

# Strathprints Institutional Repository

Zhang, Jianhua and Zhou, Shuqing and Ren, Mifeng and Yue, Hong (2016) Adaptive neural network cascade control system with entropybased design. IET Control Theory and Applications, 10 (10). pp. 1151-1160. ISSN 1751-8644 , http://dx.doi.org/10.1049/iet-cta.2015.0992

This version is available at http://strathprints.strath.ac.uk/55772/

**Strathprints** is designed to allow users to access the research output of the University of Strathclyde. Unless otherwise explicitly stated on the manuscript, Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Please check the manuscript for details of any other licences that may have been applied. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (<u>http://strathprints.strath.ac.uk/</u>) and the content of this paper for research or private study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to Strathprints administrator: <a href="mailto:strathprints@strath.ac.uk">strathprints@strath.ac.uk</a>

# Adaptive Neural Network Cascade Control System with Entropybased Design

Jianhua Zhang<sup>1</sup>, Shuqing Zhou<sup>1</sup>, Mifeng Ren<sup>2</sup>, Hong Yue<sup>3\*</sup>

<sup>1</sup>State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, Beijing, 102206, China

<sup>2</sup> College of Information Engineering, Taiyuan University of Technology, Taiyuan, 030024, China

<sup>3</sup> Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G1 1XW, UK

\*hong.yue@strath.ac.uk

**Abstract:** A neural network (NN) based cascade control system is developed, in which the primary PID controller is constructed by NN. A new entropy-based measure, named the centred error entropy (CEE) index, which is a weighted combination of the error cross correntropy (ECC) criterion and the error entropy criterion (EEC), is proposed to tune the NN-PID controller. The purpose of introducing CEE in controller design is to ensure that the uncertainty in the tracking error is minimised and also the peak value of the error probability density function (PDF) being controlled towards zero. The NN-controller design based on this new performance function is developed and the convergent conditions are. During the control process, the CEE index is estimated by a Gaussian kernel function. Adaptive rules are developed to update the kernel size in order to achieve more accurate estimation of the CEE index. This NN cascade control approach is applied to superheated steam temperature control of a simulated power plant system, from which the effectiveness and strength of the proposed strategy are discussed by comparison with NN-PID controllers tuned with EEC and ECC criterions.

### 1. Introduction

Cascade control is an easy to apply and effective strategy to reject disturbances, improve dynamic responses and compensate process nonlinearities. Cascade controllers are widely used in various industrial systems, such as superheated temperature control systems in power plants [1-3], automatic generation control of multi-area thermal systems [4], electric motors [5-7], converters [8; 9], single-phase shunt active power filters[10], networked control systems [11-13], flow-level systems[14], brake-by-wire actuator of sport motorcycles [15], diesel engines [16], under-actuated ball-and-beam systems [17], air handling units [18], robotic systems [19], and many others. In a typical cascade control system, PID controllers are used in both the inner loop and the outer loop. To cope with nonlinearities, disturbances, integral wind-up and other complex issues in process systems, a number of cascade control strategies have been developed to improve either the outer loop performance or the inner loop performance, or both. Examples such as the LQ self-tuning controller [20], neuro-fuzzy generalized predictive controller [1], model predictive controllers [2; 6; 7; 19] and neural network (NN) based controllers [3; 18] were reported for the outer loop primary controller design. Control algorithms have also been developed for the design of the secondary controller in the inner loop, for example, model-reference adaptive control based on Kalman active

observer [19], predictive control [6; 7] and sliding mode control [9], to name a few. In addition, there are methods developed to improve both the primary and the secondary controllers. Several H $\infty$  control approaches were presented for networked cascade control systems [11-13]. A global anti-windup compensator design method based on linear matrix inequality (LMI) was developed for an under-actuated mechanical system [17].

Most of the above cascade control methods have been developed based on explicit mathematical models. In practice, accurate models are difficult to establish due to high nonlinearities, uncertainties and random disturbances involved in industrial processes. To cope with limitations in (explicit) model-based control strategies, model-free control or data-driven control approaches have been developed in recent years, such as model-free adaptive control, lazy learning control, dynamic programming strategy, iterative feedback tuning, unfalsified control, virtual reference feedback tuning [21-24]. Fuzzy logic systems (FLSs) and NNs, with strength in approximation of complex nonlinear systems, are used in modelling and controller design. Adaptive fuzzy control approaches have been developed for systems with unknown functions approximated by FLSs [25-27] or NNs [28-30]. In a model-free scheme, only the input and output measurement data are utilized in controller design; while process models and un-modelled dynamics are not required and assumptions on disturbance terms do not need to be strictly formed.

Model-free controllers have been well developed through minimising the mean square error (MSE) performance indexes when a Gaussian distribution or others alike can be assumed for stochastic terms involved in the system. However, MSE-based measures capture only up to the second-order statistics in the stochastic data, could be inadequate for arbitrary non-Gaussian and nonlinear systems. Different from an MSE index, an entropy function is a scalar quantity that measures the overall information contained in the distribution of a stochastic variable. Thus, the error entropy criteria (EEC) are considered to be of more general nature for non-Gaussian systems. To this end, EEC has been employed in both learning systems [31-34] and stochastic control systems [3; 35-45]. A minimum error entropy controller was developed for superheated steam temperature control in power plants [3] in our previous work. The so-called  $(h, \phi)$ - entropy has been employed in a networked control system [41], in which the  $(h, \phi)$ - entropy of the quadratic performance index was used to characterise the randomness of the closed-loop system. Other development in minimum entropy control includes robotic manipulator with joint trajectory [42], urban transportation network with transport entropy [43], stabilisation of chaotic systems [44; 45].

Although entropy-based indexes provide general measures for non-Gaussian systems, it should be noted that an entropy metric has the shift invariance property, that is, for two random variables with the same shape of probability density functions (PDFs), they may have the same entropy values but completely different mean values. This suggests that simply minimising entropy-based measures can be misleading in output tracking control. In order to drive the tracking error towards zero, improved minimum error entropy controllers have been developed by adding either the mean error term [36-40] or the mean squared error term [41] to the entropy control scheme. Different from minimising EEC, an alternative performance index, called the error cross correntropy (ECC) index, defined as an L<sub>2</sub>-norm in the error sampling space, can be used to control the main peak of the error PDF close to zero [31]. In this work, we propose to combine EEC and ECC together to formulate a performance index in a way that the tracking error can be controlled towards zero, and at the same time, the uncertainty in the stochastic error terms can be minimised. This new measure is named as the centred error entropy (CEE) criterion.

Effective control of superheated steam temperature is crucial for safe and efficient operations in power plants. Cascade controllers have been commonly used for this type of systems [3], where the inner loop is designed to roughly regulate the intermediate steam temperature, located between the spray water injection point and the high temperature super-heater, through manipulating the position of the spray water valve. The goal of the outer loop is to maintain a specified temperature level at the outlet of the high-temperature super-heater. The whole system is nonlinear, and the disturbances in the measured temperatures, induced by exhaust and vapour, are non-Gaussian in practice. This makes control of the superheated steam temperature a difficult task. In this work, we aim to improve this cascade control system with a focus on the primary controller using NN-based PID controller, and employ entropy-based rather than MSE criterion in the controller tuning. The NN-PID controller will be tuned using the new CEE index and the convergent conditions for this controller will be investigated. Since the controller design relies on the new performance index, adaptive rules will be developed to achieve more accurate estimation of the tracking error performance index over the control process.

The remaining of this paper is organized as follows. Section 2 presents the proposed design method for cascade control. Firstly the cascade control scheme is presented, then the CEE performance index is proposed for the primary NN-PID controller followed by adaptive tuning of the kernel size in entropy calculation. The tuning algorithm for the NN-PID controller is presented next using a linearization technique and the convergent conditions of the control strategy are investigated. Implementation procedures of this approach are summarised at the end of Section 2. In Section 3, this NN adaptive cascade control method is applied to superheated steam temperature control in a simulated industrial-scale power plant system. Comparisons are made with a cascade control system whose primary NN-PID controller is designed by a previously developed minimum entropy method [3]. Conclusions are given in Section 4.

