A Neural-Network Based Control Solution to Air-Fuel Ratio Control for Automotive Fuel-Injection Systems

Cesare Alippi, *IEEE Senior Member*, Cosimo de Russis, and Vincenzo Piuri, *IEEE Fellow*

***Abstract*—Maximization of the catalyst efficiency in automotive fuel-injection engines requires the design of accurate control systems to keep the air-to-fuel ratio at the optimal stoichiometric value AF_S. Unfortunately, this task is complex since the air-to-fuel ratio is very sensitive to small perturbations of the engine parameters. Some mechanisms ruling the engine and the combustion process are in fact unknown and/or show hard nonlinearities. These difficulties limit the effectiveness of traditional control approaches. In this paper, we suggest a neural based solution to the air-to-fuel ratio control in fuel injection systems. An indirect control approach has been considered which requires a preliminary modeling of the engine dynamics. The model for the engine and the final controller are based on recurrent neural networks with external feedbacks. Requirements for feasible control actions and the static precision of control have been integrated in the controller design to guide learning toward an effective control solution.**

***Index Terms*—Air-fuel ratio control, automotive fuel injection, air pollution, neural network control, recurrent neural networks.**

I. Introduction

In THE last years, we observed an increasing attention toward problems related to the environment with a particular focus on pollutants generated by vehicles in industrialised countries. Since 1970, the European community has set strict requirements on the maximum exhaust emissions tolerated for a vehicle hence forcing the automotive industries toward the *zero emission vehicle* (ZEV) target. Fig. 1 shows the European constraints on pollutants measured in g/experiment (a benchmark of the European community) by considering the base indexes (year 1970) as reference points. Similarly, the state of Californian government has required that by 2004 10% of vehicles must be ZEV, while the others must reduce exhaust emissions of 60–84% with respect to actual values [1]. These strong constraints pushed the research toward the development of suitable electronics, embedded systems, and mechanical and chemical devices to reduce noxious emissions.

Unleaded fuel, catalytic converters, and an accurate control of the variables involved in the fuel combustion process are relevant ingredients to reach such a goal.

Manuscript received August 12, 2000; revised April 21, 2003. This paper was recommended by Associate Edtior J. Lee.

C. Alippi is with the Politecnico di Milano, 20133 Milano, Italy.

C. de Russis was with Centro Ricerche FIAT, 10043 Orbassano, Torino, Italy. He is now with the RSI Sistemi Spa, 00144 Roma, Italy.

V. Piuri is with the University of Milan, 26013 Crema, Italy (e-maill: piuri@elet.polimi.it).

Digital Object Identifier 10.1109/TSMCC.2003.814035

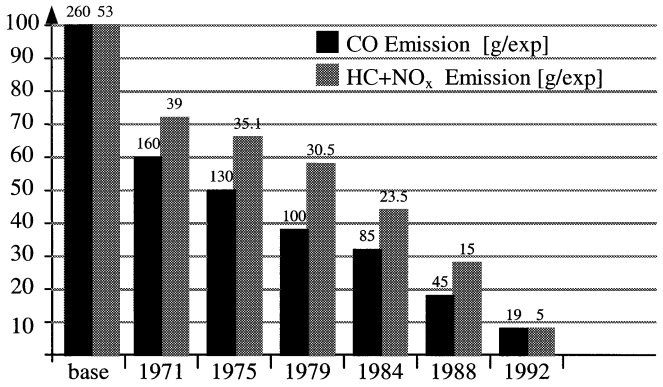

Fig. 1. Limits on exhaust emissions imposed by the European community in years 1970–1992.

In this paper, we consider a fuel-injection system composed of a spark ignition engine with a catalytic converter and a linear oxygen sensor on the exhaust manifold to measure the air-to-fuel ratio (AF) after the combustion process. The case study is an *Alfa Romeo 1.3l* engine.

Pollutants are generated during the combustion process, which can be modeled as [2]

$$\frac{1}{\mathrm{AF}}\mathrm{C_8H_{18}} + (\mathrm{O_2} + 3.77\mathrm{N_2}) \rightarrow a_1\mathrm{CO_2} + a_2\mathrm{CO} + a_3\mathrm{NO} + a_4\mathrm{O_2} + a_5\mathrm{N_2} + a_6\mathrm{H_2O} + a_7\mathrm{CH_4} \quad (1)$$

where $\mathrm{C_8H_{18}}$ is the fuel, $(\mathrm{O_2} + 3.77\mathrm{N_2})$ is the air mixture, $\mathrm{CH_4}$ accounts for all the residual unburned HCs, and a_i $(i = 1 \ldots 7)$ are some reaction coefficients affected by AF in a nonlinear way. More in detail

- when AF is in the AF_S neighborhood, the combustion process generates all the reaction products present in (1);
- in the case of a lean mixture $(\mathrm{AF} > \mathrm{AF}_S)$, the model simplifies and it is assumed that we can neglect the a_2, a_3, a_7 coefficients;
- in the case of a rich mixture $(\mathrm{AF} < \mathrm{AF}_S)$, the model is such that $a_1 = a_3 = a_4 = 0$.

In the last years, much effort has been directed toward the development of devices capable of properly working with lean AF mixtures (that means low emissions and fuel savings), but much work is still needed; [3] represents an interesting example in this direction.

