

Chapter 1

Artificial Intelligence for Modeling and Control of Nonlinear Phenomena in Internal Combustion Engines

Artificial intelligence techniques allow to solve highly nonlinear problems offering an alternative way to deal with complex and dynamic systems with good flexibility and generalization capability. They are widely used in several areas ranging from power system modeling and control to medicine and social sciences. Because of their good ability to model nonlinear phenomena together with their relatively simple application procedure, artificial intelligence systems have found an increasing usage in the modeling, diagnosis, and control of internal combustion engines. The most used techniques include Artificial Neural Networks (ANNs), Genetic Algorithms, Expert Systems, fuzzy logic and hybrid systems, with several combinations of two or more of these.

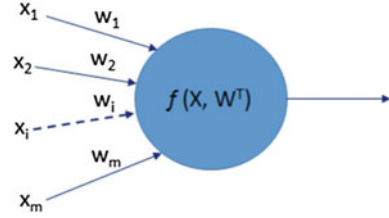
The present chapter aims to describe the use of artificial intelligence in some engine applications where the inherent nonlinear nature of the process dynamics requires alternative approaches to guarantee a more accurate control action. A special focus will be kept on the use of Artificial Neural Networks and fuzzy logic techniques.

1.1 Neural Networks Architectures for Engine Applications

Neural networks provide a wide range of functions that can be used in the field of engine control. They can be used for example to train *black box* process models of various engine subsystems with few a priori theoretical knowledge. In this way, difficulties that appear when applying classical techniques on complex nonlinear systems are suppressed.

A basic characteristic of neural networks is that of emulating the structure of human brain and, in particular, its ability to learn from experience without actually modeling the physical and chemical laws that govern the system [1–3]. In general terms, an Artificial Neural Network is a computational system able to store and

Fig. 1.1 Example of a neuron model



utilize knowledge acquired through experimenting [4] and it can be defined as an interconnected assembly of simple computational elements called *neurons* or *nodes*. Each element receives the inputs from neighboring nodes, sums all these contributions, and produces an output that is a nonlinear function of the sum. The signals flowing on every connection are appropriately scaled by programmable parameters called *weights*. In Fig. 1.1, an example of neuron model is depicted. In Fig. 1.1, x_i represents the i th input (i.e. the i th component of the vector X), w_i is the relative weight at the i th input (i th component of the vector W), and $f(W^T, X)$ is a function, known as *activation function* whose value, calculated as weighted sum of the inputs, represents the neuron outputs.

An important part of the modeling with neural networks is the so-called training of the network (*learning procedure*). This latter is the process that, using different learning algorithms chosen according to the network structure and type, assigns values to the network parameters (in particular, to the connection weights) in order to minimize the error between the outputs of the neural model and the correct or desired outputs.

The most popular learning algorithms are the *back-propagation* and its variants, which are generally applied to multi-layer feedforward networks having differentiable activation functions. The back-propagation algorithm is an iterative algorithm that updates the values of the network interconnections such that a total square error is optimized on a set of input/output data. The error can be expressed by:

$$E = \frac{1}{2} \sum_p \sum_i (t_{i,p} - y_{i,p})^2 = \frac{1}{2} \sum_p E_p, \quad (1.1)$$

where p is the number of data supplied during the learning, i is the number of outputs (i.e. of neurons in the last layer), t is the desired output value, and y is the corresponding value calculated by the network. Initially, the training is performed by assigning random values to the weights $w_{i,j}$. With every iteration, one of the training set samples is provided to the neural network and the error committed by each neuron output is calculated. Then, the gradient algorithm for back-propagation of the output error is applied backward through the network updating the value of the weights, according to the formula:

$$\Delta W_{ij}(t) = -\varepsilon \frac{\partial E_p}{\partial W_{ij}}, \quad (1.2)$$

where ε is a parameter chosen by the user and called speed of the learning.

The training of all the patterns of a training data set is called an *epoch*. The learning procedure uses a number of epochs that allows to obtain a sufficiently low error or an error that no longer decreases. This latter case indicates the incapacity of the network to solve the problem.

The learning procedure must be carefully designed. The training data should cover all nonlinearities and should contain information spread evenly over the entire range of the system. This allows to avoid significant model failure if the neural network model is used in a region where an insufficient amount of data is supplied. Moreover, for a good predictive ability of an ANN, it is important that the training and the validation are done using experimental and independent data.

An important basic operation that has to be followed to successfully handle a problem with Artificial Neural Networks is the selection of a suitable network architecture, which is a choice that mainly depends on the problem.

One of the most frequently employed neural architectures, also in engine applications, is the multi-layer perceptron (MLP). MLP networks consist of successive layers of adaptive weights, with a layer of input neurons, some internal layers (also called *hidden layers*), and a layer of output neurons. The neurons of each layer have connections that run from every unit in one layer to those in the next layers, creating a *feedforward architecture* with no feedback loops (see Fig. 1.2). Therefore, an MLP structure is characterized by the number of neurons of each layer and the number of hidden layers. The number of hidden layer can be increased depending on the problem; however, the most frequent configuration is that with one internal layer only, which is suitable for the majority of the engine modeling and control problems to be handled. On the other hand, big growing networks can be ill-posed for overtraining and be difficult to implement in real time.

Recurrent Neural Networks (RNN) are neural networks with one or more global feedback loops, usually with a unit time delay (often denoted by z^{-1}). The presence of feedback loops introduces a dynamic effect in the computational system and makes them suitable for *black box* nonlinear dynamic modeling and for *input-output mapping*. This feature is of a particular interest in engine applications.