# 2. Cascade control with NN-PID primary controller

The cascade control system proposed in this work includes an inner loop and an outer loop as shown in Fig. 1. The secondary plant (plant2) and the primary plant (plant1) are arranged in cascade series. The primary disturbance term,  $\omega$ , directly affects the primary output Y in the outer loop, and the secondary disturbance signal,  $\omega_2$ , mainly influences the secondary output,  $y_2$ , from the inner loop. The controller in the inner loop is called the secondary controller. It mainly regulates the secondary output ( $y_2$ ), which in turn reduces variations introduced to Plant1 in the primary loop. The proportional (P) or proportionalintegral (PI) controller in the inner-loop should be tuned such that the secondary control loop runs much faster than the primary loop.



Fig.1. Cascade control system with neuro-PID primary controller

The major purpose of the outer loop is to make the primary output (y) to follow its set-point ( $y_{sp}$ ). An NN- (PID) controller is used for the outer loop tracking control. At each time k, the tuning of this NN-PID controller is obtained through minimising the CEE performance index J (k). The value of J (k) is estimated based on Gaussian kernel functions, in which the kernel size,  $\sigma$ , is updated by the Kullback-Leibler divergence criterion (KLDC),  $J_{KL}(\sigma)$ , in order to obtain accurate estimation of J (k). The three parameters of the NN-PID controller,  $k_p$ ,  $k_i$ ,  $k_d$ , are tuned by a back-propagation (BP) optimisation algorithm using the proposed CEE performance index. The input to the neuro-PID controller is a sequence of the tracking errors in the outer loop, i.e.,  $e(k) = y_{sp}(k) - y(k)$ ,  $(k = 1, 2, \dots)$ . Disturbance signals can also be included in the NN input layer if they are measurable or can be estimated by certain means.

#### 2.1. Entropy-based Tracking Error Performance Index

Consider non-Gaussian disturbances in the nonlinear cascade control system (Fig.1), entropy-based metrics can be employed to measure the dispersion of the primary output variable and the tracking performance. In this context, the entropy of the tracking error should be minimised in order to reduce the uncertainty (or randomness) in the stochastic tracking error signal. This can be achieved by employing minimum entropy criterion in controller design to make the shape of the tracking error PDF as narrow as possible [21-23]. For such a purpose, the following quadratic performance index is formulated based on the Renyi's entropy of the tracking error [3].

$$H_{2}(e) = -\ln\left(\int_{-\infty}^{\infty} \gamma^{2}(e) de\right) = -\ln\left(V_{2}(e)\right)$$
(1)

where  $\ln(\cdot)$  is the logarithm function,  $\gamma(e)$  is the PDF for tracking error e, and  $V_2(e) = \int_{-\infty}^{\infty} \gamma^2(e) de$  is the quadratic information potential. It can be observed from (1) that minimising  $H_2(e)$  is equivalent to maximising the quadratic information potential term  $V_2(e)$ .

At each time k,  $V_2(e(k))$  can be estimated using the data of tracking errors within its receding horizon window whose width is denoted as L [3]. Let  $\bar{e}_{k-L+l:k} = [e(k)_1, e(k)_2, ..., e(k)_L]^T$  represent the vector of the tracking error sequence over the sliding window [k-L+1,k], the EEC can be estimated by

$$\overline{\mathbf{J}}(\mathbf{k}) = \hat{\mathbf{V}}_{2}(\mathbf{e}(\mathbf{k})) = \frac{1}{L^{2}} \sum_{j=1}^{L} \sum_{i=1}^{L} \mathbf{G}_{\sigma} \left( \mathbf{e}(\mathbf{k})_{j} - \mathbf{e}(\mathbf{k})_{i} \right)$$
(2)

where  $G_{\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$  is the Gaussian kernel function,  $\sigma$  is the kernel size or bandwidth,

here the centre of the kernel function is set to zero.

It is known that minimising EEC can reduce the uncertainty in tracking error dispersion but cannot guarantee the error term to be zero or close to zero. Therefore, an alternative performance index, ECC (error cross-correntropy), is introduced in the same sense as MSE in that it defines an  $L_2$ -norm in the error sampling space. The following definition of cross-correntropy [31] is presented firstly, which gives a generalised similarity measurement between two random variables e and  $\overline{e}$ , namely,

$$\mathbf{v}(\mathbf{e},\overline{\mathbf{e}}) = \mathbf{E}_{\mathbf{e}\overline{\mathbf{e}}} \left\{ \mathbf{G}_{\sigma} \left( \mathbf{e} - \overline{\mathbf{e}} \right) \right\}$$
(3)

where  $E\{\cdot\}$  is expectation function,  $\overline{e}$  is a zero-mean random variable distinguished from e. Sampling from the densities within a receding horizon window of width L, ECC can be estimated as follows:

$$\mathbf{v}(\mathbf{e},\overline{\mathbf{e}}) \approx \frac{1}{L} \sum_{i=1}^{L} \mathbf{G}_{\sigma} \left( \mathbf{e}_{i} - \overline{\mathbf{e}}_{i} \right) = \frac{1}{L} \sum_{i=1}^{L} \mathbf{G}_{\sigma} \left( \mathbf{e}_{i} \right) = \widetilde{\mathbf{J}} \left( \mathbf{k} \right)$$
(4)

The peak value of the tracking error PDF can be driven to be located within the vicinity of zero by maximising  $\tilde{J}(k)$  [31].

In order to establish the NN-PID controller with close-to-zero tracking errors and minimum tracking error uncertainty, a generalized performance index is formulated by combining the EEC in (2) with the ECC in (4), which is named as CEE index given as:

$$J(k) = \frac{(1-\lambda)}{L^{2}} \sum_{j=1}^{L} \sum_{i=1}^{L} G_{\sigma}(e(k)_{i} - e(k)_{j}) + \frac{\lambda}{L} \sum_{i=1}^{L} G_{\sigma}(e(k)_{i})$$
(5)

where  $\lambda$  is a constant weighting factor valued between 0 and 1. When  $\lambda=0$ , CEE in (5) reduces to EEC, and  $\lambda=1$  to ECC. Numerically, maximising the CEE performance index J is equivalent to minimising -J.

**Remark 1**: In the ideal case, the tracking error PDF should be a delta function, meaning that all the uncertainty in the tracking error is eliminated [31], which can be achieved by minimising the entropy of the tracking error. The quadratic information potential of the tracking error is the mean value of the error PDF, i.e.,  $V_2(e) = E\{\gamma(e)\}$ .

**Remark 2**: The goal of the NN- based cascade control (Fig. 1) is to drive the system output to be "as close as possible" to the set-point. The concept of closeness in this context has two aspects: randomness and distance. The randomness (or uncertainty) in tracking error dispersion can be reduced by minimising EEC. However, the entropy function is shift-invariant, which means the global minimum value of error entropy can be located at any position in the variable horizon. The distance between the output and the set-point signals (also regarded as amplitude of error signal) can be measured by ECC, which contains second-and higher-order moments of a PDF [31]. The proposed CEE performance index (5) therefore can be used to reduce both randomness and amplitude of the error signal.

#### 2.2. Adaptive Tuning of Kernel Size

It can be seen from (2) and (4) that the kernel size  $\sigma$  is a crucial parameter for estimation of both EEC and ECC, thus CEE in (5). When the weights of the NN-PID controller are updated at each control

step, the kernel size should also be updated so as to follow the varied characteristics in tracking errors. For this reason, the KLDC measure between the real tracking error PDF and the estimated tracking error PDF is employed to update the kernel size at each control step.