A different approach attempts to reduce pollutants by processing the exhaust gases. Catalytic converters follow this direction by accelerating the chemical process of oxidation for HCs

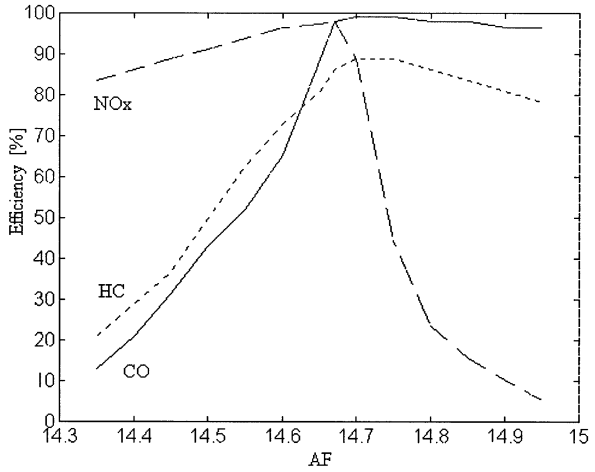

Fig. 2. Efficiency of a catalytic converter as a function of AF.

and CO to H_2O and CO_2 and reduction of NO_X to N_2. To this end, it has been proved that maximal efficiency of the catalyst can be obtained by keeping AF within a very strict band around its stoichiometric value $\mathrm{AF}_S = 14.64$ [4], [5]. Fig. 2 shows the efficiency of the catalytic converter in reducing pollutants as a function of AF; 1.0 in the ordinate means a 100% efficiency in reduction. We note that even small variations around this value may cause severe loss in efficiency [5], [6]; in Fig. 2, a 1% discrepancy in AF with respect to the stoichiometric value may cause up to a 50% reduction of the catalytic converter efficiency in reducing pollutants.

Controlling AF so that it equalizes AF_S is a difficult task. This is due to the nonlinear behavior and the cyclic nature of the engine within a wide operating range, the presence of uncertainties and unpredictable disturbances in the process, and the great difficulties in inferring relevant variables from imprecise or difficulty measurable ones.

Classic feedback controllers for AF [6]–[8] provide a control action, which depends on two additive contributions. The first contribution comes from a closed loop control of AF, which relies on information coming from the exhaust gases oxygen (EGO) sensor. This control allows for maintaining AF around its stoichiometric value but it is not effective in providing a prompt control reaction during transients. The second contribution addresses this issue by considering an open loop controller based on transient information coming from the engine (e.g., the engine angular speed, the opening of the throttle valve, etc.). The second contribution is fundamental for an accurate control of AF and is based on the transient fuel film compensation (TFC) model described in [7], [8]. In this case, the control algorithm cancels the fuel film dynamics and injects the right fuel quantity into the engine intake manifold at the correct time. The effectiveness of TFC strictly depends on the characterization of the fuel transfer mechanism: the simple model generally used to represent the phenomenon [7]–[9], and here adopted, supposes that a fraction of the fuel delivered to the intake system deposits on the manifold surfaces. Such a fuel then evaporates with a rate dependent on the mass of fuel in the puddle and a delay time. In spite of its linear formulation [8], 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 transients control.

An adaptive compensation of the fuel dynamics by means of a fuzzy-based algorithm is suggested in [10]; there the author considers a switching EGO sensor for its low cost. In this study we prefer to consider a linear sensor for its accuracy in providing a measure of the amount of unburned oxygen in the exhaust gases. The actual trend in the automotive industry is in this direction and many vehicles already implement this medium cost sensor.

Due to the presence of uncertainties and nonlinearities in the process nonlinear black-box solutions as neural networks become attractive techniques to be investigated for an optimal control of AF. The expanding use of neural networks in identification and control areas influenced the automotive field with relevant contributions, e.g., in the area of anti-lock braking systems [11], engine idle-speed control [12], transient AF control [4], [13], [14].

In this paper, we study the applicability and the effectiveness of neural networks to control the AF/ratio. We believe that for its simple nature an optimized neural network is an interesting solution for the AF control, which could be inserted in the future within the electronic control module (ECM) of a vehicle. In addition, evolution of electronics, very-large-scale integration (VLSI) integration and reconfigurable devices such as field programmable gate arrays (FPGAs) will make feasible online training for the parameters of the neural network so as to deal with aging effects, inefficiencies in modeling dynamics and support an accurate tune of the controller's parameters to the specific engine. This last aspect is particularly appealing for sport cars characterized by high performance and low productions.

An online identification of the dynamics of AF in injected engines and, therefore, a time varying model of the engine to be used for a subsequent adaptive control design is given in [15].

In the following we consider neural networks as core elements for function approximation and offline dynamic modeling to test the validity of the approach; future developments will investigate online training and adaptive solutions.

Function approximation is considered to improve the quality of the equation-based model of the system which, once assessed, can be used to design the controller; in addition to the control issue the obtained neural networks and can be used to improve the accuracy of the engine model. In developing the controller, we adopted the classic indirect control configuration of Fig. 3 (e.g., see [16], [17]), which requires a preliminary identification procedure, here carried out with recurrent neural networks. The reason for such a choice is that the final neural model I of the process P is

- continuous and differentiable (hence, allowing for the use of gradient-based algorithms to subsequently train the neural controller C);
- robust with respect to feasible perturbations in its input variables. At the beginning of learning the control input u does not assume significant values and, in a direct control configuration, it could provide unfeasible inputs for the model P (which might become, as happens in our application, numerically unstable). Conversely, the neural model

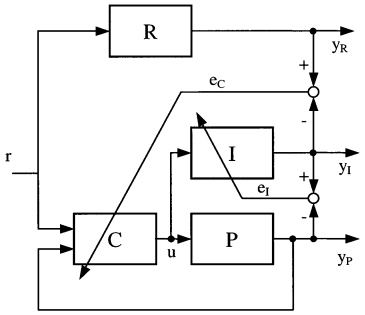

Fig. 3. Indirect control configuration.

I provides a graceful degradation in performance subject to the same perturbations;

- characterized by a reduced computational load, which simplifies the training of the controller.