Fig. 1.2 Structure of an MLP neural network model

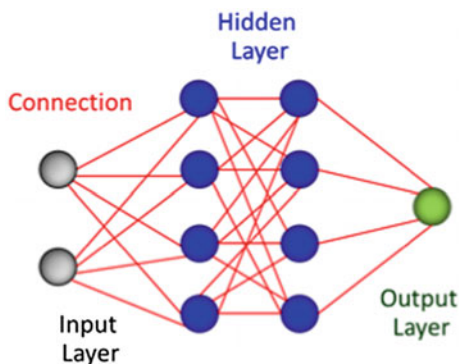
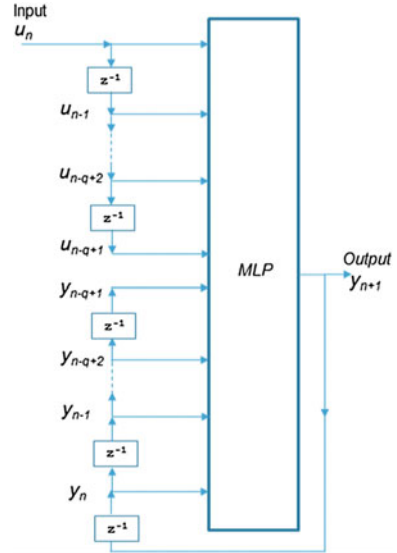


Fig. 1.3 General structure of a NARX neural network model



The architectural layout of a recurrent network can take many different forms mainly dependent on the feedback typology. In nonlinear auto regressive with exogenous input (NARX) models, the input of a generic multi-layer perceptron is applied to a tapped delay-line memory of q units, whereas the output is fed back to the input via another tapped delay-line memory, also of q units. The contents of these two tapped-delay-line memories are used to feed the input layer of the multi-layer perceptron [3]. A typical structure of NARX model is shown in Fig. 1.3.

In Fig. 1.3, the present value of the model input is denoted by $u(n)$ and the corresponding value of the output is denoted by $y(n + 1)$, which is ahead of the input by one time unit. The formulation of the NARX model can be described as:

$$y(n + 1) = F[y(n), \dots, y(n - q); u(n), \dots, u(n - m + 1)], \quad (1.3)$$

where q is the number of past output terms used to predict the current output, m is the number of input terms used to predict the current output, and F is a nonlinear function of *regressors* that are transformations of past inputs and past outputs. Therefore, NARX neural network structure allows the user to define how many previous output and input time steps are required for representing the system dynamics best. This feature can be used in all engine applications where input and output variables show related dynamics, such as in the case of EGR systems or in turbocharged engines where turbine and compressor have a strict connection of process dynamics. However, many other applications, such as Air-Fuel Ratio (AFR) or exhaust emissions prediction, can be effectively modeled and controlled using a recurrent network architecture.

A learning algorithm used for recurrent neural networks is the back-propagation through time algorithm (BPTT), which is a modification of the back-propagation

learning algorithm used for feedforward neural networks. In BPTT training, the network weights are adjusted also on the basis of the network state at previous time steps. For a recurrent neural network trained for a time interval ranging from t_1 to t_n , the total cost function $E(t)$ can be represented as:

$$E(t) = \sum_{t=t_1}^{t_n} E_p(t), \quad (1.4)$$

where the $E(t)$ is the sum of the errors $E_p(t)$ calculated at each time step, and the network weights are adjusted on the basis of the equation:

$$\Delta W_{ij}(t) = -\varepsilon \frac{\partial E(t)}{\partial W_{ij}} = -\varepsilon \sum_{t=t_1}^{t_n} \frac{\partial E_p(t)}{\partial W_{ij}}. \quad (1.5)$$

1.2 Use of ANNs for Modeling and Control of Internal Combustion Engines

The ANNs have been applied to predict the performance of various thermal systems. Their use for modeling the operation of internal combustion engines is more recent. Typically, a neural approach is used to predict the performance and exhaust emissions as well as the specific fuel consumption and fuel–air equivalence ratio of both gasoline and diesel engines [5–9]. For spark ignition engines, ANNs were originally applied to predict the effects of valve-timing on the engine performance and fuel economy [10]. The use of ANN was also proposed to determine torque, brake specific fuel consumption, and emissions in engines using alternative fuels, such as different gasoline–ethanol blends and diesel–biofuel blends [11–13].

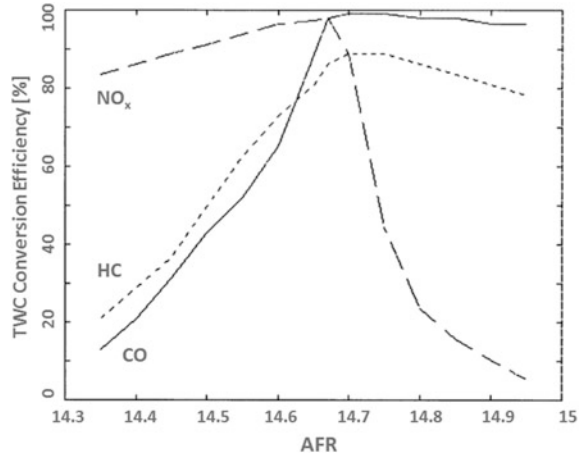
In the following paragraphs, some examples concerning the use of neural networks in the modeling and control of some specific engine applications will be presented.

1.2.1 Air–Fuel Ratio Prediction and Control

In order to achieve the optimal functioning of a three-way catalytic converter (TWC), i.e., its maximum efficiency, a spark ignition engine has to operate within a narrow band around the stoichiometric air–fuel ratio (14.7:1), with mean deviations that cannot exceed 0.1% (see Fig. 1.4).

In current technology for gasoline engines, AFR control currently relies on a mean value engine model (MVEM) representation [14, 15]. Such a controller estimates in a feedforward way the actual airflow rate in the cylinder and provides the correspondent amount of fuel to be delivered in the next engine cycle.

Fig. 1.4 Efficiency of a TWC as a function of air–fuel ratio



Moreover, the signal of an oxygen sensor placed at the exhaust is used as a feedback control signal to correct the previously calculated mass of fuel to be injected, ensuring a steady state stoichiometry of the mixture.

