Gaussian networks for fuel injection control

C Manzie1, M Palaniswami* and H Watson2
1Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, Victoria, Australia
2Department of Manufacturing and Mechanical Engineering, The University of Melbourne, Parkville, Victoria, Australia

Abstract: This paper proposes a radial basis function (RBF) based approach for the fuel injection control problem. In the past, neural controllers for this problem have centred on using a cerebellar model articulation controller (CMAC) type network with some success. The current production engine control units also use look-up tables in their fuel injection controllers, and if adaptation is permitted to these look-up tables the overall effect closely mimics the CMAC network. Here it is shown that an RBF network with significantly fewer nodes than a CMAC network is capable of delivering superior control performance on a mean value engine model simulation. The proposed approach requires no a priori knowledge of the engine systems, and on-line learning is achieved using gradient descent updates. The RBF network is then implemented on a four-cylinder engine and, after a minor modification, outperforms a production engine control unit.

Keywords: Gaussian networks, air–fuel ratio control, radial basis functions

NOTATION

Throughout this paper a large number of symbols are used. In general, these are defined the first time each symbol is used. For completeness, the following notation is provided as a reference to the reader. Note that in this list, and throughout the remainder of the paper, the term ‘flow’ is used to refer to a flow of mass per unit time.

\[ w \] network weight
\[ X \] fuel split parameter
\[ \beta_{obs} \] observed/measured air–fuel ratio
\[ \beta_{stoich} \] stoichiometric air–fuel ratio
\[ \eta_{vol} \] volumetric efficiency
\[ \tau_f \] time constant of evaporation of the fuel film
\[ \omega \] engine speed

1 INTRODUCTION

It is well known that air–fuel ratio (AFR) control represents a difficult problem in the automotive control field. It is necessary to maintain the AFR close to its stoichiometric value to ensure vehicle driveability and low pollutant levels, as stoichiometry represents the maximum point of catalytic converter efficiency for the three main pollutants of the spark ignition (SI) engine. Variations of less than 1 per cent above stoichiometry can result in a dramatic increase in CO and HC emissions, while an increase of little more than 1 per cent involves NO\textsubscript{x} emissions increasing by up to 50 per cent. Proposed government regulations such as the 1988 California Clean Air Act require improvements in emissions by an 84 per cent decrease in HC, 74 per cent decrease in NO\textsubscript{x} and 60 per cent reduction in CO by 2003. A recent Environmental Protection Authority survey [1] reported that 50–90 per cent of all air...
pollutants, including carbon monoxide, nitrogen oxides and volatile organic compounds, in the city of Melbourne during the 1995–6 period were directly related to automotive exhausts. There are also other benefits to accurate AFR control such as improved fuel efficiency.

The prohibitive cost of in-cylinder sensors capable of measuring the mass of air in the cylinder means that indirect control methods relying on other measurements must be utilized. The engine is a highly non-linear system, and this precludes the use of conventional linear controllers from achieving the desired objective of maintaining the AFR within a small boundary of stoichiometry. Previously, engine control modules (ECMs) have been used to form extensive look-up tables that dictate the amount of fuel to be injected for any engine condition. Without adaptation, this method has been shown to be unsatisfactory for the high level of accuracy desired [2].

Much of the previous research into AFR controllers has centred around model-based controllers [3–5] where an accurate model of various systems of the engine is developed and used to determine the mass of fuel to be injected. Model-based controllers with no adaptive capability have significant drawbacks limiting their implementation in a production line scenario. Engine wear and individual engine variances mean that the controller cannot perform at an optimal level over time and large numbers of vehicles. An adaptive technique that can make the necessary adjustments required for an individual engine is clearly preferable, as demonstrated in reference [6]. More recently, production ECMs have been equipped with the capability to update their look-up tables in order to allow the controller to cope with changing engine conditions. This method of adapting a look-up table is very similar to the idea of the cerebellar modular articulation controller (CMAC) neural network, which has explicitly been employed in fuel injection research by Majors [2] and in reference [7] with some success. The purpose of this paper is to demonstrate that radial basis function (RBF) networks offer greater promise in the field of fuel injection control.

Neural networks have been shown to be able to represent non-linear mappings successfully [8]. Among the many variations of networks developed, RBFs have been shown to possess the ‘best approximation property’ [9] and have been successfully applied to other control problems including adaptive control for bank-to-turn missiles [10], robot arm control [11] and automated car parking [12]. Thus, they have significant potential for non-linear controllers such as those required by the AFR control problem.