Given a batch of L samples within the sliding window [k-L+1,k], the estimated PDF using the Gaussian kernel function with bandwidth  $\sigma$  is

$$\hat{\gamma}_{\sigma}\left(\mathbf{e}(\mathbf{k})\right) = \frac{1}{L} \sum_{i=1}^{L} \mathbf{G}_{\sigma}\left(\mathbf{e}(\mathbf{k}) - \mathbf{e}(\mathbf{k})_{i}\right)$$
(6)

From an information-theoretic perspective, an optimal value of  $\sigma$  should be the one that minimises the discrimination information between the estimated PDF and the true PDF of the tracking errors. Such a discrimination function can be formulated as

$$D_{KL}(\gamma \| \hat{\gamma}_{\sigma}) = \int \gamma(e) \ln\left(\frac{\gamma(e)}{\hat{\gamma}_{\sigma}(e)}\right) de$$

$$= \int \gamma(e) \ln \gamma(e) de - \int \gamma(e) \ln\left(\hat{\gamma}_{\sigma}(e)\right) de$$
(7)

The first term on the right side of (7) is independent of the kernel size  $\sigma$ . Therefore, minimising  $D_{KL}(\gamma \| \hat{\gamma}_{\sigma})$  with respect to  $\sigma$  is equivalent to maximising  $\int \gamma(e) \ln(\hat{\gamma}_{\sigma}(e)) de$ . The latter is the crossentropy between the real PDF and the estimated PDF calculated by the kernel function. The optimisation performance index is thus simplified to be

$$\mathbf{J}_{\mathrm{KL}}(\boldsymbol{\sigma}) = \mathbf{E}\left\{\ln\left(\hat{\boldsymbol{\gamma}}_{\boldsymbol{\sigma}}(\mathbf{e})\right)\right\}$$
(8)

Using L samples within the sliding window,  $J_{KL}(\sigma)$  in (8) can be estimated by

$$\hat{\mathbf{J}}_{\mathrm{KL}}(\sigma) = \frac{1}{L} \sum_{i=1}^{L} \ln \left( \sum_{j=1}^{L} \mathbf{G}_{\sigma} \left( \mathbf{e}_{i} - \mathbf{e}_{j} \right) \right)$$
(9)

In this case, the kernel size can be updated using the gradient-based searching rule.

$$\sigma(\mathbf{k}+1) = \sigma(\mathbf{k}) + \eta_1 \frac{\partial \hat{\mathbf{J}}_{\mathrm{KL}}(\sigma(\mathbf{k}))}{\partial \sigma(\mathbf{k})}$$
(10)

where  $\eta_1$  is the learning rate factor, and the first-order derivative term in (10) is calculated by

$$\frac{\partial \hat{\mathbf{J}}_{\mathrm{KL}}(\sigma)}{\partial \sigma} = \mathrm{E}\left\{\frac{\sum_{i=1}^{L} \exp\left(-\frac{(\mathbf{e}-\mathbf{e}_{i})^{2}}{2\sigma^{2}}\right) \left(\frac{(\mathbf{e}-\mathbf{e}_{i})^{2}}{\sigma^{3}} - \frac{1}{\sigma}\right)}{\sum_{i=1}^{L} \exp\left(-\frac{(\mathbf{e}-\mathbf{e}_{i})^{2}}{2\sigma^{2}}\right)}\right\}$$
(11)

A stochastic approximation of the gradient can be made by dropping the expectation operator in (11) [34], which makes the final updating rule written as follows:

$$\sigma(\mathbf{k}+1) = \sigma(\mathbf{k}) + \eta_1 \frac{\partial \hat{J}_{KL}(\sigma(\mathbf{k}))}{\partial \sigma(\mathbf{k})}$$
$$= \sigma(\mathbf{k}) + \eta_1 \frac{\sum_{i=k-L}^{k-1} \exp\left(-\frac{1}{2}\phi(\mathbf{k})_i\right)(\phi(\mathbf{k})_i - 1)}{\sigma(\mathbf{k})\sum_{i=k-L}^{k-1} \exp\left(-\frac{1}{2}\phi(\mathbf{k})_i\right)}$$
(12)

where  $\phi(k)_{i} = \frac{(e(k) - e(k)_{i})^{2}}{\sigma^{2}(k)}$ .

**Remark 3**: To compute the CEE performance index in (5), the Parzen density estimation technique is applied, where the kernel size parameter  $\sigma$  has important effects on the properties of the produced cost functions (perhaps even more important than the choice of the kernel function itself [31]). Since the tracking error statistics varies over the control process due to stochastic disturbances, using fixed-width kernels in each control step may become inadequate to describe the tracking errors. Instead, employing the adaptive kernel width provides a more accurate estimation of CEE for this NN cascade control system.

#### 2.3. NN-PID Controller Tuning Algorithm

In the outer loop, the following PID controller is constructed by a three-layer BP NN.

$$u(k) = u(k-1) + k_{p} (e(k) - e(k-1)) + k_{i}e(k) + k_{d} (e(k) - 2e(k-1) + e(k-2))$$
(13)

where  $k_p$ ,  $k_i$  and  $k_d$  are controller parameters tuned by the NN (see Fig. 1). The input vector to the neural network is  $\mathbf{O}^{(1)} = \left[\mathbf{O}_1^{(1)}, \dots, \mathbf{O}_Q^{(1)}\right]^T = \left[\mathbf{e}(\mathbf{k}), \dots, \mathbf{e}(\mathbf{k}-\overline{\mathbf{n}}), \boldsymbol{\omega}(\mathbf{k}), \boldsymbol{\omega}_2(\mathbf{k})\right]^T \in \mathbb{R}^Q$  ( $\overline{\mathbf{n}}$  is the length of error data entered in the input layer), and the output vector from the NN contains the three PID parameters, i.e.,  $\mathbf{O}^{(3)} = \left[\mathbf{O}_1^{(3)}, \mathbf{O}_2^{(3)}, \mathbf{O}_3^{(3)}\right]^T = \left[\mathbf{k}_p, \mathbf{k}_i, \mathbf{k}_d\right]^T \in \mathbb{R}^3$ . The hidden layer is written as  $\mathbf{O}^{(2)} = \left[\mathbf{O}_1^{(2)}, \dots, \mathbf{O}_P^{(2)}\right]^T \in \mathbb{R}^P$  with P nodes included. The weighting matrix between the input layer and the hidden layer is

 $\mathbf{W}^{(2)}(\mathbf{k}) = \left[ \mathbf{w}_{pq}^{(2)}(\mathbf{k}) \right] \in \mathbb{R}^{P \times Q}, \text{ and the weighting matrix between the hidden layer and the output layer is}$  $\mathbf{W}^{(3)}(\mathbf{k}) = \left[ \mathbf{w}_{lq}^{(3)}(\mathbf{k}) \right] \in \mathbb{R}^{3 \times P}. \text{ Here the superscripts (1), (2) and (3) refer to the input layer, hidden layer and the output layer of in NN, respectively.}$ 

The input and output of the hidden layer are represented as

$$net_{p}^{(2)}(k) = \sum_{q=0}^{Q} w_{pq}^{(2)}(k) O_{q}^{(1)}(k), \qquad q = 0, 1, ..., Q$$
(14)

$$O_p^{(2)}(k) = f\left(net_p^{(2)}(k)\right), \qquad p = 0, 1, ..., P$$
 (15)

The input and the output of the output layer are

$$\operatorname{net}_{l}^{(3)}(k) = \sum_{p=0}^{P} w_{lp}^{(3)}(k) O_{p}^{(2)}(k), \qquad l = 1, 2, 3$$
(16)

$$O_l^{(3)}(k) = g(net_l^{(3)}(k)), \qquad l = 1, 2, 3$$
 (17)

The activation functions are  $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$  and  $g(x) = \frac{e^x}{e^x + e^{-x}}$  for the hidden layer and the output layer respectively.

layer, respectively.