The fast (but approximate) feedforward component is important to handle transient operating conditions and to compensate the slow dynamics of the feedback loop. There are in fact three time delays included in the value of air–fuel ratio (AFR_o) measured by means of the oxygen sensor: injection delay, combustion delay, and transport delays from the exhaust valve to the oxygen sensor. So:

$$AFR_o(t) = AFR_c(t - t_D), \quad (1.6)$$

where

$$t_D = t_{inj} + t_{comb} + t_{trans}.$$

The transport delay t_{trans} depends on factors such as engine speed, exhaust air mass flow rate, exhaust manifold geometry, etc. [16].

Mean value models for the prediction of airflow rate in the cylinder have some significant limitations, such as the high experimental burden requested for parameters identification and the intrinsic non-adaptive features. To overcome this latter problem, adaptive methodologies (based, for example, on observers, sliding mode controllers, or Kalman filters) have been proposed in order to estimate the states and tune the parameters.

Traditional mean value models for AFR prediction have also to include compensation terms for fuel path dynamics and *wall-wetting* phenomena. In fact, the liquid fuel injected into the intake port only partially enters the cylinder in the current engine cycle. Some of it is collected in fuel films on the walls of intake manifold and close to the back face of the intake valves. The fuel, then, partially evaporates later from these films. A model for this phenomenon was proposed by Aquino [14].

The fuel film dynamics model is highly nonlinear: the fuel fraction and the delay strongly depend nonlinearly on several engine variables (e.g., load, speed, and temperature). Only a good knowledge of these parameters can assure achievement of significant range compensation and hence effective transient control.

The residual gas fraction dynamics together with the mixing dynamics are also highly nonlinear phenomena that have to be modeled for a correct prediction of AFR.

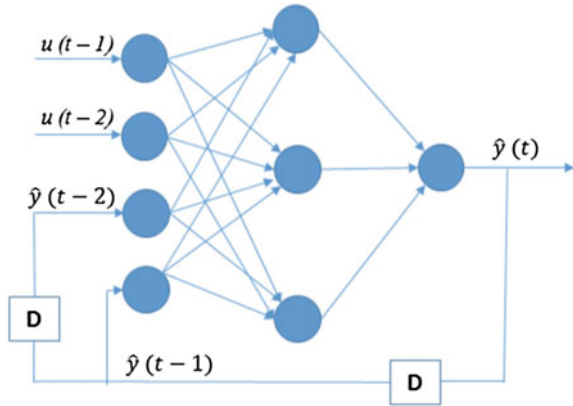
Artificial Neural Networks, which are a powerful tool for modeling highly nonlinear and dynamic systems, can be considered a good candidate for AFR process modeling or for the realization of *virtual* AFR sensors.

Several approaches have been proposed for AFR modeling and control using neural networks. Input parameters to the neural models are generally variables such as engine angular speed, throttle valve opening, absolute manifold pressure, fuel injection time, etc. In [17], AFR was estimated on the basis of spark plug voltage waveform. This latter signal, in fact, is considered to be influenced by the combustion inside the cylinder, which in turn depends, among other factors, on the value of air–fuel ratio. The neural network architecture chosen in this case was a multi-layer perceptron with a cumulative back-propagation learning algorithm. Raw data were firstly pre-processed in order to achieve a satisfactory convergence of the network. To this aim, data points corresponding to regions of the spark signal known to contain poor information about AFR were removed. Moreover, in order to enhance the SNR and reduce the effect of cyclic variation on the spark signal, a filtering was applied to the input data. With regard to the optimal number of training data (input–output vectors pairs), it was stated that a number of training vectors comparable with the number of weights in the net leads to a good generalization capability of the neural network.

In [18], the application of a multiple MLP architecture for adaptive air–fuel ratio (AFR) control is discussed. In this work, the slow AFR process dynamics were represented in the input parameters of the model. In particular, the variable time constant aspects of the process were considered by presenting input parameters to the ANN model as combinations of delayed and filtered sample data values. The filter time constants were designed to span the range of corresponding variable time constants in the system. The approach used to model the variable exhaust transport delay was to configure the ANN model with multiple delayed AFR outputs spanning the full range of speed and load-dependent delays. Of course, this approach is effective if the delays are known or can be calculated. However, when the delays are unknown and variable, a different strategy is required and a further model of the delay itself has to be developed. In this case, the authors propose to train a second supervisory ANN to attribute relevancies to each of the ANN model output predictions. In particular, at high engine speeds, the supervisor attributes more relevance to the shorter time delay model predictions and vice versa.

In [19], neural networks with feedback connections in a recursive architecture were used to model and control the nonlinear air–fuel ratio process dynamics. In particular, the authors used an RNN known in the literature as nonlinear output

Fig. 1.5 General structure of an NOE model



error model (NOE) [20] with one output variable, one hidden layer, and two output delays, which had a structure like in Fig. 1.5, where the output $\hat{y}(t)$ represents the actual AFR.

The general form of the NOE model can then be written as:

$$\hat{y}(t|\theta) = F[\hat{y}(t-1|\theta), \hat{y}(t-2|\theta), \dots, \hat{y}(t-m|\theta), u(t-1), \dots, u(t-n)], \quad (1.7)$$

where θ is the adjustable parameter, $u(t)$ are reference inputs, the indices m, n define lag space dimensions of external inputs and feedback variables, and F is a nonlinear mapping function. The input variables, like in most AFR models, included the intake manifold absolute pressure (MAP), the engine speed (RPM), and the fuel injection pulse width (FPW).

The network training is performed by minimizing a cost function estimated as function of the mean squared error. As for other neural network architectures, the learning process of a RNN model has to be a compromise between precision and generalization. High generalization cannot be guaranteed if the training data set does not provide sufficient information. These latter, in the case of AFR models, have to cover most of the system operating conditions, providing at the same time a good knowledge of the dynamic behavior. For this reason, the experimental profiles of AFR models generally include, other than engine steady state operations, also sharp accelerations and decelerations. The learning process should also avoid overtraining that occurs when the minimization task includes many iterations: this leads to a better precision but to a generalization loss due to overfitting.

