# Neural Network Identifier with Iterative Learning for Turbofan Engine Rotor Speed Control System

QIAN Kun PANG Xiangping LI Bangyuan

Department of Aerial Instrument and Electric Engineering

The First Aeronautical Institute of Air Force

Xinyang, China

qiankun_0306@163.com

***Abstract*—This paper proposed a new neural network algorithm with fuzzy iterative learning controller and applied to a certain turbofan engine rotor speed control system. A dynamic neural network was used to identify the plant on-line. The control signal was then calculated iteratively according to the responses of a reference model and the output of identified plant. A fuzzy logic block with four very simple rules was added to the loop to improve the overall loop properties. Experimental results demonstrate the proposed control strategy provides better disturbance rejection and transient properties than those achieved by conventional mechanical-hydraulic controller(MHC) and analogue engine electronic controller(AEEC). At the same time, it can improve transitional quality in control system, and meet the demands of high performance and high control accuracy in turbofan engine.**

***Keywords*—Aerial Engine; Neural Network Identifier(NNI); Iterative Learning Controller(ILC); Trans-dimensional Learning(TDL); Fuzzy Logic Compensation(FLC)**

## I. INTRODUCTION

In recent years, with the rapid development of modern control theory and the application of super large-scale integration and microprocessor in aerial engine electronic controller, full authority digital electronic controller (FADEC) has been an important developing direction in high performance aeroengine control system. Comparing with conventional mechanical-hydraulic controller and analogue engine electronic controller(AEEC), FADEC not only has the simple configuration, but also apply complicated and anvanced control algorithm to aeroengine control system. It can realize multi-variables and multi-loops decoupling control in system. So it meets the demands of high performance guideline and high control accuracy in turbofan engine[1],[2].

Neural networks are probably the most discussed intelligent control paradigm used today in model identification problems and in controller design[3],[4]. Integrated with neural network application in control field, this paper proposed a new neural network algorithm with fuzzy logic compensation and applied to an aeroengine rotate speed control system. Experimental results demonstrate the proposed control strategy is effective.

## II. CONTROL MODEL DESIGN

Figure.1 depicts the block diagram of the proposed closed-loop control scheme. The reference input signal is $r$. The control goal is that the plant output signal $y$ follows as closely as possible the output signal $y_m$ of the reference model. The reference model of the plant is an ideal model that has the desired characteristics related to the rise time, overshoot, steady-state error, etc.

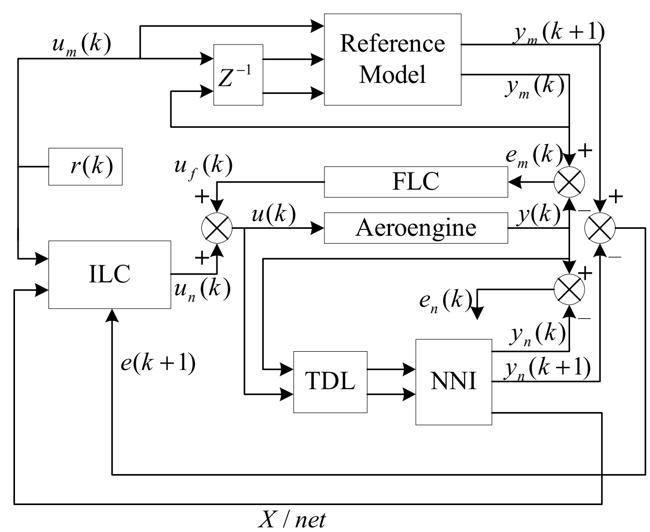


Figure.1 Neural network Control block with fuzzy logic compensation

Ignoring aeroengine combustion delay, the dynamic model of rotor speed control in a sample form as[5],[6]:

$$\frac{X_{n_L}(s)}{X_{m_f}(s)} = \frac{K_L(\tau_L s + 1)}{T_1^2 s^2 + T_2 s + 1} \tag{1}$$

Where $X_{n_L}$ and $X_{m_f}$ are low compressor rotor speed and fuel respectively. After discretization, the model can be rewritten as:

$$y_m(k+1) = a_1 y_m(k) + a_0 y_m(k-1) +$$

$$b_1 u_m(k) + b_0 u_m(k-1) \quad k=0,1,2\ldots \tag{2}$$

where $u_m$ and $y_m$ are scalars.