The controller design problem can then be transformed into updating the weighting matrices  $\mathbf{W}^{(2)}(\mathbf{k})$  and  $\mathbf{W}^{(3)}(\mathbf{k})$  by maximising the performance index J (k) in (5). The steepest descent approach is adopted to find the optimised solution in this work. Firstly, the learning rule of the weighting matrix between the hidden and the output layers can be obtained using the gradient algorithm:

$$w_{lp}^{(3)}(k+1) = w_{lp}^{(3)}(k) + \eta \cdot \frac{\partial J(k)}{\partial w_{lp}^{(3)}(k)}$$
(18)

where  $\eta$  is the learning rate for training  $w_{lp}^{(3)}$ . For simplicity, k is omitted in following equation.

$$\frac{\partial \mathbf{J}}{\partial \mathbf{w}_{lp}^{(3)}} = \frac{\lambda}{L} \sum_{i=1}^{L} \left[ \mathbf{G}_{\sigma}^{'}\left(\mathbf{e}_{i}\right) \frac{\partial \mathbf{e}_{i}}{\partial \mathbf{w}_{lp}^{(3)}} \right] + \frac{(1-\lambda)}{L^{2}} \sum_{i=1}^{L} \sum_{j=1}^{L} \left[ \mathbf{G}_{\sigma}^{'}\left(\mathbf{e}_{i}-\mathbf{e}_{j}\right) \left( \frac{\partial \mathbf{e}_{i}}{\partial \mathbf{w}_{lp}^{(3)}} - \frac{\partial \mathbf{e}_{j}}{\partial \mathbf{w}_{lp}^{(3)}} \right) \right]$$

$$\frac{\partial \mathbf{e}}{\partial \mathbf{w}_{lp}} = -\frac{\partial \mathbf{y}}{\partial \mathbf{w}_{lp}} = -\frac{\partial \mathbf{y}}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{O}_{l}^{(3)}} \cdot \frac{\partial \mathbf{O}_{l}^{(3)}}{\partial \operatorname{net}_{l}^{(3)}} \cdot \frac{\partial \operatorname{net}_{l}^{(3)}}{\partial \mathbf{w}_{lp}}$$
(19)
$$(20)$$

In (19),  $G_{\sigma}$  is the first-order derivative of  $G_{\sigma}$ . The Jacobian information  $\partial y/\partial u$  in (20) can be replaced by  $sgn(\partial y/\partial u)$  or calculated by model prediction algorithm. Other derivative terms at time k can be

represented as

$$\frac{\partial O_{l}^{(3)}(k)}{\partial net_{l}^{(3)}(k)} = g'(net_{l}^{(3)}(k))$$

$$\frac{\partial net_{l}^{(3)}(k)}{\partial w_{lp}(k)} = O_{p}^{(2)}(k)$$

$$\frac{\partial u(k)}{\partial O_{l}^{(3)}(k)} = \begin{cases} e(k) - e(k-1), & l=1\\ e(k), & l=2\\ e(k) - 2e(k-1) + e(k-2), & l=3 \end{cases}$$
(21)

g' is the derivative function of g. Hence, at time k, the connecting weights between the hidden layer and the output layer can be calculated from (18)-(21).

Similarly, the weights between the input and the hidden layers can be updated by the following laws:

$$w_{pq}^{(2)}(k+1) = w_{pq}^{(2)}(k) + \eta \cdot \frac{\partial J(k)}{\partial w_{pq}^{(2)}(k)}$$
(22)

Again, drop the time term k for simplification, there is

$$\frac{\partial J}{\partial w_{pq}^{(2)}} = \frac{\lambda}{L} \sum_{i=1}^{L} \left[ G_{\sigma}^{'}(e_{i}) \frac{\partial e_{i}}{\partial w_{pq}^{(2)}} \right] + \frac{(1-\lambda)}{L^{2}} \sum_{i=1}^{L} \sum_{j=1}^{L} \left[ G_{\sigma}^{'}(e_{i}-e_{j}) \left( \frac{\partial e_{i}}{\partial w_{pq}^{(2)}} - \frac{\partial e_{j}}{\partial w_{pq}^{(2)}} \right) \right]$$

$$\frac{\partial e}{\partial w_{pq}^{(2)}} = -\frac{\partial y}{\partial w_{pq}^{(2)}} = -\frac{\partial y}{\partial w_{pq}^{(2)}} = -\frac{\partial y}{\partial u} \sum_{i=1}^{L} \frac{\partial u}{\partial O_{i}^{(3)}} \cdot \frac{\partial O_{i}^{(3)}}{\partial net_{i}^{(3)}} \cdot \frac{\partial net_{i}^{(3)}}{\partial O_{p}^{(2)}} \cdot \frac{\partial O_{p}^{(2)}}{\partial net_{p}^{(2)}} \cdot \frac{\partial net_{p}^{(2)}}{\partial w_{pq}^{(2)}}$$

$$(24)$$

$$\frac{\partial O_{i}^{(2)}}{\partial u} \sum_{i=1}^{L} \frac{\partial u}{\partial O_{i}^{(3)}} \cdot \frac{\partial O_{i}^{(3)}}{\partial net_{i}^{(3)}} \cdot \frac{\partial O_{p}^{(2)}}{\partial O_{p}^{(2)}} \cdot \frac{\partial O_{p}^{(2)}}{\partial net_{p}^{(2)}} \cdot \frac{\partial net_{p}^{(2)}}{\partial w_{pq}^{(2)}} = \frac{\partial O_{i}^{(2)}}{\partial w_{pq}^{(2)}} = \frac{\partial O_{i}^{(2)}}{\partial u} \sum_{i=1}^{L} \frac{\partial O_{i}^{(3)}}{\partial O_{i}^{(3)}} \cdot \frac{\partial O_{i}^{(3)}}{\partial O_{p}^{(2)}} \cdot \frac{\partial O_{p}^{(2)}}{\partial net_{p}^{(2)}} \cdot \frac{\partial O_{i}^{(2)}}{\partial w_{pq}^{(2)}} = \frac{\partial O_{i}^{(2)}}{\partial u} \sum_{i=1}^{L} \frac{\partial O_{i}^{(3)}}{\partial O_{i}^{(3)}} \cdot \frac{\partial O_{i}^{(3)}}{\partial O_{p}^{(2)}} \cdot \frac{\partial O_{i}^{(2)}}{\partial o_{i}^{(2)}} \cdot \frac{\partial O_{i}^{(2)}}{\partial w_{pq}^{(2)}} = \frac{\partial O_{i}^{(2)}}{\partial v_{pq}^{(2)}} = \frac{\partial O_$$

$$\frac{\partial O_{p}^{(2)}}{\partial net_{p}^{(2)}} = f'(net_{p}^{(2)})$$
(25)

$$\frac{\partial \text{net}_{l}^{(3)}}{\partial O_{p}^{(2)}} = w_{lp}^{(3)}$$
(26)

$$\frac{\partial \text{net}_{p}^{(2)}}{\partial w_{pq}^{(2)}} = \mathbf{O}_{q}^{(1)}$$
(27)

f' is the first-order derivative function of f.

**Remark 4**: The learning rate,  $\eta$ , is a key factor for the NN-controller tuning algorithm in (18)-(27). The larger is the learning rate, the faster is the learning process. However, a large learning rate may cause

unexpected oscillations in the training process or even a non-convergent learning result. Thus a trade-off needs to be made for the speed and convergence.  $\eta$  can be time-varying if necessary. For simplicity, a constant value within the range of [0, 1] is chosen for  $\eta$  in this paper.

**Remark 5:** The NN-PID controller is a model-free controller. Only sampled input and output data are required in controller tuning.

**Remark 6:** It should be noted that the proposed adaptive NN cascade control strategy can handle disturbance rejection conveniently within the inner loop. This is the inherent nature of the cascade control structure. It doesn't matter whether the disturbance signal is measurable or not. When the disturbance signal is measurable or can be estimated by some means, they can be included in the input layer to the NN-PID controller. As a result, the NN-PID controller can function to reject the disturbances in a manner similar to a conventional feedforward controller. If the disturbance is unmeasurable or cannot be included in the input loop, it may deteriorate the tracking performance. This will be reflected in the proposed CEE index, which can be handled by the adaptive tuning of NN and its weightings. Therefore, this proposed NN-PID controller has overall enhanced performance in disturbance rejection.

#### 2.4. Convergence Conditions

In this proposed algorithm, the CEE performance index in (5) is used for NN-controller tuning. The calculation of the CEE index is dependent to both current and past error data, since at each time k, a sequence of tracking errors within the sliding window horizon are collected to update the kernel size. Also the adopted kernel function in entropy calculation is a nonlinear function. These make it difficult to investigate convergence conditions of the NN-controller especially when no explicit model is used. A linearisation technique is employed in the following to find convergence conditions for this adaptive NN-PID controller.