Fortunately, it is not required an accurate model for the whole process, but only a good model covering the operating conditions of it. The model I of P is then used to design the controller and tune its parameters for tracking the reference signal generated by R. We will require the controller to provide a null error at the end of the transient phase and a feasible control action for u.

The suggested approach is, in a way, related to [18]. Differently from them we provide a static neural model to characterize some parameters ruling the wall wetting process: this allows for maintaining the simple and effective linear model of [7] (which also grants an immediate evaluation of the model stability). As far as the neural control is considered, we opted for feedforward neural networks with external feedbacks instead of the hybrid feedforward/feedback solution given in [18] (where the hidden layer resembles the topology of a Hopfield network). Training the first model is simpler than the second, hence making more feasible the possibility of a future online implementation of the controller in the ECM. Our network design also simplifies the straight hardware implementation of the neural network both with respect to data path and control (only the delayed outputs are in fact presented to the network's input). In addition, we considered a hybrid neural controller in which requirements for feasible control actions and the static precision of control have been integrated in the control design to guide the training toward an effective control solution.

The structure of the paper is as follows. The equation based model P of the engine is briefly introduced in Section II. Section III deals with the development of a neural model for the deposition coefficient and the evaporation time constant. Such models are used to improve the model accuracy of the engine necessary for the controller design. Section IV addresses the problems related to configuration of a dynamic recurrent neural model I of the process as required by the indirect control approach. Such a neural model is finally used in Section V to design the controller C.

II. Modeling the Dynamics of the AF Ratio

In this work, we considered a sequential multipoint injection system characterized by an injector for any single cylinder; results can be easily extended to deal with single point or full-group multipoint injection systems.

The first step in designing a controller is to provide an accurate mathematical description of the whole system, here the process leading to AF. This task can be accomplished by writing all physical equations describing the processes involved and identify the unknown parameters from experimental data (e.g., see [15]). When the mechanisms ruling a process are totally unknown or its equation-based model becomes computationally prohibitive a black-box approach becomes the only viable solution.

A simple functional description for AF is given in Fig. 4. N represents the engine angular speed (measured by the engine speed pickup sensor), α (alpha) represents the opening of the throttle valve (driven by the accelerator pedal), T_e and P_e are the external temperature and pressure, respectively, T_m is the engine temperature, and t_{j_com} (or t_j) is the fuel injection time. The most interesting blocks, which directly affect AF measured by the exhaust gases oxygen sensor $\mathrm{AF}_{\mathrm{meas}}$, are the fuel film deposition block, whose model is described and improved in Section III, the exhaust pipe, and the oxygen sensor blocks. Each block is described by choosing the most suitable physical driven model either available in the literature or developed at FIAT. For such models we neglected all those dynamics, variables or secondary order effects not relevant to the design of the controller.

III. Neural Reconstruction of the Parameters Affecting the Liquid Fuel Film Deposition

As we mentioned in the introduction, the most critical tasks that the ECM has to manage are the engine transient conditions, i.e., those situations where the fuel supply rate has to be rapidly adjusted to face the air flow response on demand changes. During such situations the injected fuel does not produce the quantity required in the cylinder because of fuel deposition and transportation mechanisms [9]. These behaviors are of primary importance in affecting AF and must be carefully modeled to develop an effective model for the engine. More in detail, once a certain amount of fuel m_{fi} has been injected, only a fraction $1-\chi$ (χ being the deposition coefficient) reaches the combustion chamber (in a simple but effective linear model with a constant of time τ_{fv}). The fraction χ condenses on the manifold walls (from which the name wall wetting). Only afterwards the condensed fuel m_{ff} evaporates (in a linear model with constant of time τ_{ff}) and contributes to increase the effective fuel amount. Let m_{fo} be the effective fuel evaporated and m_{fv} the injected one both reaching the combustion chamber. The whole process is described by the following system of equations [7]:

$$\begin{cases} \dot{m}_{fv} = \frac{1}{\tau_{fv}}((1-\chi)\dot{m}_{fi} - \dot{m}_{fv}) \\ \dot{m}_{ff} = \frac{1}{\tau_{ff}}(\chi\dot{m}_{fi} - \dot{m}_{ff}) \\ \dot{m}_{fo} = \dot{m}_{fv} + \dot{m}_{ff}. \end{cases} \tag{2}$$

The model can be simplified by observing that the constant of time τ_{fv} is about one tenth of τ_{ff} [8]: the dynamic of the process can be described with a single constant of time $\tau = \tau_{ff}$. Unfortunately, τ_{ff} and χ are unknown, nonlinear, and strongly dependent on the revolution number of the engine N and the pressure

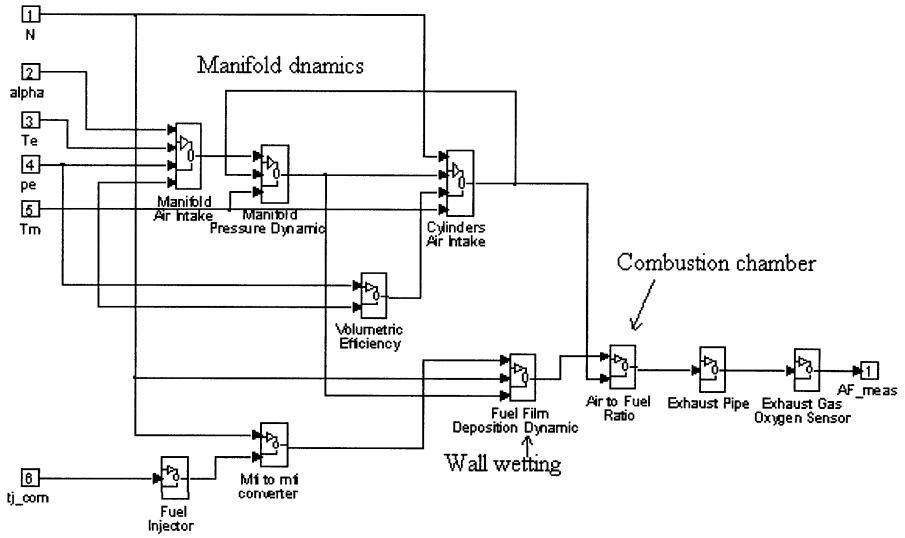

Fig. 4. Fundamental blocks constituting the process.

at the intake manifold Pm. τ_{ff} and χ are not directly measurable and must be derived from experiments. More in detail, a test engine has been constructed and driven to different N; τ_{ff} and χ are those parameters which best explained the measured data according to the process depicted in Fig. 4 and the linear model (2) for the wall wetting phenomena.