Trans-dimensional learning(TDL) is a Windows 3.1 artificial neural network. TDL allows users to perform pattern recognition by utilizing software that allows for fast, automatic construction of Neural Networks, mostly alleviating the need for parameter tuning. By supporting multi-shot learning over standard one-shot learning, multiple data sets (characterized by varying input and output dimensions) can be learned incrementally, resulting in a single coherent network. This can also lead to significant improvements in predictive accuracy.

A neural network identifier(NNI) for the on-line identification determines on-line an approximate current non-linear model of the plant. This model is necessary for good loop control. An iterative learning controller (ILC) produces a control signal $u_n$, which is combined with the output signal of the fuzzy logic block(FLB) to produce the actual plant input signal $u = u_n + u_f$.

## III. NEURAL NETWORK IDENTIFIER

The dynamic NN for plant model identification has $n$ input nodes, $m$ hidden-layer nodes and $l$ output node. The weight matrix for the input-to-hidden layer is $\boldsymbol{W}_{(n+1)\times m}$, the input pattern vector $\boldsymbol{X}_{(n+1)\times l}$, the hidden-to-output weight matrix is $\boldsymbol{V}_{(m+1)\times l}$, the output signal vector of the hidden-layer is $\boldsymbol{net}_{(m+1)\times m}$, Here $n=m+1$ and $m=2p$.

The input layer and the hidden layer have each a 1 node constant to cater for biases of the neurons in the subsequent layer. These two nodes do not have feedback signals. The output $y_n$ of NNI is the predicted system output. The values of variables are considered at moments $t_k=k*\tau$ ($\tau$ is the sampling interval). $k=1,2,\ldots$. To simplify notation we will use $u(k)$ to denote the value $u(t_k)$. Thus, at $t=t_k$ we have the system I/O values $u(k)$ and $y(k)$. We define vectors

$$S(k) = \begin{bmatrix} s(k,1) \\ s(k,2) \\ \cdots\cdots \\ s(k,p) \\ s(k,p+1) \\ \cdots\cdots \\ s(k,2p) \end{bmatrix} = \begin{bmatrix} u(k-1) \\ u(k-2) \\ \cdots\cdots \\ u(k-p) \\ y(k-1) \\ \cdots\cdots \\ y(k-p) \end{bmatrix} \tag{3}$$

$$X(k) = \begin{bmatrix} 1 \\ x(k,1) \\ \cdots\cdots \\ x(k,n) \end{bmatrix}, \; net(k) = \begin{bmatrix} 1 \\ h(k,1) \\ \cdots\cdots \\ h(k,m) \end{bmatrix} \tag{4}$$

Let $x(k, 1)=u(k)$, and by using three positive real parameters $\alpha, \beta, \gamma$, we define a recursive relation for the calculation of $x$ as

$$x(k, i+1)= \alpha*s(k, i)+ \beta*x(k-1, i+1)+ \gamma*h(k-1, i) \tag{5}$$

where $x(0, i+1)=0$, $i=1,\ldots,m$, $k=1,2,\ldots$, also

$$Z(k) = W^T * X(k) \tag{6}$$

$$net(k) = f(Z(k)) \tag{7}$$

Here, $f$ is a nonlinear sigmoid function vector. The NN output $y_n(k)$ at $k$-th time interval is linear in $net$ as follows.