A neural controller of injection time based on the information coming from an AFR neural network model is proposed in [21]. Like in traditional controls, neural controllers require an identification phase of the process to be controlled that is known as the *learning control* phase. In case of AFR control, the learning control phase might use the AFR estimated by a neural model, using the stoichiometric AFR value and the AFR value measured as output of the engine process by means of a dedicated sensor.

In order to take into account also process aging effects and to tune the control to the specific engine, the controller configuration can also be done online (online training) during the normal functioning of the engine, even if this could require a high computational effort to the ECU. For this reason, an extremely simple, from a computational point of view, learning algorithm is requested.

Once the controller has been configured, it can commute to the normal operation modality, in which it receives the AFR value measured by the oxygen gas sensor and provides the control variable *injection time*.

The AFR neural controller proposed in [21] had as inputs the engine angular speed, the intake manifold absolute pressure and the angle of throttle opening. A one-step delay was also introduced to take into account all the delays in the measured value of AFR.

In [22], the signal coming from an in-cylinder pressure sensor is used as an input for an injection time control architecture based on a neural AFR virtual sensor. The feedforward part of the control system relies on engine control maps, whereas the feedback part is composed of four subsystems:

- a *features selector* block, which extracts some parameters from pressure curve to be used for AFR prediction;
- a *virtual AFR sensor*, based on an MLP neural network model;
- a block that evaluates both the error and the error variation obtained by comparing the AFR value estimated by the virtual sensor and a reference AFR value (i.e., stoichiometric value);
- a *soft computing controller*, which modifies the injection pulse duration calculated by the feedforward section on the basis of the output of the above block.

The *features selector block* (see Fig. 1.6) performs a pre-processing of the in-cylinder pressure signal in order to extract the features most relevant to AFR prediction. In other words, it selects the best inputs for the virtual sensor. From an algorithmic point of view, the action of this block is based on *clustering analysis* techniques.

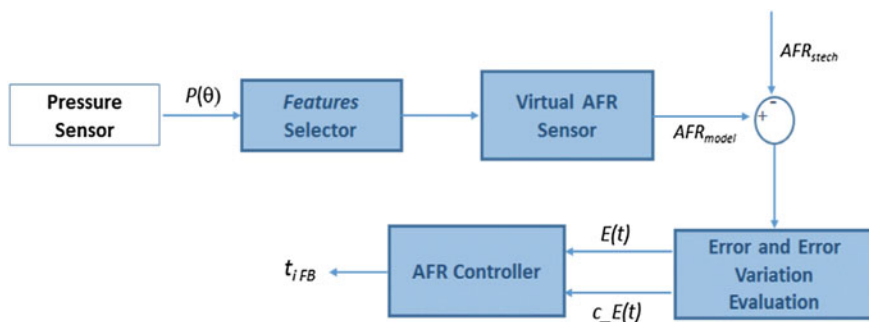


Fig. 1.6 AFR control based on the use of a neural virtual sensor

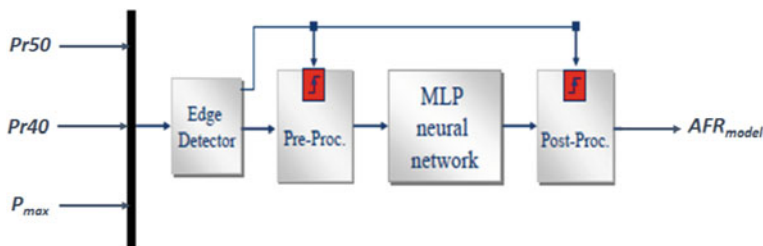


Fig. 1.7 Block scheme of a virtual AFR sensor

Moreover, the block allows to find an optimal sampling of the inputs, i.e., the number of consecutive pressure cycles that have to be averaged to improve AFR prediction.

Figure 1.7 illustrates the block scheme of the virtual AFR sensor. One of the model inputs is $Pr50$, i.e., the combustion pressure value at 50 crank angles (CA) degrees *after* combustion top dead center divided by the value that the same pressure signal assumes at 50 CA *before* top dead center ($Pr50 = P(50)/P(-50)$). Other inputs are Pr value at 40 CA ($Pr40 = P(40)/P(-40)$) and the maximum pressure value (P_{max}). These inputs have been chosen, among others, by the *features selector* subsystem.

The *Edge Detector* in Fig. 1.7 is devoted to the synchronization of the model components with the incoming data. It samples the data and provides an enabling signal to the following blocks. The *Pre-Proc* block performs a pre-processing of the incoming data in order to reduce the cycle-by-cycle variation of the input data. To this aim, it carries out a moving average of the input data. The user can modify the number of averaged cycles but, in this case, the optimal value has been chosen by the feature selector.

The heart of the virtual AFR sensor is represented by a multi-layer perceptron neural network. All the endogenous parameters of the neural network, such as the regularization parameter, the number of hidden layers, the number of neurons for each hidden layer, the type of activation function for the neurons, etc., have been set in order to maximize the “generalized forecast capability” of the learning machine. To this aim, a modified version of the *ordinary cross-validation estimate* of the endogenous parameters has been used as fitness function.

The searching of the minimum of this function has been performed by using a stochastic searching algorithm known as particle swarm optimization algorithm (PSOA) (see [23]).