In general, such parameters are filtered with linear techniques, interpolated to obtain the relationships $\tau = \tau(Pm, N)$ and $\chi = \chi(Pm, N)$, and placed in lookup tables. The availability of a limited number of noisy data belonging to nonlinear functions limits the effectiveness of linear techniques and suggests considering model-free function approximations, e.g., by using neural networks.

The knowledge of τ_{ff} and χ over the whole feasible working points of the engine is extremely important since it allows creating an accurate model for AF. In our problem such a model will be directly used for developing a neural network modeling the engine and indirectly to develop the neural controller.

A. Selecting an Appropriate Neural Model

To develop a model for $\tau = \tau(P_m, N)$ and $\chi = \chi(P_m, N)$, we considered feedforward neural networks of regression type [19]: in particular, we focused the attention on three-layered neural networks with n_i input neurons, n_n hidden units, and a single output neuron. The hidden units are characterized by hyperbolic tangent activation functions and the output of the neuron is linear in its activation value.

Several techniques have been suggested in the literature to design an optimal neural model by solving the compromise performance vs. network complexity (e.g., spectral decomposition [20], optimal brain damage [21], optimal brain durgeon [22], shared weights [23], early stopping [24]). In such methods the best model is the one minimizing the generalization error on the cross-validation set. Cross-validation presents a serious disadvantage when a limited data set is available: saving examples to cross-validate a model reduces the data available for configuring the weights of the network, hence impairing the efficiency of learning. In such a case, all data should be used for training, thereby, making it necessary to use criteria, which estimate the generalization ability of the neural model directly from the training data. The generalized prediction error [28], the network information criterion [26] and the final prediction error biased (FPEB) criterion [27] are criteria following this principle.

Our goal is to determine the function $\hat{y}(x)$ which best approximates the unknown function $\bar{y}(x)$ given the n measured input/output pairs and the classic mean squared error (MSE) loss function. In the following, the unknown functions $\bar{y}(x)$ to be inferred are $\bar{y}_1 = \tau$ and $\bar{y}_2 = \chi$, while the input vector is $x = [Pm, N]$.

Intuitively, the FPEB criterion suggests that the optimal neural model for the given application must provide good performance on the training set and be topologically simple. The criterion is defined as

$$\text{FPEB} = \text{MSE}(\hat{\theta}) + \frac{1}{n}\text{tr}(Q(\hat{\theta})(\text{MSE}''(\hat{\theta}))^{+}) \tag{3}$$

where $\hat{\theta}$ is the vector of weights and biases of the neural network, tr is the trace operator, $+$ is the Moore–Penrose pseudoinverse, $\psi(x, \hat{\theta}) = (d\varepsilon/d\theta)|_{\theta=\hat{\theta}}$ with $\varepsilon = y - \hat{y}$ is the error gradient, $Q(\hat{\theta}) = (1/n)\sum_{i=1}^{n} \varepsilon_i^2 \psi_i(\hat{\theta})\psi_i(\hat{\theta})^T$, and $\text{MSE}''(\hat{\theta})$ is an estimate of the MSE Hessian (see [27] for details). The criterion can be seen as the sum of two contributions: the first term addresses the performance of the considered model over the training set, the second one is related to the number of available data and the model complexity. It is obvious that an overdimensioned model provides good performance on the training set by overfitting the available data, but it will be penalised by the second term of FPEB.

We considered the FPEB criterion and not the more simple techniques based on early stopping since the latter are quite sensitive to the stopping criterion used [28]. Note that, in real applications, FBEB is not very computationally intensive and can be immediately computed on software platforms such as Matlab® or Mathematica®.

TABLE I
FEASIBLE RANGE FOR THE INPUTS AND THE OUTPUT WITH THEIR UNITS

$\alpha_{meas} \in (7,70)$	$N_{meas} \in (700,4700)$	$t_j \in (0.001,0.009)$	$P_{m\_meas} \in (20000,100000)$	$AF_{meas} \in (6.7,22.7)$
[deg]	r/min	[s]	[Pa]	[-]

B. Determination of the Optimal Neural Models for χ and τ

The determination of an optimal neural model requires training of different neural networks (which differ in the number of hidden units) and computing the FPEB for each of them. The network for which the FPEB is minimal solves the function approximation task.

Each training set was limited to 256 values ($16 * 16$ in N and Pm) and obtained after an experimental campaign carried out at FIAT on the Alfa Romeo 1.3l engine.

Learning was performed by considering the quasi-Newton Levenberg–Marquardt training algorithm [29] applied to a set of neural models with hidden units varying from 2 to 27. We monitored the evolution of FPEB over training time and we stopped it when the FPEB became either constant or increasing and the number of effective number of parameters used by the model [27] was constant.

This methodology selected a network with 13 hidden units to approximate $\tau = \tau(Pm, N)$ and a network with 10 hidden units for $\chi = \chi(Pm, N)$. The determined neural networks have been inserted in a library and integrated into the model of the engine as black boxes to be subsequently used to design the neural identifier I and the controller C of Fig. 3.