Rewrite the two weighting matrices in columns as  $\mathbf{W}^{(2)} = \left[ \left( \mathbf{W}_{1}^{(2)} \right)^{T}, \left( \mathbf{W}_{2}^{(2)} \right)^{T}, \cdots, \left( \mathbf{W}_{P}^{(2)} \right)^{T} \right]^{T}$  and  $\mathbf{W}^{(3)} = \left[ \left( \mathbf{W}_{1}^{(3)} \right)^{T}, \left( \mathbf{W}_{2}^{(3)} \right)^{T}, \left( \mathbf{W}_{3}^{(3)} \right)^{T} \right]^{T}$ , where  $\mathbf{W}_{i}^{(2)} \in \mathbb{R}^{1 \times Q}$  (i = 1, 2, ..., P) and  $\mathbf{W}_{i}^{(3)} \in \mathbb{R}^{1 \times P}$  (i = 1, 2, 3) are the i<sup>th</sup> rows in  $\mathbf{W}^{(2)}$  and  $\mathbf{W}^{(3)}$ , respectively. Rearrange all the NN weights into a single vector as  $\mathbf{W} = \left[ \mathbf{W}_{1}^{(2)}, \mathbf{W}_{2}^{(2)}, \cdots, \mathbf{W}_{P}^{(2)}, \mathbf{W}_{1}^{(3)}, \mathbf{W}_{2}^{(3)}, \mathbf{W}_{3}^{(3)} \right]^{T} \in \mathbb{R}^{PQ+3P}$ . At time k, the NN update rules, (18) and (22), can then be written in a compact format as follows

$$\mathbf{W}(\mathbf{k}+1) = \mathbf{W}(\mathbf{k}) + \eta \nabla \mathbf{J} \left( \mathbf{W}(\mathbf{k}) \right)$$
(28)

where

$$\nabla J \left( \mathbf{W}(\mathbf{k}) \right) = \partial J \left( \mathbf{k} \right) / \partial \mathbf{W}(\mathbf{k})$$

$$= \left[ \frac{\partial J \left( \mathbf{k} \right)}{\partial w_{11}^{(2)}}, \cdots, \frac{\partial J \left( \mathbf{k} \right)}{\partial w_{1Q}^{(2)}}, \cdots, \frac{\partial J \left( \mathbf{k} \right)}{\partial w_{P1}^{(2)}}, \cdots, \frac{\partial J \left( \mathbf{k} \right)}{\partial w_{PQ}^{(2)}}, \frac{\partial J \left( \mathbf{k} \right)}{\partial w_{11}^{(3)}},$$

$$\cdots, \frac{\partial J \left( \mathbf{k} \right)}{\partial w_{1P}^{(3)}}, \cdots, \frac{\partial J \left( \mathbf{k} \right)}{\partial w_{31}^{(3)}}, \cdots, \frac{\partial J \left( \mathbf{k} \right)}{\partial w_{3P}^{(3)}} \right]^{\mathrm{T}} \in \mathbb{R}^{1 \times (PQ+3P)}$$

is the gradient vector of J (k) over the weights, in which all elements can be obtained from (18)-(27). Note that  $G_{\sigma}(x) = G_{\sigma}(x/\sigma)/\sigma$ , and  $G_{\sigma}(x) = G_{\sigma}(x/\sigma)/\sigma^2$ . Dropping time k for simplicity, the elements of  $\nabla J$  (**W**) can be rewritten as follows

$$\frac{\partial \mathbf{J}}{\partial \mathbf{w}_{lp}^{(3)}} = \frac{\lambda}{\sigma^2 \mathbf{L}} \sum_{i=1}^{L} \left[ \mathbf{G}_{\sigma}^{'} \left( \mathbf{e}_i / \sigma \right) \frac{\partial \mathbf{e}_i}{\partial \mathbf{w}_{lp}^{(3)}} \right] + \frac{(1 - \lambda)}{\sigma^2 \mathbf{L}^2} \\ \cdot \sum_{i=1}^{L} \sum_{j=1}^{L} \left[ \mathbf{G}_{\sigma}^{'} \left( (\mathbf{e}_i - \mathbf{e}_j) / \sigma \right) \cdot \left( \frac{\partial \mathbf{e}_i}{\partial \mathbf{w}_{lp}^{(3)}} - \frac{\partial \mathbf{e}_j}{\partial \mathbf{w}_{lp}^{(3)}} \right) \right] \\ \frac{\partial \mathbf{J}}{\partial \mathbf{w}_{pq}^{(2)}} = \frac{\lambda}{\sigma^2 \mathbf{L}} \sum_{i=1}^{L} \left[ \mathbf{G}^{'} \left( \mathbf{e}_i / \sigma \right) \frac{\partial \mathbf{e}_i}{\partial \mathbf{w}_{pq}^{(2)}(\mathbf{k})} \right] + \frac{(1 - \lambda)}{\sigma^2 \mathbf{L}^2} \\ \cdot \sum_{i=1}^{L} \sum_{j=1}^{L} \left[ \mathbf{G}^{'} \left( \left( \mathbf{e}_i - \mathbf{e}_j \right) / \sigma \right) \cdot \left( \frac{\partial \mathbf{e}_i}{\partial \mathbf{w}_{pq}^{(2)}} - \frac{\partial \mathbf{e}_j}{\partial \mathbf{w}_{pq}^{(2)}} \right) \right]$$

Assume the optimal weighting vector at steady state is  $\mathbf{W}^*$ . Apply first-order Taylor expansion to the gradient term  $\nabla J(\mathbf{W}(k))$  around  $\mathbf{W}^*$ , there is

$$\nabla \mathbf{J}(\mathbf{W}) = \nabla \mathbf{J}(\mathbf{W}^*) + \mathbf{R}(\mathbf{W} - \mathbf{W}^*)$$
(29)

where **R** is the Hessian matrix, i.e.,  $\mathbf{R} = \frac{\partial \nabla J^{\mathrm{T}}(\mathbf{W})}{\partial \mathbf{W}} \Big|_{\mathbf{W}=\mathbf{W}^{*}} \in \mathbb{R}^{PQ+3P}$ , in which

$$\begin{split} \frac{\partial^{2} \mathbf{J}}{\partial w_{pq}^{(2)} \partial w_{lp}^{(3)}} &= \frac{\lambda}{\sigma^{2} L} \sum_{i=l}^{L} \left[ \frac{1}{\sigma} \mathbf{G}_{\sigma}^{"}(\mathbf{e}_{i}/\sigma) \frac{\partial \mathbf{e}_{i}}{\partial w_{lp}^{(3)}} \frac{\partial \mathbf{e}_{i}}{\partial w_{pq}^{(2)}} + \mathbf{G}_{\sigma}^{'}(\mathbf{e}_{i}/\sigma) \frac{\partial \mathbf{e}_{i}^{2}}{\partial w_{pq}^{(2)} \partial w_{lp}^{(3)}} \right] \\ &+ \frac{(1-\lambda)}{\sigma^{2} L^{2}} \sum_{i=l}^{L} \sum_{j=l}^{L} \left[ \frac{1}{\sigma} \mathbf{G}_{\sigma}^{"}(\mathbf{e}_{i}/\sigma) \left( \frac{\partial \mathbf{e}_{i}}{\partial w_{lp}^{(3)}} - \frac{\partial \mathbf{e}_{j}}{\partial w_{lp}^{(3)}} \right) \left( \frac{\partial \mathbf{e}_{i}}{\partial w_{pq}^{(2)}} - \frac{\partial \mathbf{e}_{j}}{\partial w_{pq}^{(2)}} \right) \\ &+ \mathbf{G}_{\sigma}^{'} \left( \left( \mathbf{e}_{i} - \mathbf{e}_{j} \right) / \sigma \right) \left( \frac{\partial \mathbf{e}_{i}^{2}}{\partial w_{pq}^{(2)} \partial w_{lp}^{(3)}} - \frac{\partial \mathbf{e}_{j}^{2}}{\partial w_{pq}^{(2)} \partial w_{lp}^{(3)}} \right) \right] \end{split}$$