This paper is formulated as follows. In Section 2 some of the basic mathematical models formulated to describe the engine systems are described. A preliminary examination of the CMAC and RBF neural networks is also included. Section 3 describes how these neural schemes are implemented into the control problem, while Section 4 describes the simulation parameters and provides simulation results. Finally, in Section 5 the controller with the best simulation performance is implemented on a four-cylinder Mitsubishi Magna engine.

### 2.1 Mean value engine model

The engine is an inherently non-linear system, and accurate modelling is extremely difficult unless some assumptions are made. The mean value engine model (MVEM) is a mathematical formulation of the engine, which involves decoupling the engine into three independent systems under the assumption that engine speed is treated as a time-varying parameter. The three distinct systems are the fuel, the air and the engine dynamic systems, and these systems are modelled independently. Cho and Hedrick [13] provide a full description of the three-state MVEM. For the present simulation of the engine, this model has been adopted for the air and engine dynamic systems. However, a limitation of the fuel system was pointed out in reference [14], so a slightly more complex fuelling model as used by Hendricks et al. [15] has been used, based on the model by Aquino [16]. The other assumption made is that temperature is approximately constant for simulation purposes. This is because the algorithm will be implemented on a warmed engine. Hendricks et al. [15] demonstrate that there is only a weak temperature dependence on the fuel split parameter, $X$, but the fuel time constant is influenced by temperature—this variation is not considered in this paper. Only the parts of the model relative to the discussion in this paper are reproduced. Interested readers are directed towards the papers listed above for a more comprehensive discussion.

The simplified fuel model describing the fuel pooling process is given by

$$\dot{m}_{fu} = (1 - X)\dot{m}_{fc}$$  \hspace{1cm} (1a)

$$\dot{m}_f = \frac{1}{\tau_f}(-\dot{m}_f + X\dot{m}_{fc})$$  \hspace{1cm} (1b)

$$\dot{m}_{fc} = \dot{m}_{fc} + \dot{m}_f$$  \hspace{1cm} (1c)

These three equations allow the mass rate of fuel entering the cylinder, $\dot{m}_f$, to be modelled. The parameters $X$ and $\tau_f$ are dependent on engine operating point, and their functional form was determined experimentally and is detailed in Section 3.

The mass rate of air entering the combustion chamber, $m_{so}$, is modelled as a scaled product of mass of air in the intake manifold, $m_a$ (which is analogous to the intake manifold pressure, $P_{man}$), and engine speed, $\omega$:

$$m_{so} = \frac{\zeta}{\eta_{ew}}m_ao \hspace{1cm} (2)$$

The first symbol on the right-hand side of equation (2)
is a constant and equal to

$$\zeta = \frac{V_c}{4\pi V_m}$$  \hspace{1cm} (3)

The volumetric efficiency, $\eta_{vol}$, is a complex function of many engine parameters including port geometry, valve timing and intake manifold pressure [13]. It is a measure of the induction process effectiveness of the engine and is extremely difficult to model analytically. The following equation was used in the simulation work:

$$\eta_{vol} = (24.5\omega - 3.1 \times 10^4)m_c^2 + (-0.167\omega + 222)m_a$$

$$+ (8.1 \times 10^{-4}\omega + 0.352)$$  \hspace{1cm} (4)

The simulations detailed in Section 4 utilize the MVEM described above in a multitasking real-time engine simulation program developed at The University of Melbourne.

### 2.2 Cerebellar model articulation controller (CMAC) neural network

The CMAC network was developed by Albus [17] in 1975. CMAC networks have previously been used in fuel injection control (e.g. references [2] and [18]), with some success claimed by the authors. In a sense, the CMAC neural network is similar to an adaptive look-up table. It accomplishes the look-up table function by dividing the input space into a grid of dimension equal to the number of inputs, $n$. When the input comes into the network it is first mapped into a binary vector, $I$, of length $m$, where all those elements within some distance of the input are denoted 1 and all other elements are 0. A generalization parameter, $C$, is used to determine how many 1’s are present in the binary vector.