IV. NEURAL IDENTIFICATION OF THE AF RATIO

Identification of a dynamic system with neural networks requires few steps to be accomplished: extraction and decimation of data for training, test and validation; selection of a suitable family of recurrent neural models; choice and implementation of a recursive training algorithm. Each issue plays a relevant role in configuring an effective neural model and requires a careful analysis of the *a priori* knowledge about the process to be identified. To this end, we followed the identification methodology suggested in [30].

Determination of a proper neural family is one of the main design topics: a wrong choice generates a large model bias, which —most of times— leads to bad results. Despite the large literature on configuring recurrent networks (see [31] for a review) a trial and error approach is somehow necessary to select the most appropriate neural network. To identify AF ratio, as required by the indirect control approach, we chose the neural output error models for their successful performances on preliminary experiments (we also experimented memory neurons networks [31] with scarce results). In particular, we considered a recurrent single-layered network with an arbitrary number of hidden units characterized by hyperbolic tangent activation functions and a single linear output. The suitably delayed output and external inputs constitute the network inputs [16].

Data extraction is one of the most critical phases. Data need to be extracted (possibly in an automatic way) so as to cover all relevant working points of the process. If the data set is not sufficiently informative (as it happens when inputs are not able to fully excite the process dynamics), the best model, in the best scenario, is able to approximate the behavior of the process only in the interval defined by the training pairs.

Data extraction requires a preliminary analysis to determine the relevant inputs affecting AF. By relying on the equation-based model presented in Section II, we discovered that the measured AF $\mathrm{AF}_{\mathrm{meas}}$, obtained by processing data coming from the exhaust gas oxygen sensor can be expressed as

$$\mathrm{AF} = \mathrm{AF}(Pm, N, \alpha, t_j). \tag{4}$$

All inputs are either directly measurable, or known. A time dependency analysis was then carried out to determine the appropriate time delays for the inputs and the output to be presented to the neural network. To this purpose, the most critical element is the exhaust manifold, which pipes the post-combustion gases toward the oxygen sensor. These gases reach the sensor with an unknown delay Δt_f function of with N, modeled as a pure delay in classic models. This restrictive assumption was relaxed in our application.

By indicating with T_c the sampling time and with T the integer number closest to $\Delta t_f / T_c$, expression (4) becomes

$$\begin{aligned}\mathrm{AF}_{\mathrm{meas}}(t+1) = g(&\mathrm{AF}_{\mathrm{meas}}(t), \alpha(t-T+1),\\ &N(t-T+1, t-T), t_j(t-T+1, t-T),\\ &P_m(t-T+1, t-T)).\end{aligned} \tag{5}$$

where $x(t-T+1, t-T)$ means that there is a dependence in the $(t-T+1)-(t-T)$ time interval. Equation (5) completely describes, at a functional level, the relevant inputs and their time dependencies on $\mathrm{AF}_{\mathrm{meas}}$.

The determination of the unknown T_c and T gives information about the structure of the inputs for the neural network, reduces the number of networks to be trained, and simplifies the identification procedure. To estimate such parameters, we identified the feasible working operational domain for the input and the output signals, as presented in Table I. Inputs must resemble feasible signals: steps (a pressure or a depression of the accelerator pedal) and ramps (constant acceleration) for α, t_j, and P_m, and ramps for N. The use of pseudorandom binary signals to excite the inputs is prevented by the fact that the model P becomes numerically unstable. Steepness, time duration and amplitudes for steps and ramps must be chosen randomly so as to model all the actions the driver and the ECM might envisage. We extracted such parameters from a uniform distribution. Both transient and steady-state conditions have been generated to provide signals covering the whole frequency spectrum.

Data have been then decimated to reduce the training set size (and implicitly the training time), avoid irrelevant data redundancy, and filter high frequency noise. The sampling time T_c, has been identified by means of a fast fourier transform (FFT) applied to inputs and output as follows. We determined

TABLE II
PERFORMANCES OF DIFFERENT NEURAL TOPOLOGIES

Network Topology	MSE	Mean Error	Mean % Error
1 Hidden Layer (15 neurons)	0.45	0.17	1.27%
1 Hidden Layer (20 neurons)	0.07	0.38	2.70%
1 Hidden Layer (25 neurons)	0.11	0.53	3.95%
2 Hidden Layers (8+8 neurons)	0.10	0.50	3.68%
2 Hidden Layers (10+8 neurons)	0.11	0.52	3.79%
2 Hidden Layers (10+10 neurons)	0.67	0.34	2.11%

the signal among inputs and output with the largest band and its Nyquist frequency; we then multiplied such a value by a confidence parameter hence discovering that it was possible to decimate data at $1/T_c = 30$ Hz without any information loss.

By investigating the feasible signals we observed that Δt_f, spans the 0.025–0.166-s interval. This implies, according to the sampling rate, that we need from one up to five time delays for the external inputs. Such a number has an immediate impact on the number of inputs feeding the neural networks and on training time. We should consider up to five delays for each input variable, which leads to a network with 24 external inputs. We discovered that it was possible to consider two delays for the inputs and three delays for the output dynamic with a small loss in accuracy. The final neural network receives 15 inputs: P_m, N, α, t_j at time $t, t-1, t-2$, and $\mathrm{AF}_{\mathrm{meas}}$ at time $t-1, t-2, t-3$.