12

and

$$\begin{split} & \frac{\partial e_{i}^{2}}{\partial w_{pq}^{(2)} \partial w_{lp}^{(3)}} \\ &= \frac{\partial \left( -\frac{\partial y}{\partial u} \sum_{l} \frac{\partial u}{\partial O_{l}^{(3)}} \cdot O_{q}^{(1)} \cdot f^{'}\left(net_{p}^{(2)}\right) \cdot g^{'}\left(net_{l}^{(3)}\right) \cdot w_{lp}^{(3)}\right)}{\partial w_{lp}^{(3)}} \\ &= -\frac{\partial y}{\partial u} \sum_{l} \frac{\partial u}{\partial O_{l}^{(3)}} \cdot O_{q}^{(1)} \cdot f^{'}\left(net_{p}^{(2)}\right) \\ & \cdot \left( g^{''}\left(net_{l}^{(3)}\right) \sum_{p=0}^{P} O_{p}^{(2)} \cdot w_{lp}^{(3)} + g^{'}\left(net_{l}^{(3)}\right) \right) \end{split}$$

Substituting (29) into (28), and subtracting  $\mathbf{W}^*$  from both sides of the equation, we have

$$\mathbf{W}(\mathbf{k}+1) = \mathbf{W}(\mathbf{k}) + \eta \mathbf{R} \mathbf{W}(\mathbf{k}) \tag{30}$$

where  $\overline{\mathbf{W}}(\mathbf{k}) = \mathbf{W}^* - \mathbf{W}(\mathbf{k})$ . Denote  $\mathbf{\Omega} = \overline{\mathbf{Q}}^T \overline{\mathbf{W}} \in \mathbb{R}^{P_{Q+3P}}$ ,  $\overline{\mathbf{Q}} \in \mathbb{R}^{(P_{Q+3P}) \times (P_{Q+3P})}$  is the orthonormal matrix consisting of the eigenvalues of **R**. Thus

$$\mathbf{\Omega}(\mathbf{k}+1) = (\mathbf{I}+\eta\mathbf{\Lambda})\mathbf{\Omega}(\mathbf{k}) \tag{31}$$

where  $\Lambda$  is the diagonal eigenvalue matrix with entries,  $\lambda_i$ , ordered corresponding to the ordering in  $\overline{\mathbf{Q}}$ . For each entry in  $\Omega$ , there is

$$\Omega_{1}(k+1) = (1+\eta\lambda_{1})\Omega_{1}(k), \ i = 1, 2, ..., PQ+3P$$
(32)

and its mean value calculation is

$$E\{\Omega_{1}(k+1)\} = E\{(1+\eta\lambda_{1})\Omega_{1}(k)\}$$
(33)

Since the eigenvalues of the Hessian matrix **R** are negative, the condition of  $|1 + \eta \lambda_i| < 1$  can guarantee a stable dynamics [32; 34]. Therefore, the following condition can ensure that the proposed algorithm is convergent:

$$\begin{cases} \lambda_{i} < 0, \\ 0 < \eta < \frac{2}{\max(|\lambda_{i}|)} & i = 1, 2, ..., P(Q+1) + 3(P+1) \end{cases}$$
(34)

Furthermore, the approximate time constant corresponding to  $\lambda_i$  is

$$\tau_{i} = \frac{-1}{\log(1 + \eta\lambda_{i})} \approx \frac{1}{\eta|\lambda_{i}|}$$
(35)

It can be concluded that the weights of the proposed NN-PID controller can be converged when the condition in (34) is satisfied. In addition, the learning rate should be chosen to increase the CEE

performance index in (5) along with time, which raises another convergent condition as described in the following theorem.

**Theorem 1:** The CEE performance index in (5) will be strictly increasing with respect to sliding windows, if the chosen learning rate can satisfy the following nonlinear inequality:

$$\eta \left(\frac{\partial J(k)}{\partial W(k)}\right)^{1} \frac{\partial J(k-1)}{\partial W(k-1)} > 0$$
(36)

where

$$\frac{\partial \mathbf{J}}{\partial \mathbf{W}} = \left[\frac{\partial \mathbf{J}}{\partial \mathbf{w}_{11}^{(2)}}, \cdots, \frac{\partial \mathbf{J}}{\partial \mathbf{w}_{1Q}^{(2)}}, \cdots, \frac{\partial \mathbf{J}}{\partial \mathbf{w}_{P1}^{(2)}}, \cdots, \frac{\partial \mathbf{J}}{\partial \mathbf{w}_{PQ}^{(2)}}, \frac{\partial \mathbf{J}}{\partial \mathbf{w}_{11}^{(3)}}, \cdots, \frac{\partial \mathbf{J}}{\partial \mathbf{w}_{31}^{(3)}}, \cdots, \frac{\partial \mathbf{J}}{\partial \mathbf{w}_{31}^{(3)}}, \cdots, \frac{\partial \mathbf{J}}{\partial \mathbf{w}_{3P}^{(3)}}\right]^{\mathrm{T}}$$

can be obtained from (18)-(27).

Proof: For an increasing CEE performance index, there is

$$J(k-1) < J(k) \tag{37}$$

This will lead to the following condition to achieve the convergence:

$$\frac{\partial J(k)}{\partial k} > 0 \tag{38}$$

Using the same chain principles as in (18)-(27), it can be verified that the above condition will result in the following inequality:

$$\left(\frac{\partial J(k)}{\partial W(k)}\right)^{T} \Delta W(k) = \eta \left(\frac{\partial J(k)}{\partial W(k)}\right)^{T} \frac{\partial J(k-1)}{\partial W(k-1)} > 0$$
(39)

The conditions provided in (34) and (39) would guarantee the closed-loop convergence of the NN-PID controller.

#### 2.5. Controller Design Procedures and Parameter Settings

Figure 1 shows the schematic diagram of the NN cascade control system. The implementation procedures of the proposed approach are summarised as follows:

1) Tune the secondary P or PI controller for the inner loop, and set the receding horizon L for the sliding window.

2) Estimate the CEE performance index in (5) using the sequence of the tracking errors within the sliding window.

- 3) Update the kernel size using (12).
- 4) Update the weights of the primary NN-PID controller using (18)-(27).
- 5) On obtaining  $k_p$ ,  $k_i$  and  $k_d$  via the NN outputs, compute the control input with (13).
- 6) Apply the control input to the cascade system, increase k to k+1, repeat the procedure from 2) to 5) until the end of the control process.

To effectively implement the proposed algorithm, several key parameters need to be set up properly. This includes parameters for NN training such as the learning rate ( $\eta$ ) and the number of nodes in the hidden layer (Q); and other parameters such as the window width (L) in the receding horizon window technique, the learning factor ( $\eta_1$ ) for update of kernel size, and the weighting factor ( $\lambda$ ) in the CEE index. The selection rules for these parameters are briefed as follows.

- The learning rate, η, affects the rate of convergence in NN training and the training result. A larger η makes a faster learning process. However, a large learning rate may cause unexpected oscillations in the NN training process or even a non-convergent learning result. Thus a trade-off needs to be made for η. The selection of η must satisfy the convergent conditions given in (34) and (39).
- The learning rate factor  $\eta_1$  in (10) is used to update the kernel size. Here the steepest descent approach is employed, where a larger value of  $\eta_1$  will give a faster but more sparse search and a smaller value of  $\eta_1$  will make a slower but refined search. A trade-off needs to be made to reach a balance of search speed and searching precision.
- The number of nodes (Q) in the hidden layer corresponds to the NN structure. If Q is too small, the NN may be inadequate for approximating the nonlinear system; if Q is too large, the computational cost is inevitably large, also the NN may provide an overfitting from the training data. Again a trade-off needs to be made. In practice, Q is normally determined by the number of inputs and the number of outputs of the NN.
- The window width (L) in the receding horizon window technique should be chosen large enough to contain adequate number of tracking errors in historical time horizon. If L is large, then it collects more data on tracking errors but is computationally more expensive. In general L can be chosen according to the order of the system.
- The weighting constant  $\lambda$  in the CEE performance index (5) is in the range of 0 and 1. It

reflects the balance between EEC and ECC in the proposed index.