$$y_n(k+1) = V^T * net(k) \tag{8}$$

In (5), $\alpha$ is the forward gain, $\beta$ is the selfish feedback gain on the input layer and $\gamma$ is the feedback gain from the hidden layer to the input layer. We define the error function $J = e_n^2/2 = (y - y_n)^2/2$, so that the necessary gradients and updates of the network parameters are given by

$$\frac{\partial J}{\partial V} = \frac{\partial J}{\partial y_n}\frac{\partial y_n}{\partial V} = -e_n * net \tag{9}$$

$$\frac{\partial J}{\partial W} = \frac{\partial J}{\partial y_n}\frac{\partial y_n}{\partial W} = -e_n * \frac{\partial y_n}{\partial net}\frac{\partial net}{\partial W} \tag{10}$$

$$V_j(k) = V_j(k-1) + \Delta V_j(k) \tag{11}$$

$$\Delta V_j(k) = -\eta \frac{\partial J}{\partial V_j(k)}$$

$$= \eta * e_n * net_j(k)[1+ \mu * net^T(k) * net(k)]^{-1} \tag{12}$$

$$W_{ij}(k) = W_{ij}(k) + \Delta W_{ij}(k) \tag{13}$$

$$\Delta W_{ij}(k) = -\eta \frac{\partial J}{\partial W_{ij}(k)}$$

$$= \eta * e_n * V_j(k) * Z_j(k) * [1-Z_j(k)] * X_j(k) \tag{14}$$

Here, $\eta$ is the learning rate; $\mu$ is a small positive coefficient; subscripts; $i, j$ indicate the $i$-th node in the input layer and the $j$-th node in the hidden layer, respectively; $k$ means the $k$-th sample control period.

## IV. ITERATIVE LEARNING CONTROLLER

Our objective is to control the plant so as that its output $y$ follows the response of the reference model $y_m$. At time $t_k$ we can determine $y_m(k+1)$, which is the next value of the model output. Since the current neutral network model of the plant is identified by NNI, we can directly calculate $u(k+1)$ by the iterative algorithm in order to obtain the NNI output $y_n$ close to $y_m(k+1)$. To do this we will look for $u(k+1)$ that will make the expected output of NNI equal to $y_m(k+1)$. The value of $u(k+1)$ will be obtained iteratively according to [5], [6]. During the calculation of the iterative values we assume that the system does not change its characteristics. Here, the NN model serves as a known invariant nonlinear model of the plant. We will use the PD control structure to calculate the next value of $u$ using the current iterative value of $u$, as well as the current and previous iterative values of $e=y_m-y_n$.

Some explanations of the notation are necessary. Let the control time interval be $T_k=[t_k, t_k+\tau]$. Here, $t_{k+1}=t_k+\tau$ and it represents the next sampling moment. During $T_k$ the iteration calculation is run. Let $kk$ correspond to the current iteration step. The current moment at the $kk$-th iterative step is $t=t_k+\Delta_{kk-1}$, where $\Delta_{kk-1}$ is the time necessary to perform the total of $kk$ iterations. Also, notation $u(kk, t)$ represents the $kk$-th iteration value of $u$. The error during the $kk$-th control iterative input is denoted by $e(kk, t)$. The iterative calculation of the control signal $u$ is given by

$$u(kk+1,t)=u(kk,t)+k_p*e(kk,t)+k_d*[e(kk,t)-e(kk-1,t)], \quad t=t_k \tag{15}$$

$$e(kk,t)=y_m(t)-y_n(kk,t), \quad t=t_k \tag{16}$$

Here $k_p$ and $k_d$ are the PD parameters. The whole iteration session has to be finished during $T_k$.

The whole iteration control step is calculated as follows:

a) kk=1； e(0,k+1)=0；
b) $\hat{u}(kk,k+1)=u(kk,k)$ , where $\hat{u}$ is the estimated value of u;
c) calculate $y_n(kk,k+1)$ by（5）~（8）
d) $e(kk,k+1)=y_m(k+1)-y_n(kk,k+1)$
e) if $|e(kk,k+1)|\le\varepsilon_c$, then GO TO g), ELSE：

$$\hat{u}(kk,k+1)=\hat{u}(kk,k+1)+k_p*e(kk,k+1)+k_d*[e(kk,k+1)-e(kk-1,k+1)]$$

f) kk=kk+1； if $\Delta_{kk-1}<\tau$ RETURN to step b).
g) $u(k+1)=\hat{u}(kk,k+1)$ , END iterations.

