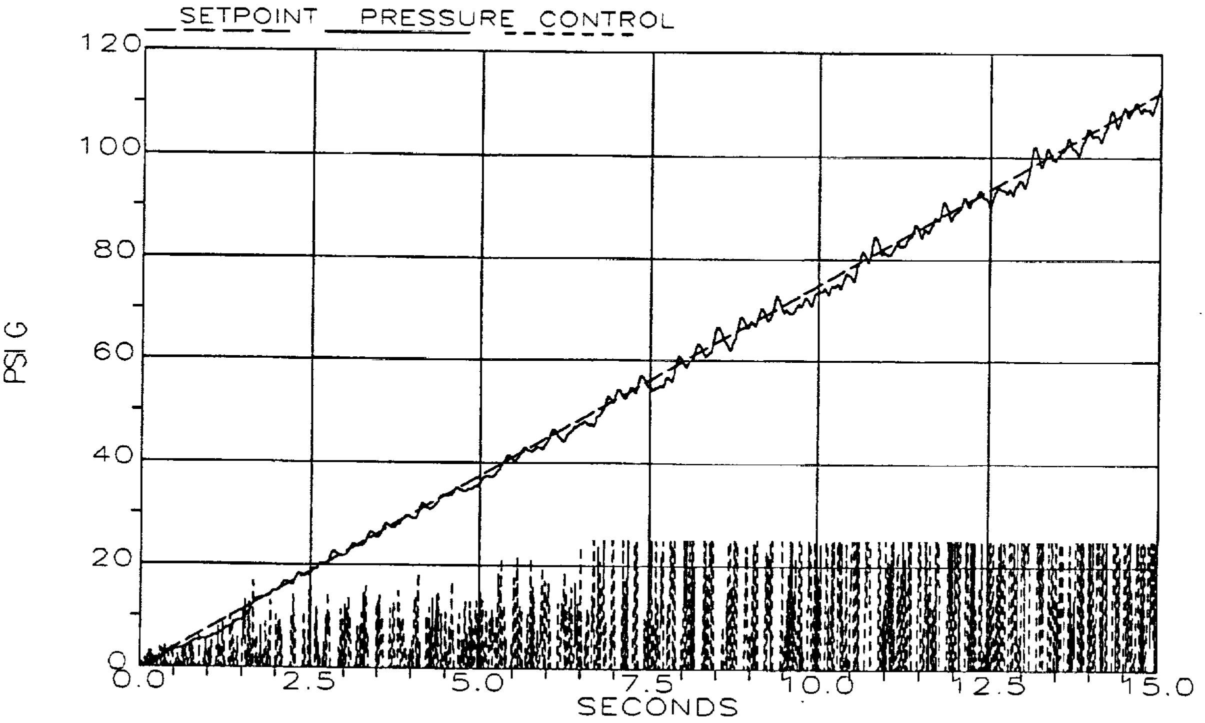

Figure 3.4 Case 6 Simulation Results

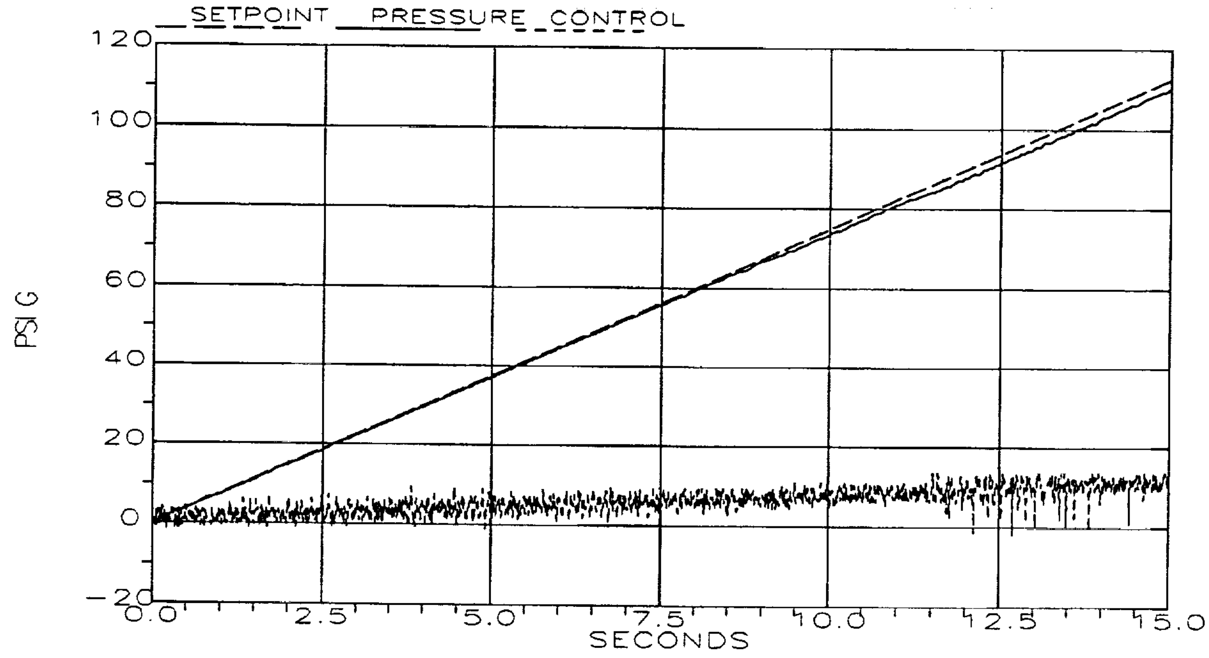

Figure 3.5 Case 7 Simulation Results

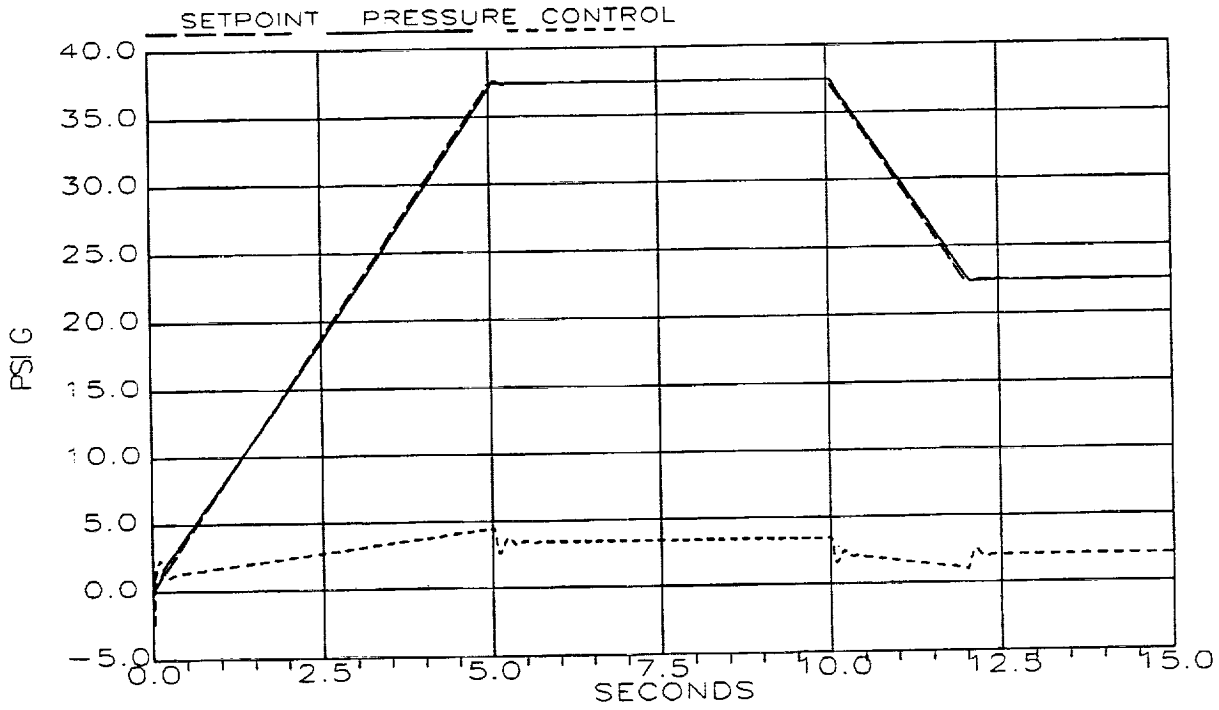

Figure 3.6 Case 8 and 10 Simulation Results

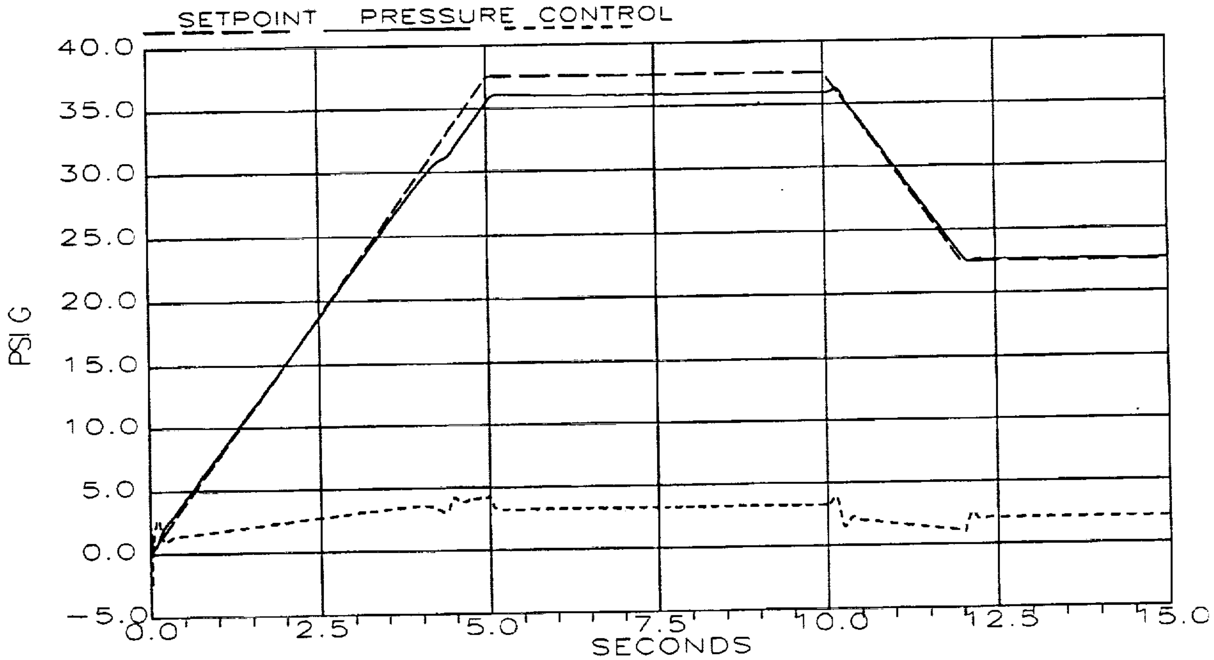

Figure 3.7 Case 9 Simulation Results

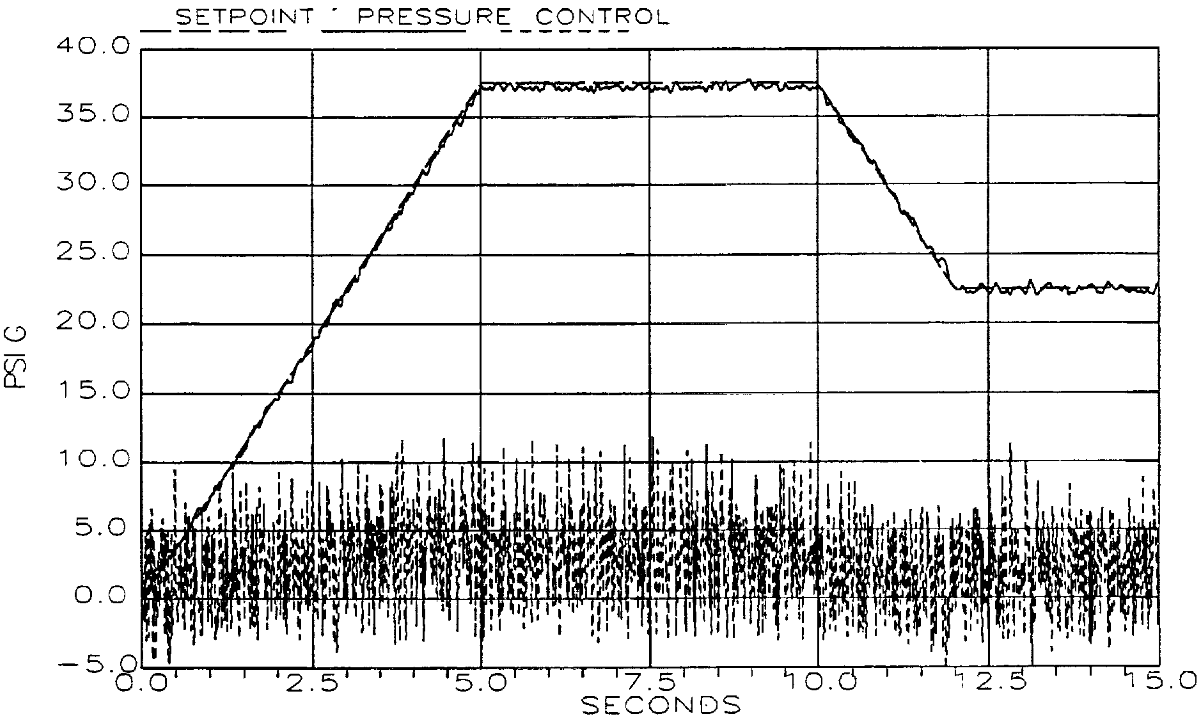

Figure 3.8 Case 11 Simulation Results

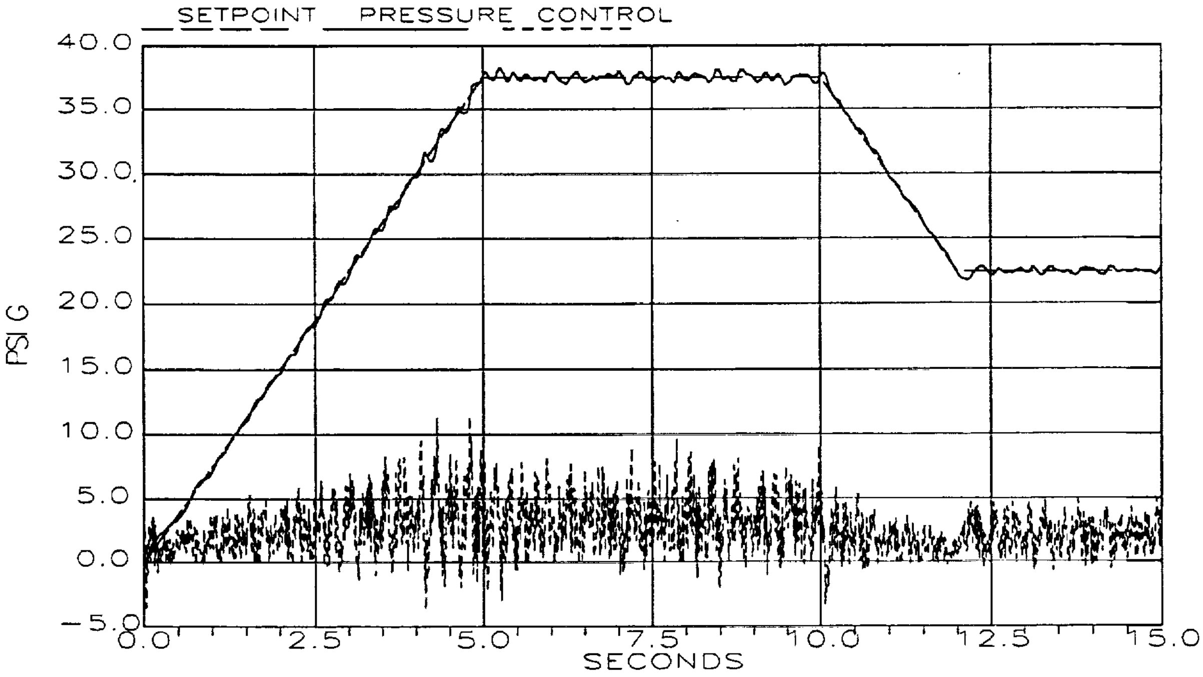

Figure 3.9 Case 12 and 14 Simulation Results

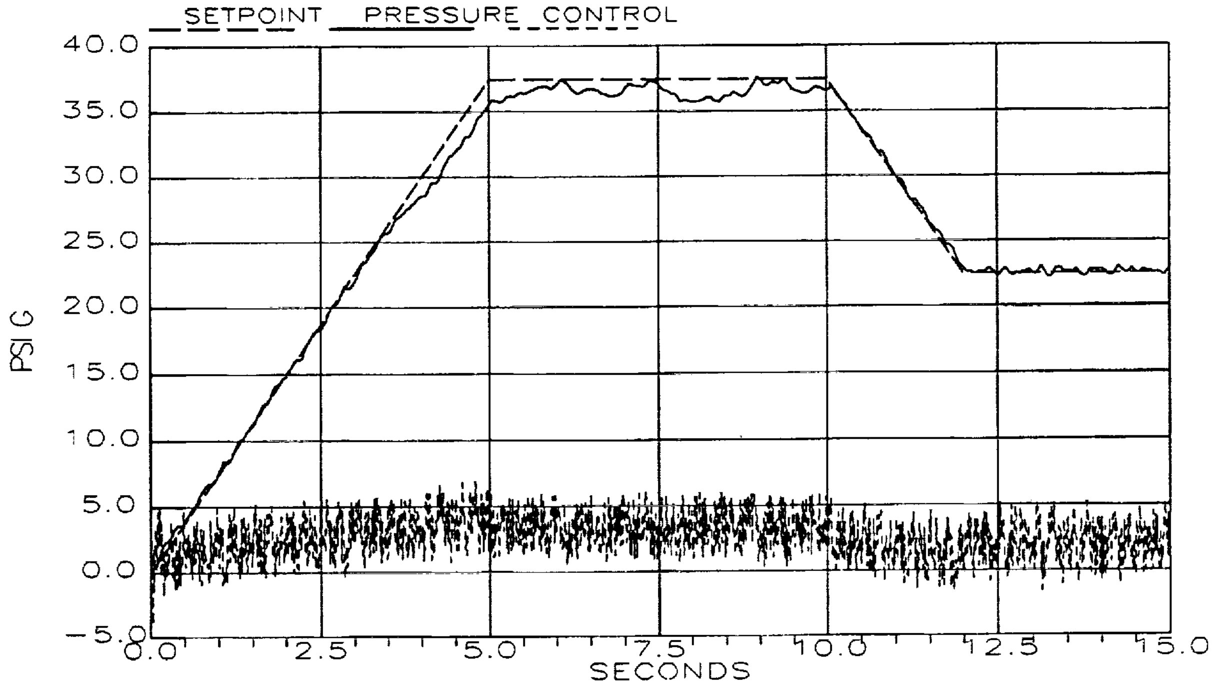

Figure 3.10 Case 13 Simulation Results

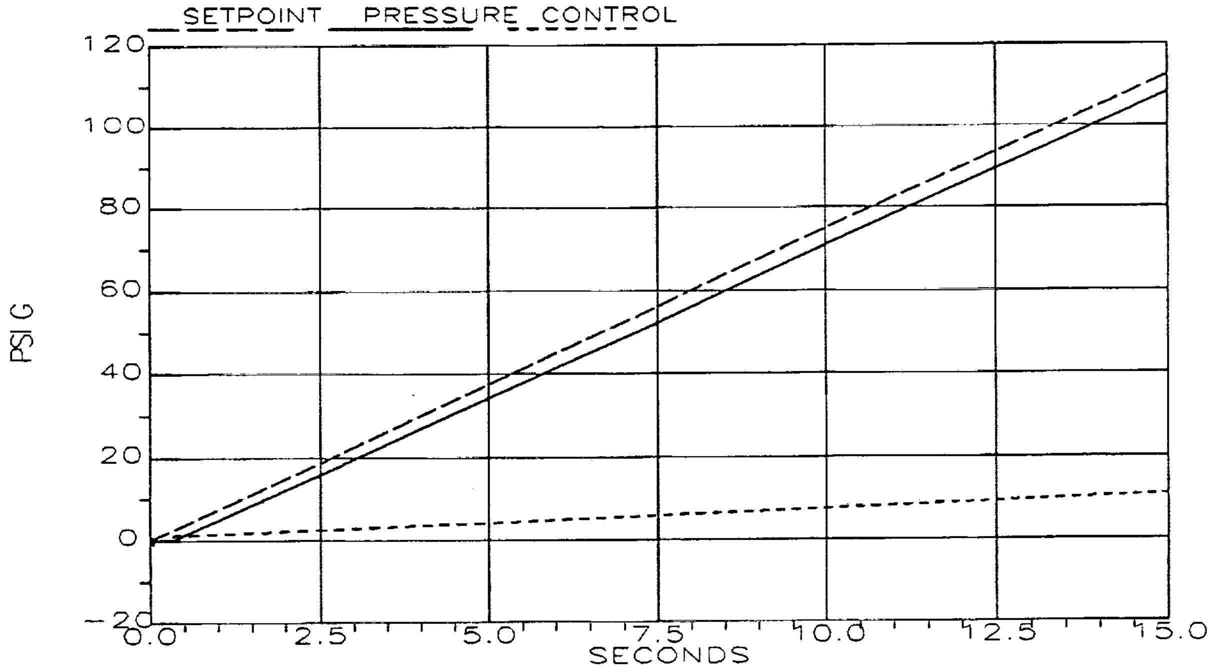

Figure 3.11 Case 15 Simulation Results