Training was implemented with the quasi-Newton Davidon–Flecter–Powell algorithm [29]. We obtained poor results by using simpler training algorithms based on a straight gradient descent as back-prop through time and a quick-prop modified to account for the recurrent configuration. The considered algorithm was modified according to the William–Zipser correction to account for the recurrent configuration and the teacher-forcing modality [32] to optimize the restart of the batch algorithm. The first correction accounts for the fact that training data are related in time and allows for writing an iterative formulation for the gradient computation which speeds up training. The teacher-forcing modality forces the real initial conditions to the network inputs instead of using the ones estimated by the network so as to reduce the initial misalignment between the real and the neural-provided output trajectory. In our application, we could not obtain acceptable performance without these two mechanisms. Other equivalent algorithms have been presented in [33].

Experimental evidence proved that it was necessary to consider a training set of at least 30 000 pairs for a total of 16.5 h of driving in order to achieve a good accuracy. The test set was introduced to implement some sort of controlled early stopping.

Different experiments were carried out by varying the number of hidden layers (1 and 2) and the number of neurons per layer. Results are given in Table II: MSE is the MSE during test, while the mean error and the mean % error are defined as the averaged

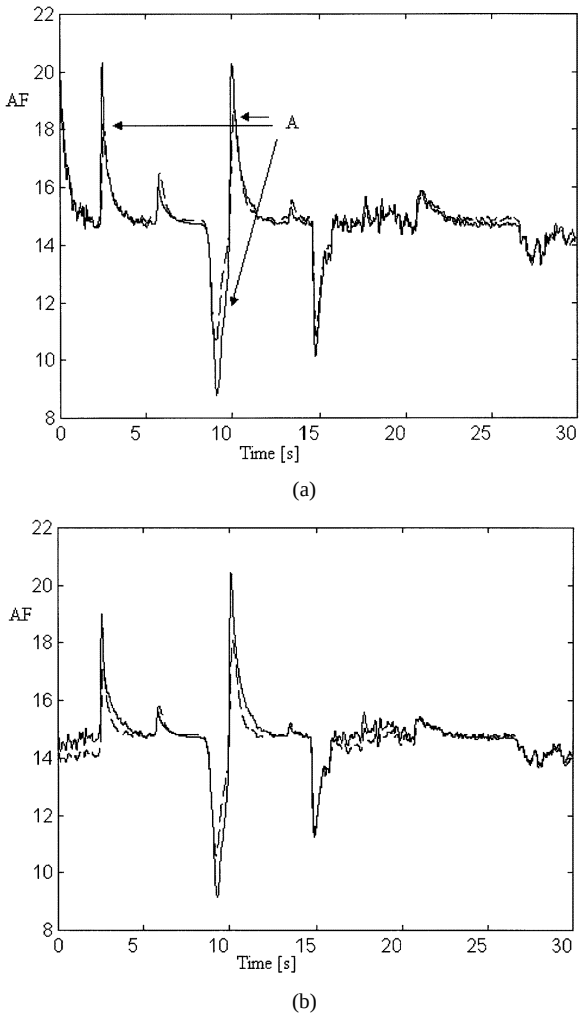

Fig. 5. Validation on a real data set. (a) $N = 4000$ r/min. (b) $N = 3000$ r/min.

value of $|\mathrm{AF}_{\mathrm{meas}} - \mathrm{AF}|$ and $|\mathrm{AF}_{\mathrm{meas}} - \mathrm{AF}|(100/\mathrm{AF}_{\mathrm{meas}})$ in validation, respectively. Here, AF is the air to fuel ratio provided by the neural model.

Validation was carried out with real data coming from the Alfa Romeo engine. Fig. 5 shows the AF validation over time; the neural output and the real one are plotted with a continuous line and a dashed line, respectively. In particular, Fig. 5(a) refers to a $N = 4000$ r/min case, Fig. 5(b) to $N = 3000$ r/min.

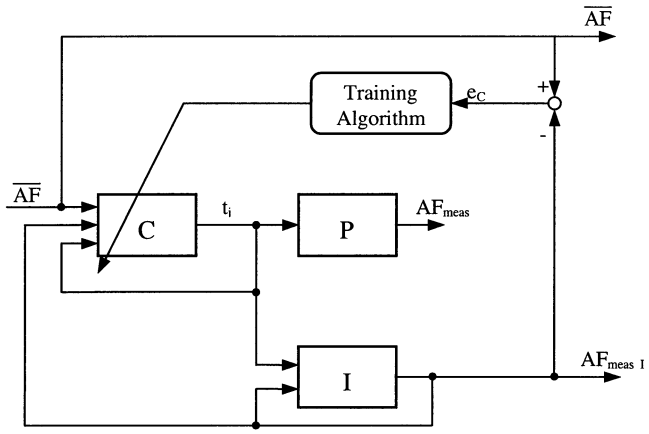

Fig. 6. Training configuration for the neural controller.

The neural model provides a good accuracy around the stoichiometric value, while performance slightly decreases in correspondence with high peaks (e.g., points A): there is no need to improve the model since we are interested in developing a controller working in the stoichiometric neighborhood and high peaks are unfeasible situations. Moreover, it has been proved [6] that fast oscillations ($>$1 Hz) in AF with small amplitude (<0.1) do not significantly degrade the conversion efficiency due to the converters averaging properties.

V. Neural Control of the AF Ratio

The final step is to determine the controller C. The variable to be controlled is the injection time t_j, provided by the ECM on the bases of the information coming from the sensors and the control algorithm, which contains a neural network similar to the one used to identify AF. Training is somewhat different now and relies on the information coming from AF estimated by the neural model $\mathrm{AF}_{\mathrm{meas}\_I}$ designed in Section 4. A block description of the training configuration is given in Fig. 6, where $\overline{\mathrm{AF}} = 14.64$ is the stoichiometric value for AF.