Once the binary vector is obtained, it is used to denote an active region corresponding to all the 1’s. A vector of weights, $W$, with each weight corresponding to an element of $I$, forms the final stage of the CMAC network. The output of the CMAC is obtained by summing all the active weights:

$$\hat{y} = W^T I = \sum_{j=1}^{N} w_j j_j$$  \hspace{1cm} (5)

The output is then compared with the desired value, and the weights may then be adjusted by a learning rule such as the recursive LMS weight update:

$$w(k + 1) = w(k) + \mu[y_{des}(k) - \hat{y}(k)] \frac{I}{C}$$  \hspace{1cm} (6)

This ensures that, with a learning rate $\mu$ in the range $0 < \mu < 2$, the CMAC output will approach the desired output, $y_{des}(k)$.

### 2.3 Radial basis function networks

Radial basis function networks have one hidden layer with a non-linear transfer function and the output layer, as illustrated in Fig. 1. The outputs of the hidden layer are dependent on the Euclidean distance of the input from the location of the centre corresponding to the hidden layer node.

Several types of radial function are used in the RBF networks, such as the inverse multiquadric and thin plate splines function, although for control applications the Gaussian transfer function is common:

$$g_i(x) = \exp\left(-\frac{|x - x_i|^2}{2\sigma_i^2}\right)$$  \hspace{1cm} (7)

The location of the $i$th centre and its width are given by $x_i$ and $\sigma_i$ respectively. Each output layer weight is denoted $w_i$, hence the output of the RBF network with $N$ nodes, for a given input $x$, is

$$\hat{y}(x) = \sum_{i=1}^{N} w_i g_i(x)$$  \hspace{1cm} (8)

Note that a comparison of equations (5) and (8) demonstrates that the CMAC network provides a piecewise approximation to the continuous RBF network, with $i(x)$ a binary approximation to $g_i(x)$. In the case of the RBF network, there are three parameters that may be updated: the centre weight, the centre position and the centre width. The following three lemmas derive Lyapunov stable update laws for each parameter. The update laws will be required to remain Lyapunov stable while reducing the estimation error, $E(x)$, where

$$E(x) = \hat{f}(x) - \hat{f}(x)$$  \hspace{1cm} (9)

In equation (9), $\hat{f}(x)$ represents the network output and $\hat{f}(x)$ represents the optimal output for the network of size $N$. Because of the best approximation property, it is known that there is a unique parameter set defining $\hat{f}(x)$ [9]. The superscript $^*$ is used to represent those quantities that are current estimates and therefore may be updated. For example, the output of a network with
N nodes and all parameters capable of being updated would be
\[ \hat{f}(x) = \sum_{i=1}^{N} \tilde{w}_i \exp\left(-\frac{|x - \hat{c}_i|^2}{2\sigma_i^2}\right) \] (10)

In the following lemmas, adaptation laws are derived for the case where one of the three sets of parameters is being updated at any time. The parameter set, \( P_{A,B} \), is considered to have \( A \) and \( B \) held fixed.

**Lemma 1**

Consider any continuous function, \( f(x) \), and an RBF network estimate of this function which is subjected to updates in the weights only. The function estimate is defined in equation (8):
\[ \hat{f}(x) = \sum_{i=1}^{N} \tilde{w}_i \exp\left(-\frac{|x - \hat{c}_i|^2}{2\sigma_i^2}\right) \] (11)

The weight update law
\[ \dot{\tilde{w}} = -k_w E(x) \] (12)

satisfies the Lyapunov stability criteria, where \( E(x) \) denotes the estimation error and \( k_w \) is a positive constant.

**Proof.** For the function being approximated, \( f(x) \), denote the best estimate using an RBF network with \( N \) centres and the fixed parameter set \( P_{C,S} \) as
\[ \hat{f}(x) = \sum_{i=1}^{N} \tilde{w}_i \exp\left(-\frac{|x - \hat{c}_i|^2}{2\sigma_i^2}\right) \] (13)

Now let the approximation to this best estimate with the same fixed parameter set and number of centres be denoted by
\[ \hat{f}(x) = \sum_{i=1}^{N} \tilde{w}_i \exp\left(-\frac{|x - c_i|^2}{2\sigma_i^2}\right) \] (14)

Using the definition of estimation error given in (9), the following Lyapunov function candidate may be defined:
\[ V = \frac{1}{2} E(x)^2 \] (15)
\[ \dot{V} = E(x) \dot{\tilde{w}} \exp\left(-\frac{|x - c_i|^2}{2\sigma_i^2}\right) \] (16)

Hence, the choice of the weight update law, \( \dot{\tilde{w}} \), given in equation (12) ensures Lyapunov stability for an appropriately chosen positive constant \( k_w \).