## V. Fuzzy Logic Block

In the proposed control structure the PD type fuzzy logic block(FLB) is used. The FLB acts as a damper in the loop. The output of the FLB is governed by

$$u_f = k_{fu} * F_{fuzzy}(e_m, \dot{e}_m) \tag{17}$$

Here $u_f$ is the FLB output and $k_{fu}$ is the gain; $e_m$, $\dot{e}_m$ are the error signal and its derivative; $F_{fuzzy}$ is a non-linear function whose output signal represents crisp values resulting from internal fuzzy logic processing of the FLB. The crisp input signals $e_m$ and $\dot{e}_m$ are normalized to the range [-1,1]. Simple Gaussian membership functions for the normalized values of $e_m$ and $\dot{e}_m$ are used. Let N and P denote 'negative' and 'positive' respectively; then the membership functions are

$$\mu_N(x)=exp(-10(x+0.3)^2)$$
$$\mu_P(x)=exp(-10(x-0.3)^2) \tag{18}$$

TABLE.I FUZZY LOGIC CONTROL RULES

| $e_m \backslash \dot{e}_m$ | N | P |
|---|---|---|
| N | NB | N |
| P | P | PB |

The four fuzzy rules used in the design of the FLB are represented by the following Tab 1. where NB and PB denote 'negative big' and 'positive big', respectively. As a result of the fuzzy inference mechanism, the output fuzzy value is produced. The membership functions for the normalized output values are:

$$\mu_{NB}(x)=[1+exp(6(x+0.3))]^{-1}$$
$$\mu_{PB}(x)=[1+exp(-6(x-0.3))]^{-1}$$
$$\mu_N(x)=exp(-12.5(x+0.3)^2)$$
$$\mu_P(x)=exp(-12.5(x-0.3)^2) \tag{19}$$

After the output fuzzy value is obtained, it is defuzzyfied, denormalized, and multiplied by the gain $k_{fu}$ to produce $u_f$.

## VI. Experimental Results

Our experiments are based on a model of the aeroengine rotor speed control system used in laboratory. Obviously, Ignoring aeroengine combustion delay, the model can be simplified to a control system of the second order as expression (2). The proposed rotor control system configuration shows as figure.2. the control time interval is $\tau$ =0.02s. The reference discrete model with good performance and realization for rotor speed control system in certain turbofan engine can be taken as

$$y(k+1)=1.8953y(k)-0.8978y(k-1)+0.0107u(k)-0.0102u(k-1) \tag{20}$$

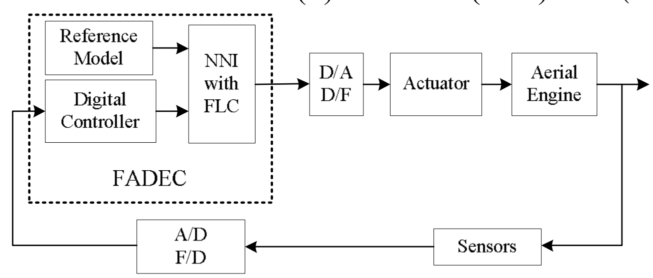


figure.2 The proposed speed control system block for an aeroengine

System dynamic parameters, such as $T_1^2$ 、$T_2$、$K_L$、 $\tau_L$, will change in limited scope with the different flight altitude $H$ and mach number M for aeroengine reference model (1). So these dynamic parameters are uncertain for aeroengine rotor control system. By experiment, in whole aeroengine flight envelope, their possible ranges are $T_1^2\in[0.1, 0.2]$, $T_2\in[0.5, 1.5]$, $K_L\in[0.1, 0.4]$, $\tau_L\in[0.2, 0.5]$. In conventional aeroengine mechanical-hydraulic controller, because hydraulic amplifier valve with speed feedback is plus overlap configuration, it leads to dead zone effect inevitably in aeroengine rotor speed control system. the input saturation limit is [-6, +6]; the dead-zone gap is $\pm0.2$. All numerical values of the simulation model are obtained either by measurements or identification from laboratory experiments. The software environment used for these simulation experiments is Matlab 6.0, with the Simulink package; numerical integration is done by the Runge-Kutta 4 algorithm.