Such a control scheme is quite flexible and allows the network for subsequent online training. This aspect has been tackled in [17], where the normal operation is interleaved with training both the controller and model I. As we mentioned, on line training is particularly appealing since it allows dealing with the process aging evolution and outcomes the fact that recurrent neural networks are able to approximate a dynamic system only for a finite amount of time [31]. Online training implies that the learning algorithm is inserted in the ECM and must be therefore extremely simple from the computational point of view. To this end, we tested the feasibility of this possibility by considering a very simple online training based on a straightforward gradient descent algorithm in which parameters are updated after the presentation of a single pair, as suggested in [16].

Today, given the limited onboard computational resources and the amount of tasks an ECM has to perform (including on-board diagnosis), the low memory requirement and real time constraint, online training seems unfeasible. This problem should be anyway solved within few years thanks to the advance of electronics and the trend of architectural design toward processors containing independent functional units, as happens with very-long-instruction-word (VLIW) processors. It is reasonable to imagine a scenario in which the actual processes running within the ECM leave a functional unit free to issue those instructions necessary for online training.

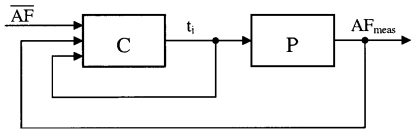

Fig. 7. Operating configuration for the neural controller.

Online training must not necessarily be continuous but must be enabled to refine the controller only when required. This, in addition to face ageing effects, also improves the features of the controller by tuning the controller to the specific engine. Whether or not continuous training is envisaged, once the controller has been configured within the ECM, it commutes to the normal operation modality. All the information comes directly from the sensors, as shown in the configuration of Fig. 7: the controller C receives $\mathrm{AF}_{\mathrm{meas}}$ now measured by the exhaust gases oxygen sensor and provides the control variable t_j.

In developing a controller for AF we have to satisfy two requirements: 1) a null error at the end of the transient phase; and, 2) a feasible control signal (here, the injection time t_j) characterized by a limited control action. The first requirement can be tackled by considering one simple integrator, which acts on $\overline{\mathrm{AF}} - \mathrm{AF}_{\mathrm{meas}\_I}$ during training and on $\overline{\mathrm{AF}} - \mathrm{AF}_{\mathrm{meas}}$ during the normal operations. The integrator must be inserted before the neural controller and guarantees a null error at the end of the transient phase. This requirement could have been directly integrated in the training procedure; however, we experimented that this approach is only partly effective. In fact, because of its recurrent structure, the neural network may be significantly biased so to introduce an error at the steady state. Even if this error may be tolerated in other applications, the control of AF imposes a strict requirement on high accuracy.

The feasibility of the control action was obtained by inserting directly in the training error function a term penalising severe control signals.

Experiments proved that the above two modifications were not sufficient to grant an acceptable control action. In fact, we experimented that, to force a prompt control, the neural controller was using a control action t_j with high frequency components. Even if appealing, such a controller design is not acceptable in an effective control of the fuel injector. We had therefore to limit the bandwidth of the control action by requiring t_j to be more regular. This requirement was obtained by inserting in the training error function a further penalty term acting on the first derivative of t_j. The constraints force the training algorithm to restrict the search in the neural parameter space toward a controller able to provide a regular control signal over time. The final training function is the following:

$$J(t) = \alpha(\overline{\mathrm{AF}} - \mathrm{AF}_{\mathrm{meas}\_I}(t))^2 + \beta(t_j(t))^2 + \gamma(t_j(t) - t_j(t-1))^2 \quad (6)$$

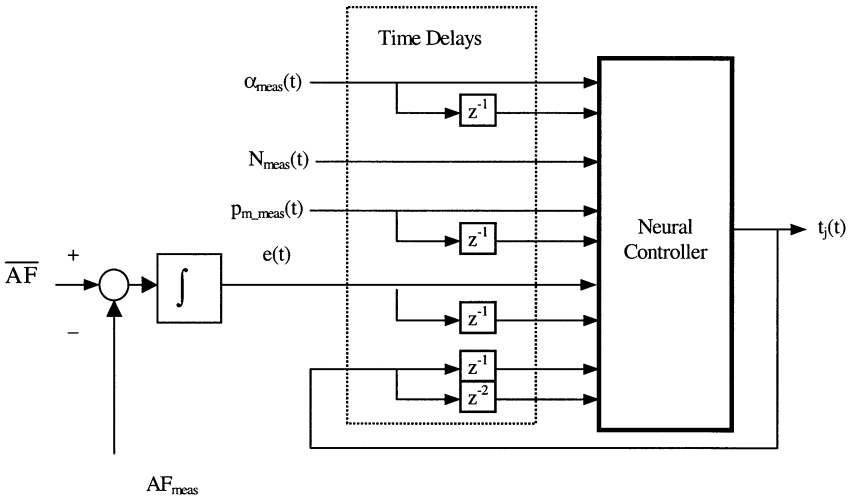

Fig. 8. Final operational scheme for the neural controller.

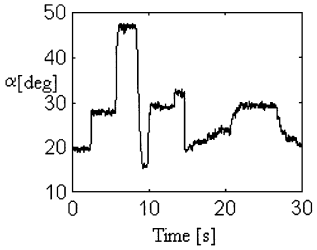

Fig. 9. α for experiment numbers 1 and 2.

where α, β, and γ are some coefficients weighting the relevance of the three constraints.

Again, we have to configure the neural controller in terms of inputs and output. As for the module I, we determined the functional and the time dependencies of t_j. In full agreement with more classic controllers (e.g., see [6]–[8]), this phase suggests that the controller receives Pm, N, and α (which can be modeled here as a disturbance) and provides the injection time t_j to be given to the engine. The structure of the controller during training is shown in Fig. 8, where z^{-1} represents the one step delay. During the normal operations, it must receive the measured $\mathrm{AF}_{\mathrm{meas}}$ instead of $\mathrm{AF}_{\mathrm{meas\_}I}$.