**Remark 1**

This update law can be modified to provide a more intuitively pleasing result containing an exponential term in the gain, i.e.
\[ k_w = k_e \exp\left(-\frac{|x - c_i|^2}{2\sigma_i^2}\right) \] (17)

Thus, the weight update law (12) becomes
\[ \dot{\tilde{w}} = -k_e \exp\left(-\frac{|x - c_i|^2}{2\sigma_i^2}\right) E(x) \] (18)

This ensures that those nodes whose contribution to the output is greatest will be subjected to the largest changes in parameters. The concept of the exponential term in the update law is maintained in the following two lemmas.

**Lemma 2**

Consider a given function, \( f(x) \), and an RBF network estimate of \( f(x) \), with centre-only updating:
\[ \hat{f}(x) = \sum_{i=1}^{N} w_i \exp\left(-\frac{|x - \hat{c}_i|^2}{2\sigma_i^2}\right) \] (19)

The centres update law
\[ \dot{\hat{c}}_i = k_c E(x) w_i \exp\left(-\frac{|x - \hat{c}_i|^2}{2\sigma_i^2}\right) \left(\frac{\hat{c}_i - x}{\sigma_i^2}\right) \] (20)

satisfies the Lyapunov stability criteria.

**Proof.** The structure of this proof is the same as for Lemma 1, so is omitted here.

**Lemma 2**

For a given function, \( f(x) \), and an RBF network estimate of \( f(x) \), which is subject to updating of widths only:
\[ \hat{f}(x) = \sum_{i=1}^{N} w_i \exp\left(-\frac{|x - c_i|^2}{2\sigma_i^2}\right) \] (21)

The width update law
\[ \dot{\sigma}_i = -k_e E(x) w_i \exp\left(-\frac{|x - c_i|^2}{2\sigma_i^2}\right) \left(\frac{|x - c_i|^2}{\sigma_i^4}\right) \] (22)

satisfies the Lyapunov stability criteria.

**Proof.** The structure of this proof is the same as for Lemma 1, so is omitted here.

**Remark 2**

A basic relaxation algorithm can be used to update all parameters while maintaining Lyapunov stability. This relaxation algorithm consists of three steps:

1. The centres and widths are held fixed (i.e. their update is temporarily ‘relaxed’) and the weights are updated towards the best solution using the update law given in equation (18).
2. The weights and widths are fixed while the centres are updated according to the law in lemma 2.
3. The widths are updated according to the update law in lemma 3.
3 APPLICATION OF NEURAL NETWORKS TO A FUEL INJECTION CONTROL SCHEME

In the proposed control approach, a neural network is used to quasi-estimate the amount of air entering the cylinder based on measurements of manifold pressure and engine speed. This estimate is then divided by the required air–fuel ratio to determine the amount of fuel to inject. Because of the presence of the fuel film, there will be an error involved with injecting this amount of fuel. This error can be associated with either the fuel or air system with the same overall effect; i.e. if too much fuel is injected, the estimate of air flow can be reduced in order to reduce the amount of fuel fed into the cylinder. Hence, it is not the true air flow that is being estimated but a similar quantity, which will be termed \( \dot{m}_{\text{ao}} \) and is defined as

\[
\dot{m}_{\text{ao}} = \dot{m}_{\text{a}} \frac{m_{\text{fi}}}{m_{\text{fc}}}
\]  

This approach is similar to the method of Shiriashi et al. [18] who assumed that measurements of engine speed and manifold pressure are sufficient fully to describe the current state of the engine, under the assumption that temperature is a slow varying parameter in comparison. The proposed approach hinges on the assumption that the trajectories of the input through the parameter space are unique for different throttle conditions. For example, if a tip in throttle situation occurs, the relatively high pressure and low engine speed will mean more fuel will be injected to compensate for the fuel film dynamics. A schematic of the neural network control scheme is included in Fig. 2.

Any feedback-based update law, such as those derived in Lemmas 1 to 3, requires the error in previous training points to be ascertained, and this error to be used as a factor in the update. However, the problems associated with the inability to measure directly certain engine variables, e.g. mass of air in the cylinder, provides a dilemma for traditional weight update. Since no feedback information regarding \( \dot{m}_{\text{ao}} \) is directly available through measurement, the observed AFR is used to develop an appropriate feedback term.

From the definitions of air–fuel ratio and the quasi-air flow parameter \( \dot{m}_{\text{ao}} \), it follows that

\[
\frac{\dot{m}_{\text{ao(des)}}}{\beta_{\text{stoich}}} = \dot{m}_{\text{ao}} = \frac{m_{\text{fi}}}{m_{\text{fc}}}
\]  

(24)

where \( \beta_{\text{stoich}} \) and \( \beta_{\text{obs}} \) are the stoichiometric and observed AFR respectively.