Error and error variation between output of virtual AFR sensor (AFR_{model}) and reference AFR (AFR_{stech}) are calculated at time t in the subsystem shown in Fig. 1.8. To this aim, input signals are suitably dealt with sum blocks and a time delay:

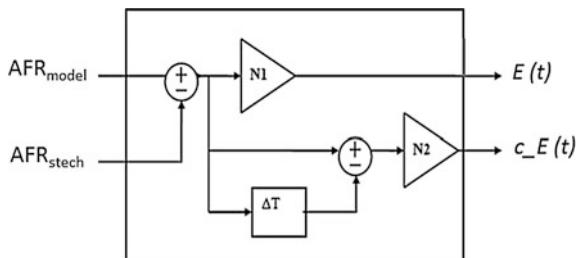


Fig. 1.8 Subsystem for calculation of $E(t)$ and $c_E(t)$

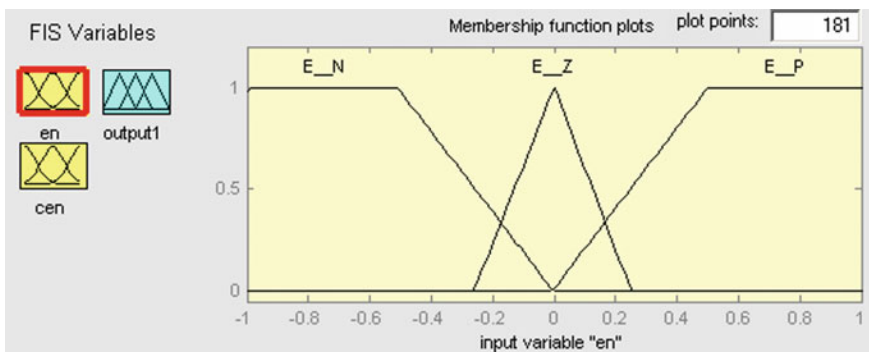


Fig. 1.9 Membership functions of the variable $E(t)$

$$E(t) = N_1 * (AFR_{stech} - AFR_{model})$$

$$c_E(t) = N_2 * (E(t) - E(t - \Delta T));$$

where N_1 and N_2 are two normalization factors.

The variables $E(t)$ and $c_E(t)$ represent the input to the AFR control system. AFR soft computing control was implemented by means of fuzzy techniques. In particular, the following fuzzy sets of the input variable $E(t)$ were considered:

- E_P “positive error,”
- E_N “negative error,”
- E_Z “zero error,”

with the corresponding membership functions shown in Fig. 1.9.

The following fuzzy sets have been considered for the variable $c_E(t)$:

- CE_P “variation of the positive error”;
- CE_N “variation of the negative error”;
- CE_Z “variation of the zero error”;

with the corresponding membership functions depicted in Fig. 1.10.

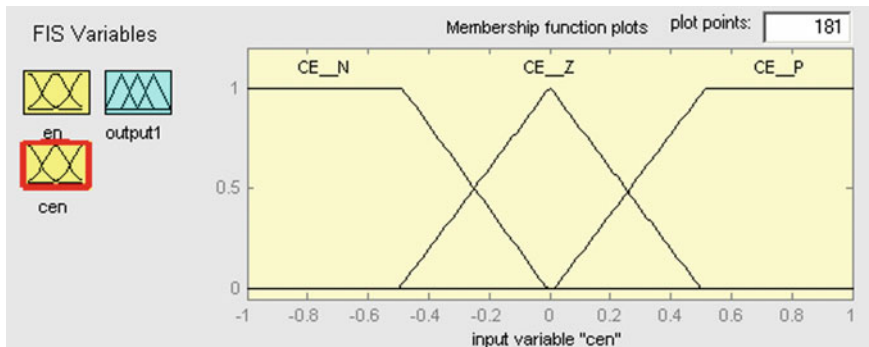


Fig. 1.10 Membership functions of the variable $C_E(t)$

The fuzzy controller is a fuzzy system having as antecedents the fuzzy sets of $E(t)$ (error) and $c_E(t)$ (error variation), and as consequents the fuzzy sets of the output variable Δ_{DI} (injection time).

- I_N “positive injection,”
- I_P “negative injection,”
- I_Z “zero injection,”

whose membership functions are shown in Fig. 1.11.

The fuzzy algorithm conceived for the controller was composed by nine rules having the following traditional form:

- IF E_P AND CE_P THEN output IS I_P
- IF E_N AND CE_N THEN output IS I_N
- IF E_P AND CE_Z THEN output IS I_P
- IF E_N AND CE_Z THEN output IS I_N
- IF E_P AND CE_N THEN output IS I_Z
- IF E_N AND CE_P THEN output IS I_Z

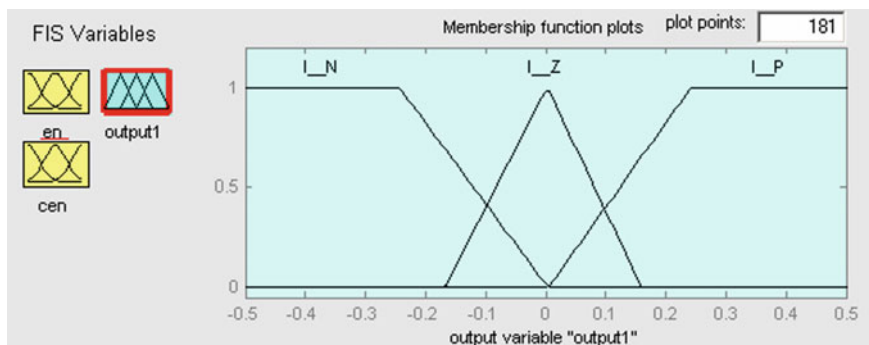
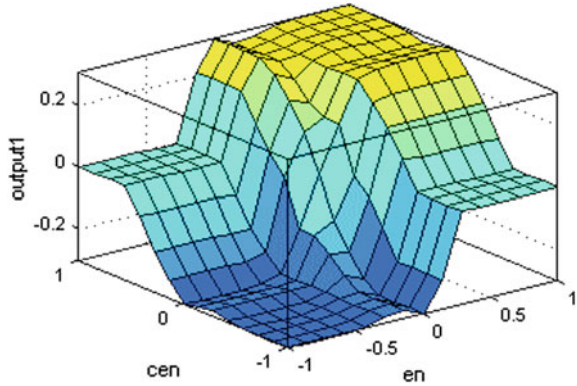


Fig. 1.11 Membership functions of the output variable Δ_{DI}

Fig. 1.12 Tridimensional map of AFR fuzzy controller



- *IF E_Z AND CE_Z THEN output IS I_Z*
- *IF E_Z AND CE_N THEN output IS I_N*
- *IF E_Z AND CE_P THEN output IS I_P*

The aim of the fuzzy algorithm described by the previous rules is to provide for each value of input variables a correspondent value of the control variable Δ_{DI} (injection time). Figure 1.12 illustrates the tridimensional map summarizing the way of functioning of the controller.