IV. DISCUSSION AND CONCLUSIONS

From Table 3.1 and the presented simulation results, the back-propagation trained and the HOMNA neural systems are proven to track varying command setpoints within the bounds of the training i/o histogram. Without incorporation of the integration scheme of [1], the HOMNA system still proved its tracking ability, though with varying levels of offsets. The proportional-integral-derivative (PID) system results exhibited no offsets due to the inherent integral scheme of the PID controller. From the simulation results, it is concluded that the integration scheme of [1] was simple to employ with equally satisfying results. Both back-prop trained, HOMNA, and PID systems proved their ability to accommodate for the varying levels of random noise injection.

In the use of NETS for the back propagation trained neural network, the ability to adjust the learning rate, momentum term, and the scaling factor (globally or locally) allowed for various configurations for starting conditions in the training. For the this project, the default global momentum was used, 0.09. A global learning rate of 0.5 through 1 was used for all cases. A scaling factor of 0.1 was used for all cases.

For the HOMNA trained system, larger i/o histogram sets were attempted with no significant difference in performance for select cases. With more effort or other techniques, it is believed that the difference could be corrected. In this project it was discovered that stripping the first few exemplar vectors from the i/o histogram (or the established training set) made a significant difference in the performance. For some cases, without stripping the first few inherent exemplar state vectors resulted in erroneous results ranging from wide dispersion (between setpoint and system state) to complete instability. The justification for stripping the first few exemplars stems from the scheme of [1]. That is, for the first few exemplars there is always inherent membership in the training set kernel. For select cases, the effects of stripping the exemplars before or after the Kernels algorithm software routine had no indicative difference.

For the neural controller of step 6 (i/o training set generated by the Kernels algorithm), Figure 3.11 illustrates that the controller can still track the command setpoint; however, the amount of offset, unseen in other backprop cases where the training set was the full i/o histogram (300 samples), is obviously due to the reduced size of the training set (40 samples). This was expected since the purpose of the Kernels algorithm is to select critical exemplars from a large data set. It is these critical exemplars that best represents the set (or population) as a whole. The choice of a back-prop trained or a HOMNA based neural controller to serve as a standalone or parallel backup to an existing PID controller is certainly realizable.

REFERENCES

1. Porter, W. A. and Liu, Wie, "Neural Controllers For Systems With Unknown Dynamics," The Laboratory for Advanced Computer Studies, The University of Alabama in Huntsville, March 1994.

2. Trevino, L. C., **MODELING, SIMULATION, AND APPLIED FUZZY LOGIC FOR INLET PRESSURE CONTROL FOR A SPACE SHUTTLE MAIN ENGINE AT TECHNOLOGY TEST BED**, Masters Thesis, The University of Alabama in Huntsville, June, 1993.