Hence, it is possible to restate the error between the desired output and the actual output as

\[
e(k) = \dot{m}_{\text{ao(des)}} - \dot{m}_{\text{ao}} = \dot{m}_{\text{ao}} \left( \frac{\dot{m}_{\text{ao(des)}}}{\dot{m}_{\text{ao}}} - 1 \right)
\]  

(25)

\[
e(k) = \dot{m}_{\text{ao}} \left( \frac{\beta_{\text{stoich}}}{\beta_{\text{obs}}} - 1 \right)
\]  

(26)

Therefore, substituting equation (26) into the weight update law given in equation (18) yields the update

\[
w_i(k + 1) = w_i(k) - k_w \dot{m}_{\text{ao}} \left( \frac{\beta_{\text{stoich}}}{\beta_{\text{obs}}} - 1 \right) \times g_i(P_{\text{man}}(t), \omega(t))
\]  

(27)

3.1 Time delay compensation

For both controllers, compensation of time delay in measurement of AFR involves synchronizing the inputs to the network \( (P_{\text{man}}, \omega) \) with the observed AFR. This time delay is a function of three parts:

1. There is a two engine cycle delay between the injection of fuel and the expulsion from the exhaust valves.
2. There is a propagation delay for the exhaust gases to reach the lambda meter, which may or may not be dependent on engine speed.
3. There is a sensor output delay.

By perturbing the fuel supply at a wide range of engine speeds and intake manifold pressures, the time delay was measured as the time until the associated perturbation on air–fuel ratio was observed. The sensor response is that of a low-pass filter, so the time difference was taken to be that until the air–fuel ratio reached 67 per cent of the maximum. Manifold pressure was found to have a negligible impact on the delay, and the speed dependence is demonstrated in Fig. 3.

The line of best fit of the data points for engine speed given in rad/s is

\[
t_d = 0.045 + \frac{10\pi}{\omega}
\]  

(28)

Note that this delay differs from the expected speed.
dependence of two engine revolutions plus a constant transport delay. The reason for the difference may be because the propagation time of the exhaust gases to the AFR sensor is dependent on engine speed.

A buffer of \( L \) consecutive cells was set up to store the inputs \((P_{\text{man}}, \omega)\) and output \((\hat{m}_{ao})\) of the network. Each cell corresponds to a sampling time. By recording the current position of the buffer, it is easy to match a parameter set and AFR measurement by evaluating when the time delay has expired. Thus, the weight update law (27) is expressed as

\[
w_i(k + 1) = w_i(k) - k_w \hat{m}_{ao} \left( \frac{\hat{P}_{\text{stoich}}}{\hat{P}_{\text{obs}}} - 1 \right) \\
	imes g_i(P_{\text{man}}(t - t_d), w(t - t_d))
\]  

(29)

### 3.2 Fuel parameters

The fuel parameters were obtained from the Magna engine by perturbing the fuel supply with a square wave and adjusting the model parameters until a square wave appeared on the (delay-compensated) air–fuel ratio measurement. Hendricks \textit{et al.} proposed this procedure \cite{15}. The relationship at constant warm engine temperature was found to be speed dependent with the following forms:

\[
\dot{X} = 9.6 \times 10^{-5} \omega + 0.7236
\]

(30)

\[
\tau_f = 1.67 \omega^{-0.65}
\]

(31)

These relations were used in the simulations to model the engine response, but not in determining the controller output.

### 4 ENGINE SIMULATION

#### 4.1 Simulation parameters

##### 4.1.1 CMAC network parameters

The parameter range mapped in the CMAC neural network was for pressures from 0 to 104 kPa and engine speeds from 0 to 750 rad/s. The quantization width used was 0.25 kPa and 1 rad/s along the pressure and engine speed axes respectively. This was chosen to mimic closely the architecture of the CMAC network employed by Shiriasi \textit{et al.} \cite{18}. The generalization parameter, \( C \), was chosen to be 36.