The size of neural network is 8×7×1; the learning rate of neural network is $\eta$=0.2; the control iterative parameters are

$k_p$=0.2 and $k_d$=0.01, the normalization factors for inputs $e_m$ and $\dot{e}_m$ to the FLB are $k_{em}$=1 and $k_{dem}$=0.1 respectively, while the gain $k_{fu}$=1.

The maximal number of iteration steps during the sampling interval is 20. The initial values of vectors $\boldsymbol{S}$, $\boldsymbol{X}$ and ***net*** are all set to zero for the variable components; the initial weights of matrices $\boldsymbol{W}$ and $\boldsymbol{V}$ are taken at random; the three gains for recursive calculations of $\boldsymbol{X}$ are $\alpha$=0.2, $\beta$=0.5, $\gamma$=0.5.

Figure.3 shows the rotor speed loop response in certain turbofan engine flight/train maximum condition when the step type reference input is main fuel signal $m_f$ in flight altitude $H$=0km. From the moment $t$=2.5sec and until $t$=7.5sec, a step type disturbance signal is injected into the loop. Figure.4 and Figure.5 show enlarged sections for a better comparison of the transient properties and disturbance rejection properties of the proposed control algorithm and that of the conventional mechanical-hydraulic controller(HMC). Remarkably, the proposed NNI with FLC can eliminate steady-state error, and the conventional HMC exists steady-state error about 0.24%.

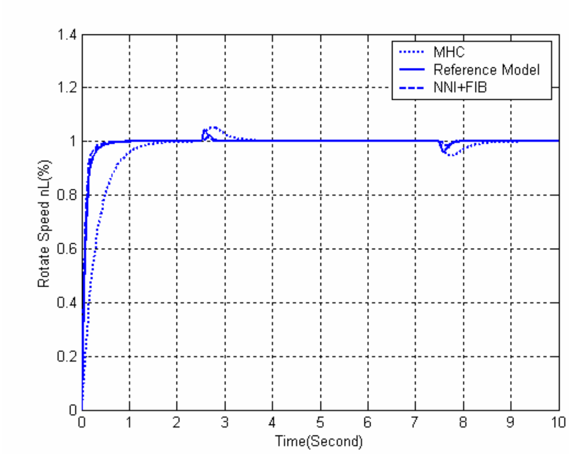


Figure.3 Loop response with step type reference input signal

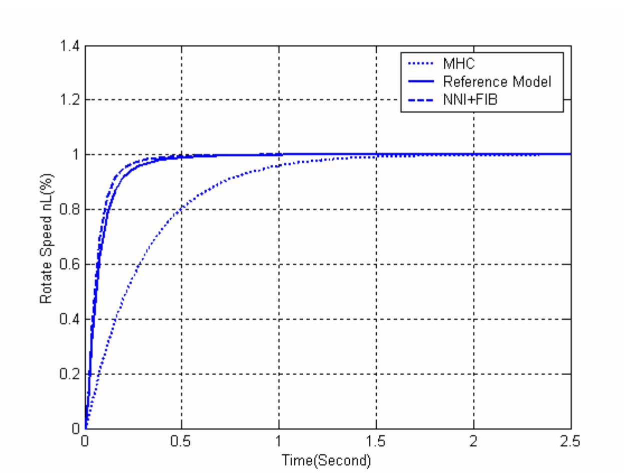


Figure.4 System transient properties with the proposed algorithm and speed feedback algorithm

Figure.5 shows the rotor speed loop response in certain turbofan engine flight/train maximum condition when the step type reference input is main fuel signal $m_f$ in static test. As seen from Figure.6, for mechanical- hydraulic controller, the surge occurred at steady-state point under the influent of components nonlinear characteristic and aeroengine combustion delay. It led to a long surging time and a bad transient properties. For analogue engine electronic controller(AEEC), although it can eliminate steady-state error, it also led to a long surging time and a less overshot. The proposed algorithm brought about good control results, such a short surging time and overshot restraint. So it can improve transitional quality in control system, and meet the demands of high performance and high control accuracy in turbofan engine.