#### 3. Case study: superheated steam temperature control

The proposed NN cascade control approach is applied to regulate the superheated steam temperature of a boiler in a 300 MW power plant using our previously developed process model [3]. In the simulation, the sampling period is taken to be  $T_s = 1 s$ , the transfer functions of the primary plant and the secondary

plant are 
$$G_{\text{plant1}}(s) = \frac{2.09}{(1+22.3s)^4}$$
 and  $G_{\text{plant2}}(s) = -\frac{2.01}{(1+16s)^2}$ , respectively [3]. The NN-PID controller is

used for the primary controller, and the secondary controller is a proportional controller with  $G_{C2}(s) = 8$ . The control results obtained in [3] are also presented to support the comparison, in which a NN-PID controller designed on EEC is served as the primary controller. The secondary controller in [3] is the same proportional controller as in this work.

In the development of the NN-PID controller, the receding horizon window technique is used to estimate the CEE index. The window width should be large enough to cover the dynamic characteristics of the plant and it is set to be L = 200 here. The weighting factor in the CEE performance index is set to be  $\lambda = 0.6$  as a good compromise between contribution of EEC and ECC in the proposed index. For NN-PID training, the learning rate influences the rate of convergence and the training results. If the learning rate is chosen to be small, the NN is more likely to be stable but the training cost will be large. On the other hand, using a large learning rate will speed up the training process but may cause unexpected oscillations or even divergent results. Considering the convergent condition in (34) and (39), the NN learning rate is set to be  $\eta = 0.00015$ . The learning factor for update of the kernel size is chosen as  $\eta_1 = 0.0001$ . The inputs to the NN-PID controller include three tracking errors e(k), e(k-1), e(k-2), and the measured disturbance signals  $\omega$  and  $\omega_2$ . Therefore there are five inputs at the input layer of the NN-PID controller (P = 5). The number of outputs is 3 for the three PID terms. The number of the hidden nodes is selected experimentally to be Q = 4.

In this NN cascade control method, EEC and CEE are combined in the CEE performance function to tune the primary NN-PID controller. In addition, the kernel size to estimate the performance index is updated using the KLDC. In the following simulations, three controllers are attempted using EEC, ECC and the proposed CEE indexes, respectively. To make a reasonable comparison, adaptive adjustment of the kernel size is applied to all three controllers; all NN-PID controllers take the same structure and the parameters for NNs are also kept at the same values.

In the simulation study, the superheated process operates at a steady state from the beginning until 500s. The set point of the superheated steam temperature takes a step increase at 500s from  $535^{\circ}$ C to  $545^{\circ}$ C. The responses of the superheated steam temperature under EEC control, ECC control and the proposed CEE control are shown in Fig.2. With all controllers, the system can be stabilised and the superheated steam temperature is controlled towards the neighbourhood around the set point with small oscillations. The mean value and the standard deviation of the tracking errors under the three controllers are listed in Table 1. It can be observed that with the CEE index, the time response of the proposed method achieves the best tracking performance among the three controllers.



Fig.2. Response of the superheated steam temperature under three controllers

Table 1 Mean and variance of tracking errors under three controllers

criterion	mean	variance
EEC	0.4470	3.1054
ECC	0.4947	3.4196
CEE	0.3436	2.8452

The parameter variations of the three NN-PID controllers are illustrated in Fig. 3. In Fig.4, under the three controllers, the performance index EEC is shown in Fig.4 (a), the ECC is shown in Fig.4 (b), and the proposed CEE index is shown in Fig.4 (c). It can be seen that all the three criterions are numerically compatible, as indicated by their formulations in (2), (4) and (5), respectively. When using different criterions in controller design, the best performance is always achieved for the particular index used in controller design. This shows that using the proposed CEE index, a balance can be conveniently achieved between the previously used EEC and ECC indexes.



Fig.3. Tuning of NN-PID parameters under 3 performance indexes



(c) Time profile of the CEE index in (5) **Fig.4.** Time profile of the EEC, ECC and CEE indexes under three controllers

Fig. 5 illustrates the tracking error PDF evolution using the proposed NN-PID controller. The three PDFs at the selected sampling time points are shown in Fig. 6.



Fig.5. PDF evolution of tracking errors under the proposed neuro-PID controller



Fig.6. Selected tracking error PDFs under the proposed neuro-PID controller

The efficiency of adaptive changing of the kernel size is also investigated with the temperature control system. Using the proposed CEE index, simulations are made under fixed kernel size ( $\sigma = 1$ ) and the updated kernel size. The KLDC performance indexes are shown in Fig. 7, from which it can be seen that the controller with updated kernel size achieves a better performance (larger value of KLDC).



Fig.7. Time profile of the performance index (9) used to update kernel size

#### 4. Conclusions

In this paper, an adaptive NN-PID controller is proposed for the primary controller design in a cascade control system. To enhance the tracking performance for non-Gaussian and nonlinear stochastic systems, a new CEE performance index is proposed to train the NN-PID controller in the primary control loop, which combines the minimum entropy and the maximum cross correntropy of the tracking error. The kernel function for this entropy-based performance index is recursively updated by employing a receding horizon window technique. The training algorithm of the NN-PID controller is derived using the steepest descent approach, and its convergence conditions are investigated with the use of a linearisation technique.

This algorithm is applied to simulation study of superheated steam temperature control in an industrial-scale boiler system. The controller performances are investigated for several entropy-based metrics. Simulation results demonstrate with the proposed method, a balanced performance of reducing randomness and amplitude of tracking errors can be achieved conveniently.

The proposed method expands cascade control to model-free scheme for nonlinear, non-Gaussian, single-input and single-output (SISO) control systems. While positive results have been achieved in theoretical discussions and numerical studies based on an industrial-scale system model, it needs a lot more efforts to apply the proposed controller to real industrial systems. Implementation issues with PI/PID controllers, noises measurement and estimation, constraints with actuators and temperatures, require further investigation.

# 5. Acknowledgments

This work was supported by National Basic Research Program of China under Grant (973 Program, Grant No. 2011CB710706), the Natural Science Foundation of China (61503271, 61374052 and 61511130082) and Beijing (4142048). These are gratefully acknowledged.