##### 4.1.2 RBF networks

Two RBF networks were used in the simulations. The first was subject to weight-only updates (this network is referred to as RBF1), while the other had all the parameters updated according to a relaxation algorithm (referred to as RBF2). Under the relaxation algorithm, during one iteration of the updating procedure the weights were updated and other parameter updates were temporarily ‘relaxed’, while on the next iteration the weight update was ‘relaxed’ and the centres and \( \sigma \) values were updated. The relaxation algorithm was used to guarantee a reduction in error following the gradient descent technique outlined in Lemmas 1 to 3. However, the centre update law will result in all centres moving closer to the input point. This will improve steady state operation where the most time is spent at the cost of reducing performance during transients, which is not desired. To avoid this occurrence, the relaxation algorithm was only switched on during throttle changes, and weight-only updates were performed at all other times.

##### 4.1.3 Modification to RBF output calculation

To avoid costly computational delays in calculating the output of the RBF network, the concept of contributing
nodes was borrowed from the CMAC architecture. In short, only those nodes within a certain distance of the input had their outputs calculated, and these were used in the weighted sum for the network output. Given that the network output is relatively smooth, it is clear that those nodes outside a certain Euclidean distance from the input will have negligible influence on the overall output, so this method was adopted with the arbitrary choice of 64 centres used. Thus, the outputs of 64 centres needed to be calculated at each iteration, rather than the outputs of the entire set of centres.

4.1.4 RBF centre positions

Since a highly accurate map was sought over all the parameter space, it was decided to arrange the centres in a grid spanning the entire space initially. This was done so as to provide a uniformly accurate mapping throughout the parameter space for RBF1.

4.1.5 RBF Network size

The size of the RBF networks was chosen to be almost 10 times smaller than the corresponding CMAC network. This meant that the centre separation for the manifold pressure and engine speed axes was 1 kPa and 1.5 rad/s respectively. The purpose was to demonstrate that significantly better results could be obtained using an RBF network one order of magnitude smaller than a CMAC.

4.1.6 Deadzone

To reduce sensitivity to engine noise and to avoid over-training the networks, a deadzone of size $d_0$ is used. A simple deadzone was applied to the error term, resulting in another term, $s_d$, where

$$
s_d = \begin{cases} 
\text{error} - d_0 & \text{for } y - y_{\text{des}} > d_0 \\
0 & \text{for } |y - y_{\text{des}}| < d_0 \\
\text{error} + d_0 & \text{for } y - y_{\text{des}} < -d_0 
\end{cases}
$$

(32)

For the simulations, the deadzone width used was 0.025 air–fuel ratios, and this was increased to 0.05 air–fuel ratios for the experimental results to allow for the effects of measurement noise.

4.1.7 Choice of $\sigma$

A heuristic choice of 2.0 (relative to centre separation) was used for RBF1 and as the starting point for RBF2. This value was chosen to ensure significant overlap between adjacent centres.

4.1.8 Normalization of separation

The separation of the centres was normalized to one in order to weight both parameters equally. Hence, instead of calculating the actual distance in parameter space, the distance metric used in calculation of the basis function outputs was

$$D(P_{\text{man}}, \omega) = \left(\frac{P_{\text{man}} - c_p}{P_{\text{sep}}}\right)^2 + \left(\frac{\omega - c_\omega}{\omega_{\text{sep}}}\right)^2
$$

(33)

where $(P_{\text{man}}, \omega)$ is the two-dimensional input to the network, $c_p$ and $c_\omega$ are the $P_{\text{man}}$ and $\omega$ coordinates of the centre and $P_{\text{sep}}$ and $\omega_{\text{sep}}$ are the separations of centres along $P_{\text{man}}$ and $\omega$ axes respectively.

4.2 Simulation results

After approximately 10 min of training, the CMAC and RBF network controllers were implemented on to the engine simulation program based on the mean value engine model. A noise-free environment was initially used in order to demonstrate the tracking capabilities of the networks.

The neural networks could now be tested on an example throttle scenario. A throttle change of 5° was presented every 10 s, with the change occurring over a period of 100 ms, as demonstrated in Fig. 4. The simulated performance of each of the networks can be seen in Figs 7 to 9.

The performance of the networks under this scenario in the steady state regions is quite good. For both RBF networks, the air–fuel ratio is either constant within the desired stoichiometric band or oscillates within the deadzone region where no updating takes place. The oscillation is caused by slight movements in one or both of the input parameters, causing small differences in the outputs of the radial functions which are not corrected until the air–fuel ratio reaches the deadzone boundary. Because of the binary nature of the CMAC regions, the network is more resistant to these slight variations in inputs, and so does not show this oscillation, instead holding a constant value somewhere within the deadzone. As an aside, the small oscillations observed with the RBF networks serve to increase catalytic converter efficiency by using the lean deviations to store oxygen, which can then be used to neutralize the hydrocarbons on a rich deviation [19]. The magnitude of the oscillations about the stoichiometric level is determined by the width of the deadzone.