Finally, to estimate the update of the control variable induced by the feedback part of the air–fuel control system the output of the controller is processed in the following way:

$$t_{iFB} = N_3 * \Delta_{DI}(t) + \Delta_{DI}(t - \Delta T),$$

where N_3 and ΔT are two factors to be tuned.

The proposed approach allowed to obtain a strict control of AFR with an error, with an error below of 1% both in steady state and during fast transients.

1.2.2 Use of Neural Networks to Predict Combustion Pressure Parameters

Real-time combustion process monitoring in internal combustion engines may provide a strong tool regarding engine operation and may be profitably used for closed-loop electronic engine controls that allow internal combustion engines to comply with the severe normative on pollutants emission and fuel consumption. One of the most important parameters used for the evaluation of the combustion quality is the in-cylinder pressure. However, this kind of measure requires an intrusive approach to the cylinder and a special mounting process. Moreover, the combustion pressure transducers used for this kind of applications still have a high

cost for mass production automotive engines and still remains some problems of robustness and performances. Due to the disadvantage of direct pressure measurement, several non-intrusive techniques have been proposed to reconstruct the cylinder pressure and obtain information about the combustion quality [24–26]. The non-intrusive diagnostics offer several advantages: the sensors are generally placed externally the engine and no engine structural modifications are required. Moreover, the non-intrusive sensors are quite cheap, as they are not requested to resist very high pressures and temperatures.

Artificial Neural Networks can have a role in estimating several combustion pressure parameters on the basis of information coming from sensors already present on engine. An example of this is combustion pressure monitoring using the *engine crankshaft speed*.

Cylinder pressure reconstruction on the basis of instantaneous crankshaft speed is considered to be a successful approach both for its simplicity and its low cost [27–32]. Crankshaft angular speed, in fact, contains information to reconstruct cylinder pressure, and hence, it can be chosen as an input to a neural network model for cylinder pressure estimation. In particular, the following correlation can be written:

$$\dot{\omega} = \frac{1}{J(\theta)} \left[-\frac{1}{2} \frac{dJ(\theta)}{d\theta} \omega^2 + T_{\text{ind}} - T_{\text{fric}} - T_{\text{load}} \right], \quad (1.8)$$

where ω is the crankshaft velocity, $J(\theta)$ is the moment of inertia of the engine, T_{ind} , T_{fric} , and T_{load} are the indicated torque, frictional torque, and load torque, respectively. The indicated torque is strictly correlated to the in-cylinder pressure $P(\theta)$, and for a single cylinder engine, it is given by:

$$T_{\text{ind}} = (P(\theta) - P_{\text{man}}) A_p \frac{ds(\theta)}{d\theta}, \quad (1.9)$$

where A_p is the piston head surface area, P_{man} is the intake manifold pressure, and $s(\theta)$ is the piston stroke from top dead center (TDC). Therefore, Eq. (1.8) can be re-written as:

$$\dot{\omega} = f(\theta, P, \omega) = \frac{1}{J(\theta)} \left[-\frac{1}{2} \frac{dJ(\theta)}{d\theta} \omega^2 + (P(\theta) - P_{\text{man}}) A_p \frac{ds(\theta)}{d\theta} - T_{\text{fric}} - T_{\text{load}} \right]. \quad (1.10)$$

Equation (1.10) illustrates clearly the nonlinear correlation existing between engine crankshaft speed and in-cylinder pressure.

In [31], cylinder pressure reconstruction was carried out using a RNN with instantaneous crankshaft speed fluctuations and motored pressure as input signals. In addition, spark advance was chosen as another input, since the values of peak pressure and its location and the rate of pressure rise are significantly affected by the ignition time value. Equivalence ratio also has a significant influence on the

cylinder pressure peak, hence it was chosen as one of the inputs. On the basis of these inputs, the regression vector for neural network was given by:

$$P(t|w) = NN(P(t-1), \dots, P(t-a), \omega(t-1), \dots, \omega(t-b), P_{\text{mot}}(t-1), P_{\text{mot}}(t-c), SA(t-1), \varnothing(t-1), w), \quad (1.11)$$

where a , b , and c are the number of past cylinder pressure output, the number of past speed, and past motored pressure data given to the neural network as feedback variables.

The authors demonstrated that the minimum mean square error in pressure prediction was obtained for $a = 3$, $b = 7$, $c = 7$, and with 13 neurons in the hidden layer. By using the above-described RNN model, the residuals between the experimental pressure curve and the simulated one lay in the range ± 0.5 bar for all the investigated operating conditions.

In [32], a radial basis function (RBF) neural network model having as input the instantaneous angular speed was used for pressure waveform reconstruction. Using an RBF, the correlation between cylinder pressure P and engine angular speed is given by:

$$P_{kj}(\theta) = \sum_{i=1}^n h_{ki}(\theta) w_{ij}.$$

Therefore, the pressure can be expressed as a linear combination of the output of the hidden neurons, that is, as a linear combination of a set of n fixed basis functions $h_{ki}(\cdot)$; the coefficients of the linear combinations, w_{ij} , are the weights or model parameters.

The results of the proposed approach showed that the measured and estimated pressure traces matched well over all stages of the pressure process: compression, onset of combustion, peak pressure, and the rise and fall of the combustion.

The maximum deviation interval between the predicted and measured pressures resulted less than 0.46 bar. This is the same quantitative order as that of the cycle-to-cycle variation and, therefore, the prediction can be judged to be accurate enough for averaged analysis over the cyclic variation. Deviation of peak pressure was less than 3 bar, whereas the deviation of the angular location of pressure peak was ± 1 crank angle.