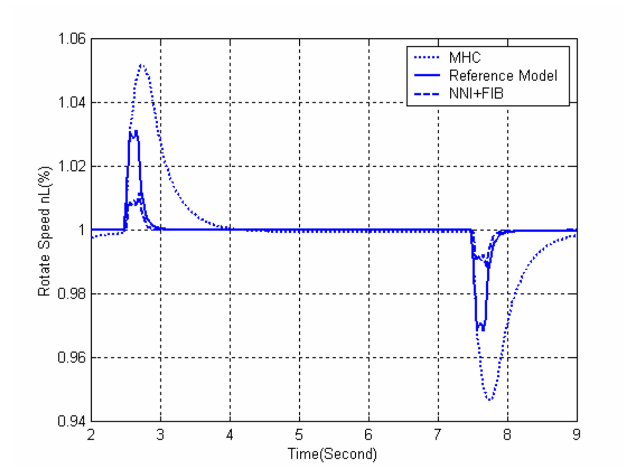


Figure.5 System disturbance rejection properties with the proposed algorithm and speed feedback algorithm

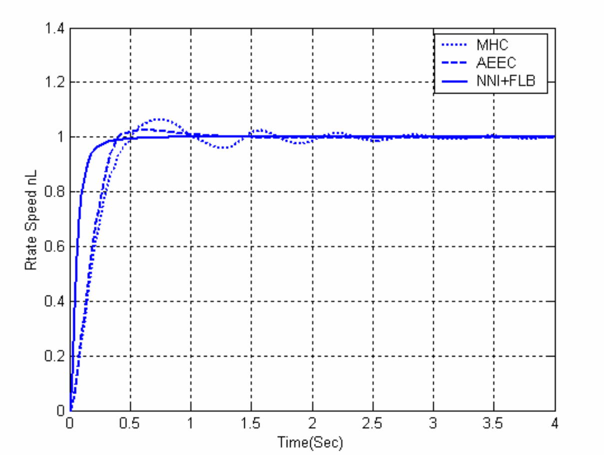


Figure.6 System response curve in ground testing

## VII. CONCLUSION

This paper proposed a new neural network algorithm with fuzzy logic compensation and applied to an aeroengine rotate speed control system. A dynamic neural network was used to identify the plant on-line. The control signal was then calculated iteratively according to the responses of a reference model and the output of identified plant. A fuzzy logic block with four very simple rules was added to the loop to improve the overall loop properties. Experimental results demonstrate the proposed control strategy provides better disturbance rejection and transient properties than those achieved by conventional MHC and AEEC.

## REFERENCE


[1] He Erming, Civil Aerial Engine Control Theory and its Typical Systems. Beijing: National Defense Publishing House. 2002.

[2] Zhang Sheng-liang, Xie Shou-sheng. “Test System for FADEC of an aeroengine”. Journal of Propulsion Technology, vol. 24(2), pp. 190~192, April 2003.

[3] Yan Pingfan, Zhang Changshui, Artificial Neutral Networks and Analog Evolution Algorithm. Beijing: Qinghua University Publishing House, 1999.

[4] D.Sbarbaro, J.P.Segovia, S.Alcozer and J. Gonzales. Applications of Radial Basis Network Technology to Process Control[J]. IEEE Trans. Control Systems Technology, 2000, 8(1): 14-20.

[5] Xie Shousheng, Certain Afterburning turbofan engine, Xi’an: The Air Force Engineering University Publishing House, 2002.

[6] Fan Siqi, Xu Yunhua, Aerial Propulsion System Control. Xi’an: Northwestern Technical University. 1995

[7] K. L. Moore, M. Dahleh and S. P. Bhattachargya. Iterative Learning Control: A Survey and NewResults[J]. Journal of Robotic Systems, 1992, 9(5): 563-594.

[8] T. C. Chen and C. Y. Liaw. Robust Speed controlled Induction Motor Drive Based on Model Reference with Neural Network[J]. International Journal of Knowledge-Based Intelligent Engineering Systems, 1999, 3(3): 162-171.