During transient operation, the performance of the networks does differ. While some deviation on a throttle change may be unavoidable, it can be seen from Figs 5 to 9 that the magnitude of the deviations for the RBF2 network and the CMAC network are significantly larger than for the RBF1 network. This can be stated quantitatively by measuring the standard deviation in the air–fuel ratio for each type of controller, as shown in Table 1. The performance difference between the two RBF networks stems from the result that continuous updating of the weights is taking place during the throttle change, as the network tracks the desired output after the transport delay has expired. The resolution of the RBF1 net-
work is sufficient for the benefit of moving the centres to be outweighed by the cost of not updating the weights. In the event that far fewer centres are included in the network architecture, it is expected that the performance of the two RBF networks will be different.

The CMAC performance is not quite as good because the function approximation technique is not as smooth when the inputs are rapidly changing. The CMAC performance could be improved by reducing the generalization factor or increasing the number of regions in the network. However, the former approach will add significantly to learning time, and the latter would also benefit the RBF network performance as a more accurate function approximation of $\tilde{\hat{m}}_{ao}$ could be made. However, the purpose of this paper is to demonstrate that an RBF network with significantly fewer nodes can
produce better performance, and this result has been demonstrated.

After approximately 30 min of training under RBF1 control, the surface map was obtained as shown in Fig. 10. While the surface obtained is not completely smooth as a result of the small number of engine operating conditions to which the network has been exposed, it demonstrates an overall learning capacity as desired for a fuel injection application.

In order to demonstrate the adaptation capabilities of the RBF network, a parameter in the engine model should be subjected to a sharp change and the output response observed. In the MVEM, the volumetric efficiency, \( \eta_{\text{vol}} \), was modelled according to (4). Changes in volumetric efficiency alter the air system in the MVEM according to (3). This in turn affects the engine speed and manifold pressure. Figure 11 illustrates the adaptation control capability of the RBF control, as a result
Fig. 8  Air–fuel ratio for a $5^\circ$ tip-in tip-out scenario using RBF2

Fig. 9  Air–fuel ratio for a $5^\circ$ tip-in tip-out scenario using the CMAC controller

Table 1  Comparison of CMAC and RBF controllers

<table>
<thead>
<tr>
<th>Type of controller</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBF1</td>
<td>0.0704</td>
</tr>
<tr>
<td>RBF2</td>
<td>0.1094</td>
</tr>
<tr>
<td>CMAC</td>
<td>0.0963</td>
</tr>
</tbody>
</table>

of a variation caused by simulating a 7.5 per cent decrease in the volumetric efficiency, $\eta_{vol}$, at $t = 5$ s. The overall effect is to simulate a sudden occurrence such as a blocked air filter if the automobile has travelled off road. It is clear that on-line learning is achieved, and the controller maintains the desired AFR shortly after the model has been suddenly changed.
5 ENGINE RESULTS

The proposed controller was evaluated on a dynamometer test rig at the Mechanical Engineering Department at the University of Melbourne, and compared with a Motec production controller. The engine used was a four-cylinder, 2.4 l Mitsubishi Magna TE Series SEFI engine with no exhaust gas recirculation (EGR). The proposed RBF controller was implemented in Labview on a Pentium PC equipped with a National Instruments Data Acquisition (NIDAQ) PC-LPM 16 card to interface the PC. The Motec fuel pulse was used as a trigger for the fuel injection driver circuitry. A Bosch UEGO sensor was positioned in the exhaust, approximately 1 m from the exhaust valves. A Motec linearizing kit was used to convert the sensor output into a d.c. voltage linearly related to AFR in the range 0–1.6 V. The throttle was changed by hand for about 5° over a period of approximately 0.2 s.

The network was trained using the dynamometer to control speed and adjusting the throttle to vary pressure for the operating range of MAP and speed. The dead-zone used with the network was increased to 0.05, to account for the extra noise present on the engine. The
throttle change and corresponding MAP change are shown in Figs 12 and 13. The air–fuel ratio for the RBF
network controlled system is shown in Fig. 14.