# 6. References

- Liu, X.-J., Chan, C.: 'Neuro-fuzzy generalized predictive control of boiler steam temperature', IEEE Trans. Energy Convers., 2006, 21, (4), pp. 900-908
- [2] Silva, R., Shirley, P., Lemos, J., Goncalves, A.: 'Adaptive regulation of super-heated steam temperature: a case study in an industrial boiler', Control Eng. Pract., 2000, **8**, (12), pp. 1405-1415
- [3] Zhang, J., Zhang, F., Ren, M., Hou, G., Fang, F.: 'Cascade control of superheated steam temperature with neuro-PID controller', ISA Trans., 2012, **51**, (6), pp. 778-785
- [4] Dash, P., Saikia, L. C., Sinha, N.: 'Automatic generation control of multi area thermal system using Bat algorithm optimized PD–PID cascade controller', Int. J. Elec. Power, 2015, **68**, 364-372
- [5] Son, Y. I., Kim, I. H., Choi, D. S., Shim, H.: 'Robust cascade control of electric motor drives using dual reducedorder PI observer', IEEE Trans. Ind. Electron., 2015, 62, (6), pp. 3672-3682
- [6] Chai, S., Wang, L., Rogers, E.: 'A cascade MPC control structure for PMSM with speed ripple minimization', IEEE Trans. Ind. Electron., 2013, **60**, (8), pp. 2978-2987
- [7] Errouissi, R., Ouhrouche, M., Chen, W.-H., Trzynadlowski, A. M.: 'Robust cascaded nonlinear predictive control of a permanent magnet synchronous motor with antiwindup compensator', IEEE Trans. Ind. Electron., 2012, 59, (8), pp. 3078-3088
- [8] Bocker, J., Freudenberg, B., The, A., Dieckerhoff, S.: 'Experimental comparison of model predictive control and cascaded control of the modular multilevel converter', IEEE Trans. Power Electron., 2015, **30**, (1), pp. 422-430
- [9] Chen, Z., Gao, W., Hu, J., Ye, X.: 'Closed-loop analysis and cascade control of a nonminimum phase boost converter', IEEE Trans. Power Electron., 2011, **26**, (4), pp. 1237-1252
- [10] Abouloifa, A., Giri, F., Lachkar, I., et al.: 'Cascade nonlinear control of shunt active power filters with average performance analysis', Control Eng. Pract., 2014, **26**, 211-221
- [11] Huang, C., Bai, Y., Liu, X.: 'H-infinity state feedback control for a class of networked cascade control systems with uncertain delay', IEEE Trans. Ind. Informat., 2010, **6**, (1), pp. 62-72
- [12] Mathiyalagan, K., Park, J. H., Sakthivel, R.: 'New results on passivity-based H∞ control for networked cascade control systems with application to power plant boiler - turbine system', Nonlinear Anal-Hybri., 2015, 17, 56-69
- [13] Du, Z., Yue, D., Hu, S.: 'H-infinity stabilization for singular networked cascade control systems with state delay and disturbance', IEEE Trans. Ind. Informat., 2014, **10**, (2), pp. 882-894
- [14] Li, C., Choudhury, M. S., Huang, B., Qian, F.: 'Frequency analysis and compensation of valve stiction in cascade control loops', J. Process Contr., 2014, 24, (11), pp. 1747-1760
- [15] Todeschini, F., Corno, M., Panzani, G., Fiorenti, S., Savaresi, S. M.: 'Adaptive cascade control of a brake-bywire actuator for sport motorcycles', IEEE/ASME Trans. Mechatronics, 2014, **20**, (3), pp. 1310-1319
- [16] Tschanz, F., Zentner, S., Onder, C. H., Guzzella, L.: 'Cascaded control of combustion and pollutant emissions in diesel engines', Control Eng. Pract., 2014, 29, 176-186
- [17] Mehdi, N., Rehan, M., Malik, F. M., Bhatti, A. I., Tufail, M.: 'A novel anti-windup framework for cascade control systems: An application to underactuated mechanical systems', ISA Trans., 2014, **53**, (3), pp. 802-815
- [18] Guo, C., Song, Q., Cai, W.: 'A neural network assisted cascade control system for air handling unit', IEEE Trans. Ind. Electron., 2007, **54**, (1), pp. 620-628
- [19] Dominici, M., Cortesão, R.: 'Cascade force control for autonomous beating heart motion compensation', Control Eng. Pract., 2015, 37, 80-88
- [20] Fessl, J., Jarkovský, J.: 'Cascade control of superheated steam temperature with lq self-tuning controllerss of conference paper'. Prep. 8th IFAC Symp. Identification and System Parameter Estimation, Place of Conference, 1988, pp. 408-413

- [21] Spall, J. C., Cristion, J. A.: 'Model-free control of nonlinear stochastic systems with discrete-time measurements', IEEE Trans. Autom. Control, 1998, **43**, (9), pp. 1198-1210
- [22] Xu, D., Jiang, B., Shi, P.: 'A novel model-free adaptive control design for multivariable industrial processes', IEEE Trans. Autom. Control, 2014, **61**, (11), pp. 6391-6398
- [23] Wei, Q., Liu, D.: 'Data-driven neuro-optimal temperature control of water–gas shift reaction using stable iterative adaptive dynamic programming', IEEE Trans. Autom. Control, 2014, **61**, (11), pp. 6399-6408
- [24] Bu, X., Hou, Z., Yu, F., Wang, F.: 'Robust model free adaptive control with measurement disturbance', Control Theory Appl., 2012, 6, (9), pp. 1288-1296
- [25] Tong, S.-C., Sui, S., Li, Y.-M.: 'Fuzzy adaptive output feedback control of MIMO nonlinear systems with partial tracking errors constrained', IEEE Trans. Fuzzy. Syst., 2015, 23, (4), pp. 729–742
- [26] Liu, Y.-J., Tong, S.: 'Adaptive fuzzy identification and control for a class of nonlinear pure-feedback MIMO systems with unknown dead zones', IEEE Trans. Fuzzy. Syst., 2015, 23, (5), pp. 1387–1398
- [27] Tong, S.-C., Zhang, L., Li, Y.-M.: 'Observed-based adaptive fuzzy decentralized tracking control for switched uncertain nonlinear large-scale systems with dead zones', IEEE Trans. Syst., Man, Cybern., Syst., 2016, 46, (1), pp. 37–47
- [28] Tong, S.-C., Wang, T., Li, Y.-M., Zhang, H.-G.: 'Adaptive neural network output feedback control for stochastic nonlinear systems with unknown dead-zone and unmodeled dynamics', IEEE Trans. Cybern., 2014, 44, (6), pp. 910–921
- [29] Liu, Y.-J., Tong, S.: 'Adaptive NN tracking control of uncertain nonlinear discrete-time systems with nonaffine dead-zone input', IEEE Trans. Cybern., 2015, 45, (3), pp. 497–505
- [30] Liu, Y.-J., Tang, L., Tong, S., Chen, C. L. P.: 'Adaptive NN controller design for a class of nonlinear MIMO discrete-time systems', IEEE Trans. Neur. Net. Lear. Syst., 2015, 26, (5), pp. 1007–1018
- [31] Chen, B., Zhu, Y., Hu, J.: 'Mean-square convergence analysis of ADALINE training with minimum error entropy criterion', IEEE Trans. Neural Netw., 2010, **21**, (7), pp. 1168-1179
- [32] Chen, B., Zhu, Y., Hu, J., Principe, J. C.: in 'System parameter identification: information criteria and algorithms' (Newnes, 2013, 1st edn).
- [33] Erdogmus, D., Principe, J. C.: 'An error-entropy minimization algorithm for supervised training of nonlinear adaptive systems', IEEE Trans. Signal Process, 2002, **50**, (7), pp. 1780-1786
- [34] Principe, J. C.: in 'Information theoretic learning: Renyi's entropy and kernel perspectives' (Springer Science & Business Media, 2010, edn).
- [35] Guo, L., Wang, H.: in 'Stochastic distribution control system design: a convex optimization approach' (Springer Science & Business Media, 2010, 1st edn).
- [36] Yue, H., Wang, H.: 'Minimum entropy control of closed-loop tracking errors for dynamic stochastic systems', IEEE Trans. Autom. Control, 2003, **48**, (1), pp. 118-122
- [37] Yue, H., Zhou, J., Wang, H.: 'Minimum entropy of B-spline PDF systems with mean constraint', Automatica, 2006, **42**, (6), pp. 989-994
- [38] Ren, M., Zhang, J., Wang, H.: 'Minimized tracking error randomness control for nonlinear multivariate and non-Gaussian systems using the generalized density evolution equation', IEEE Trans. Autom. Control, 2014, 59, (9), pp. 2486-2490
- [39] Zhang, J., Ren, M., Wang, H.: 'Minimum entropy control for non-linear and non-Gaussian two-input and twooutput dynamic stochastic systems', IET Control Theory Appl., 2012, **6**, (15), pp. 2434-2441
- [40] Ren, M., Zhang, J., Jiang, M., Yu, M., Xu, J.: 'Minimum entropy control for non-Gaussian stochastic networked control systems and its application to a networked DC motor control system', IEEE Trans. Control Syst. Technol., 2015, 23, (1), pp. 406-411
- [41] Zhang, X., Ren, X., Na, J., Zhang, B., Huang, H.: 'Adaptive nonlinear neuro-controller with an integrated evaluation algorithm for nonlinear active noise systems', J. Sound Vib., 2010, **329**, (24), pp. 5005-5016
- [42] Chen, H., Xing, G., Sun, H., Wang, H.: 'Brief paper-Indirect iterative learning control for robot manipulator with non-Gaussian disturbances', IET Control Theory Appl., 2013, 7, (17), pp. 2090-2102
- [43] Zhou, H., Bouyekhf, R., El Moudni, A.: 'Modeling and entropy based control of urban transportation network', Control Eng. Pract., 2013, 21, (10), pp. 1369-1376
- [44] Sadeghpour, M., Salarieh, H., Alasty, A.: 'Controlling chaos in tapping mode atomic force microscopes using improved minimum entropy control', Appl. Math. Model., 2013, 37, (3), pp. 1599-1606

[45] Sadeghpour, M., Salarieh, H., Alasty, A.: 'Minimum entropy control of chaos via online particle swarm optimization method', Appl. Math. Model., 2012, 36, (8), pp. 3931-3940