In [33], a neural network approach for real-time prediction of in-cylinder pressure peak value (PP) and its angular location (LPP) has been proposed. The trained network, which can be viewed as a nonparametric model of the engine process, had as inputs the engine angular crankshaft speed and the crankshaft speed derivative (i.e., crankshaft acceleration) (see Fig. 1.13).

In order to train and validate the neural network model, measurements were carried out over the engine speed range 1000–2000 rpm, with steps of 200 rpm, and absolute intake pressure values ranging from 1000 to 1600 mbar.

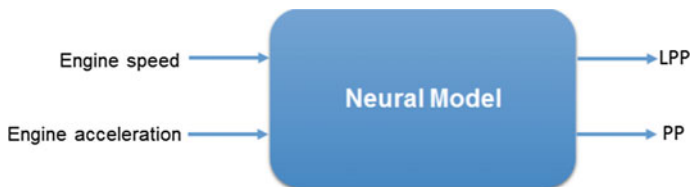


Fig. 1.13 Neural network model for prediction of LP and LPP

Table 1.1 Main features of the neural model for PP and LPP prediction

Neural network structure	feedforward MLP
Neuron model	$\tanh(x)$
# Neurons in hidden layer	30
Training algorithm	<i>Trainbr</i> with 0.3 as regularization factor

The overall data set was, then, divided into two groups. One group was used as training data set, i.e., to train the neural network model and to set the internal model parameters. The other group was used as testing data set, i.e., to validate the trained network.

As neural model, a multi-layer perceptron (i.e., MLP) neural network was chosen. The MLP had only a hidden layer with 30 neurons and *arctan* as activation function. In order to train the network, a Bayesian regularization back-propagation was used. This latter is a process that minimizes a combination of squared errors and weights and then determines the correct combination so as to produce a network that generalizes well. Use of this process guaranteed a satisfactory generalization capability of the neural network model and, at the same time, allowed to avoid overfitting issues. The tuning of internal parameters (e.g., the value of the regularization parameter, neuron biases) of the neural network model was optimized by using an evolutionary algorithm. In Table 1.1, the main features of neural network model are shown.

The overall *Relative Error* in the prediction of pressure peak (PP), obtained from scaling the root mean square error (RMSE) by the maximum of the peak pressure of the experimental data set, was 4% at 1000 rpm, 5% at 1500 rpm, and 7% at 2000 rpm. The model also showed a good capability of predicting the angular location of pressure peak (LPP) with a RMSE ranging from 1.38 to 5.2 crank angles degrees.

The results confirmed that the neural model can be effectively used for PP and LPP estimation. Moreover, the model also revealed its capability to predict pressure peak reductions due for example to inefficient combustions, misfiring events, and other combustion abnormalities. As a consequence, it can be used as a non-intrusive tool for real-time diagnosis of engine combustion quality in advanced closed-loop control systems.

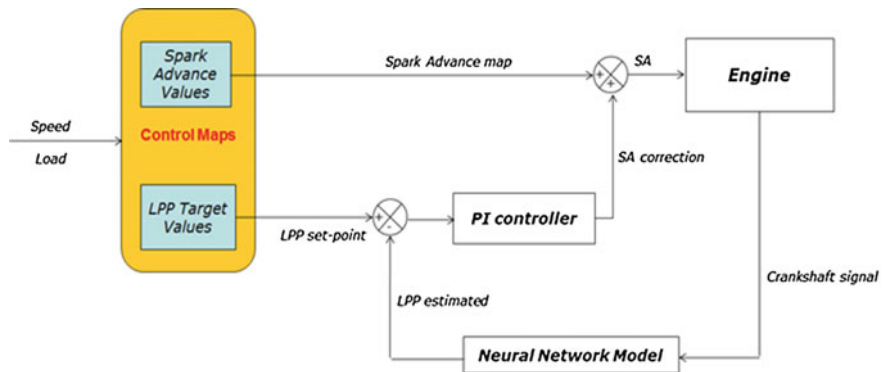


Fig. 1.14 Spark advance closed-loop control based on LPP prediction by means of a neural model

Location of pressure peak, as it is strictly connected to the ignition angle, represents a promising control variable for gasoline engine spark advance control. A possible engine controller could estimate the LPP from the crankshaft speed information and could use it as a feedback variable in an ignition timing controller. This allows to maintain the LPP close to its reference value modifying, if requested, the spark advance value stored in the engine control maps. For each engine speed and engine load, the LPP set-point can be defined as the optimal value to obtain the desired engine behavior. In high load ranges, for example, late pressure peak locations could be requested to hold down the NO_x emissions.

A block scheme of a possible engine closed-loop control is shown in Fig. 1.14. The architecture includes a proportional–integral controller for LPP control.

A similar controller was proposed in [34]. Also in this work, a spark advance control strategy based on the location of peak pressure (LPP) is presented. A feedforward MLP neural network is introduced in this study to predict LPP by using only few samples (five) extracted from in-cylinder pressure voltage signal (the entire acquisition of pressure curve was not required).