From this figure it is clear that there are significant deviations on a throttle change. The reason for these
deviations lies in the structure of the controller design. In simulation, the effect of the fuel puddle is implicitly
contained in the neural network mapping; i.e. a throttle tip in will result in an increase in pressure followed
shortly afterwards by an increase in speed, so extra fuel will be injected when high pressure and low engine speed
combinations are observed. However, on the dynamometer, engine speed is held roughly constant and
pressure may be varied so the implicit fuel puddle information is lost. If an electronically controlled throttle
were used, the same throttle scenario could be presented repeatedly and the controller would adapt so as to elim-
inate deviations. However, this represents an academic exercise and is not a general solution, as cruise control
on a production automobile represents a similar prob-

**Fig. 12** Throttle scenario presented

![Throttle Angle (degrees) vs. Time (seconds)](chart12.png)

**Fig. 13** Manifold pressure

![MAP (kPa) vs. Time (seconds)](chart13.png)
lem. A simple solution is to use the rate of change in throttle angle as an input to the controller as well in order explicitly to acknowledge the presence of a throttle change. The experiment was repeated with this modification, and the air–fuel ratio was compared with that for the Motec controlled engine under a similar throttle scenario. Because the throttle is hand controlled, the repeatability of the experiment is not perfect and the throttle changes may be slightly different, so the throttle scenarios for each controller are shown in Figs 15 and 16. The resulting air–fuel ratios are given in Figs 17 and 18.

As can be seen from these figures, the performance of the modified RBF system is far superior to that of the production ECM. Even prior to the modification it can be seen that, although the RBF network experiences similar size deviations to the production ECM, it returns to the desired range significantly faster. The difference in emissions performance is quantified in Table 2 by examining three important statistics from the data.
Firstly, the maximum rich and lean deviations will give some idea as to how well the controller copes with the transient throttle conditions. The r.m.s. error will then give a numerical idea of how close to stoichiometry the controller is able to maintain the air–fuel ratio. This latter index will demonstrate how well the controller operates during steady state throttle scenarios, as well as how quickly the air–fuel ratio is returned to stoichiometry after a deviation. Because a deadzone is used in the RBF learning, the r.m.s. error was modified so that errors of less than 0.05 were set to zero in the calculation.

However, there is one important point to note regarding the modified system. It was necessary to use the derivative of throttle change as a modification to the proposed control law. This means that the original RBF network is estimating air flow rather than the quasi-air flow defined in equation (23), and so the performance may be further improved by treating the fuel puddle...
Table 2  Implementation statistics for Motec and RBF controllers under 5° throttle changes

<table>
<thead>
<tr>
<th></th>
<th>Motec control scheme</th>
<th>RBF control scheme</th>
<th>Modified RBF control scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum deviation</td>
<td>0.97</td>
<td>0.93</td>
<td>0.38</td>
</tr>
<tr>
<td>above stoichiometry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum deviation</td>
<td>1.18</td>
<td>1.23</td>
<td>0.69</td>
</tr>
<tr>
<td>below stoichiometry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modified r.m.s. error</td>
<td>0.243</td>
<td>0.347</td>
<td>0.157</td>
</tr>
</tbody>
</table>

being compared. The performance of the RBF network controller with weight-only updates was found to be superior to that of a CMAC network with much greater resolution. The RBF controller was also shown to be capable of adapting on-line to significant parameter changes in the engine systems.

Implementation results using an RBF network show significant promise for use of this type of network, although better performance may be achieved if it is used in conjunction with other control strategies. The look-up table approach used by many production ECUs is very similar to the operation of the CMAC network, and the RBF has shown better mapping capability. Thus, one distinct possibility is to use an RBF network in place of the look-up tables currently used in fuel injection systems to estimate air flow into the cylinder.

6 CONCLUSION

An RBF network was shown via simulation to be capable of controlling the fuel injection system in an automobile. No a priori knowledge is required about the internal transfer functions of the engine, and thus the success of the controller is not dependent upon the accuracy of the models used, although better transient performance could be achieved if some knowledge of the fuel pooling process were included in the controller. The controller may be updated on-line, which is highly desirable as changing engine conditions owing to factors such as engine wear are compensated for and their effects upon performance are eliminated.

A comparison between two types of RBF controller and the CMAC controller previously used in fuel injection applications was carried out. The RBF controllers differed in the number of parameters that were updated, with a weight-only update and all-parameter update explicitly, i.e. using a conventional control strategy in conjunction with the RBF network.

REFERENCES

1 Draft air quality improvement plan. In EPA Publication 707, 2000 (Environmental Protection Authority, Port Phillip Region).