References

1. J.M. Zurada, *Introduction to Artificial Neural Systems* (West Publishing Company, St. Paul, MN, 1992)
2. F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **65**, 386–408 (1958)
3. S.S. Haykin, *Neural Networks and Learning Machines* (Prentice Hall, 1994)
4. L. Fortuna, G. Rizzotto, M. Lavorgna, G. Nunnari, G. Xibilia, R. Caponetto, *Soft Computing—New Trends and Applications* (Springer, Heidelberg, 2001)
5. S.A. Kalogirou, Application of artificial neural-networks for energy systems. *Appl. Energy* **67**, 17–35 (2000)

6. C. Sayin, H.M. Ertunc, M. Hosoz, I. Kilicaslan, M. Canakci, Performance and exhaust emissions of a gasoline engine using artificial neural network. *Appl. Therm. Eng.* **27**, 46–54 (2007)
7. E. Arcaklioglu, I. Celikten, A diesel engine's performance and exhaust emissions. *Appl. Energy* **80**, 11–22 (2005)
8. G. Najafi, B. Ghobadian, T. Yusaf, H. Rahimi, Combustion analysis of a CI engine performance using waste cooking biodiesel fuel with an artificial neural network. *Am. J. Appl. Sci.* **4**(10), 756–764 (2007)
9. N. Ladommatos, R. Balian, R. Horrocks, L. Cooper, The effect of exhaust gas recirculation on soot formation in a high-speed direct-injection diesel engine. SAE Technical Paper 960841
10. M. Golcu, Y. Sekmen, P. Erduranli, S. Salman, Artificial neural network based modelling of variable valve-timing in a spark ignition engine. *Appl. Energy* **81**, 187–197 (2005)
11. G. Najafi, B. Ghobadian, T. Tavakoli, D.R. Buttsworth, T.F. Yusaf, M. Faizollahnejad, Performance and exhaust emissions of a gasoline engine with ethanol blended gasoline fuels using artificial neural network. *Appl. Energy* **86**, 630–639 (2009)
12. B. Ghobadian, H. Rahimi, A.M. Nikbakht, G. Najafi, T.F. Yusaf, Diesel engine performance and exhaust emission analysis using waste cooking biodiesel fuel with an artificial neural network. *Renew. Energy* **34**, 976–982 (2009)
13. M.I. Canakc, A. Erdil, E. Arcaklioglu, Performance and exhaust emissions of a biodiesel engine. *Appl. Energy* **83**, 594–605 (2006)
14. C.F. Aquino, Transient A/F control characteristics of the 5 liter central fuel injection engine. SAE Paper 810494 (1981)
15. E. Hendricks, S. Sorenson, SI engine controls and mean value engine modelling. SAE Technical Paper 910258 (1991). doi:[10.4271/910258](https://doi.org/10.4271/910258)
16. A. di Gaeta, S. Santini, L. Glielmo, F. De Cristofaro et al., An algorithm for the calibration of wall-wetting model parameters. SAE Technical Paper 2003-01-1054 (2003). doi:[10.4271/2003-01-1054](https://doi.org/10.4271/2003-01-1054)
17. J. Howlett, S.D. Walters, P.A. Howson, I.A. Park, Air-fuel ratio measurement in an internal combustion engine using a neural network, in *Advances in Vehicle Control and Safety International Conference, AVCS*, vol. 98 (1998)
18. A.M. Frith, C.R. Gent, A.J. Beaumont, Adaptive control of gasoline engine air-fuel ratio using artificial neural networks, in *4th International Conference on Artificial Neural Networks* (1995), pp. 274–278
19. I. Arsie, M. Sorrentino, C. Pianese, A neural network air-fuel ratio estimator for control and diagnostics in spark-ignited engines. *IFAC Proc. Vol.* **40**(10), 227–234 (2007)
20. M. Nørgaard, O. Ravn, N.L. Poulsen, L.K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems* (Springer, Heidelberg, 2000)
21. C. Alippi, C. de Russis, V. Piuri, A neural-network based control solution to air-fuel ratio control for automotive fuel-injection systems. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **33**(2) 259–268 (2003)
22. N. Cesario, M. Di Meglio, F. Pirozzi, G. Moselli et al., Air/fuel control system in SI engines based on virtual lambda sensor. SAE Technical Paper 2005-24-058 (2005). doi:[10.4271/2005-24-058](https://doi.org/10.4271/2005-24-058)
23. V.N. Vapnik, An overview of statistical learning theory. *IEEE Trans. Neural Netw.* **10**(5), 988–999 (1999)
24. C. Mobley, Non-intrusive in-cylinder pressure measurement of internal combustion engines. SAE Paper no 1999-01-0544 (1999)
25. D. Panousakis, J. Patterson, A. Gasiz, R. Chen, Analysis of SI combustion diagnostics methods using ion-current sensing techniques. SAE Paper no 2006-01-1345 (2006)
26. M. Włodarczyk, High accuracy glow plug-integrated cylinder pressure sensor for closed loop engine control. SAE Paper no 2006-01-0184 (2006)
27. S.J. Citron, J.E. O'Higgins, L.Y. Chen, Cylinder by cylinder engine pressure and pressure torque waveform determination utilizing speed fluctuations. SAE Paper no 890486 (1989)

28. G. Rizzone, Diagnosis-of individual cylinder misfires by signature analysis of crankshaft speed fluctuations. SAE Paper no 890884 (1989)
29. F.T. Connolly, A.E. Yagle, Modeling and identification of the combustion pressure process in internal combustion engines using engine speed fluctuations. *Am. Soc. Mech. Eng. Dyn. Syst. Control Div.* **44**, 191–206 (1992)
30. D. Moro, N. Cavina, F. Ponti, In-cylinder pressure reconstruction based on instantaneous engine speed signal. *J. Eng. Gas Turbines Power* **124**, 220–225 (2002)
31. S. Saraswati, S. Chand, Reconstruction of cylinder pressure for SI engine using recurrent neural network. *Neural Comput. Appl.* **19**, 935–944 (2010)
32. F. Gu, P.J. Jacob, A.D. Ball, A RBF neural network model for cylinder pressure reconstruction in internal combustion engines, in *IEEE Colloquium on Modelling and Signal Processing for Fault Diagnosis (Digest no: 1996/260)* (1996)
33. F. Tagliatalata, M. Lavorgna, E. Mancaruso, B.M. Vaglieco, Determination of combustion parameters using engine crankshaft speed. *Mech. Syst. Signal Process.* **38**(2), 628–633 (2013)
34. S. Park, P. Yoon, M. Sunwoo, Feedback error learning neural networks for spark advance control using cylinder pressure. *Proc. Inst. Mech. Eng. Part D: J. Automob. Eng.* **215**(5), 625–636 (2015)