The neural controller was trained at the beginning with the robust algorithm used to identify the process I and, then, was commuted over a simple gradient descent algorithm applied to the error function $J(t)$. The inputs have been defined over a rough input profile of the type used for the identification process. Again, we considered a test set to decide when to interrupt the training phase.

We noted that the presence of a control action created problems in the initial phases of learning, where t_j provides wrong values. In this case, the integrated error and the training procedure diverge. The problem can be solved either by bounding the integrator with a saturation mechanism or by initially training the controller without the integral action and inserting it later on. Both solutions were equally effective in our application. We determined from experiments that the best neural network was characterized by ten hidden units.

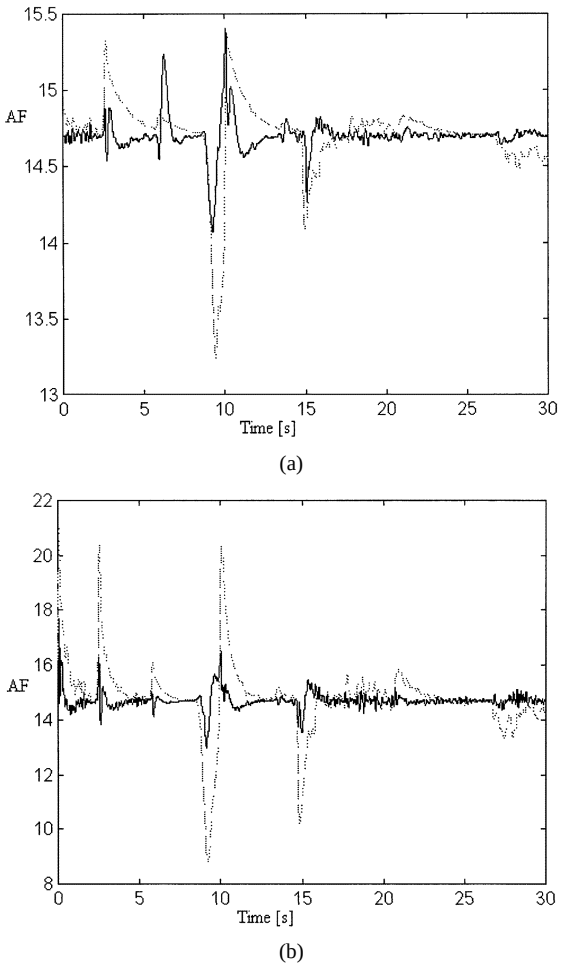

Fig. 10. Controlled AF. (a) $N = 1000\,\mathrm{r/min}$. (b) $N = 4000\,\mathrm{r/min}$.

The neural structure of the controller was validated over different validation sets composed of real data; the two experiments we present refer to data measured at FIAT. The input profile for α is given in Fig. 9. The profile has been obtained by pushing and releasing the accelerator pedal: we start from a constant pressure of a 20° angle and, then, we accelerate by reaching a 47° angle, and so on. N is equal to 1000 r/min in the

TABLE III
COMPARISON IN PERFORMANCES BETWEEN THE HYBRID CONTROLLER AND THE TFC ONE

Experiment	>3 TFC	>3% NNC	Mean % TFC	Mean % NNC
1	6.40%	2.17%	0.93	0.35
2	37.77%	8.33%	4.69	1.09

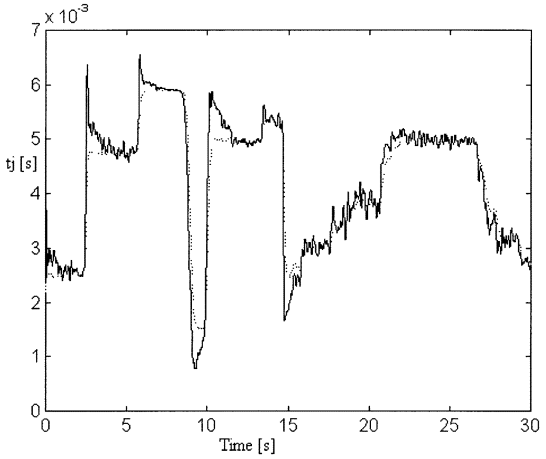

Fig. 11. Control actions (t_j) corresponding to the second experiment.

first experiment and 4000 r/min in second one. In Fig. 10(a) and (b), we present the main results achieved in our experiments: the control action provided by the neural controller is shown in a continuous line while performance provided by the TFC controller is given by a dashed line. The TFC controller is the one provided in [7], [8], based on the inversion of the Aquino's model. We consider the TFC as the baseline control. For sake of correctness, we should have inserted our controller in the ECM and compared its real performance with the baseline TFC tailored to the engine. Since our study was to test the feasibility of the neural approach and not to substitute the controller present in the ECM, we were not allowed to realize such an experiment. Despite this, it is still possible to test the validity of the approach by assuming that the simplified model describes the real engine; this is correct since both TFC and the neural controller have been configured on such a model. Once validated, the neural controller could be inserted in the ECM and tuned to the specific engine by slightly modifying its weights with a simple gradient descent algorithm. In the following, the inputs for the experiments have been extracted from the real engine, while the output must be intended as above.

In the plots, we can appreciate the null error at the steady state (as forced by the modified training loss function) and the significant reduction in AF. Peaks in Fig. 10(a) and (b) correspond to the transient of α. We note that the suggested controller provides an effective and prompt control action, which drives AF toward AF_S.

To compare the capabilities of the neural controller with those provided by the TFC, we evaluated some indexes for each experiment; comparisons are summarized in Table III. We computed the percentage of values greater than 3% for the two controllers and the mean of the percentage of time that the values are outside the critical phase. In the worst case, the neural controller is outside the confidence region for at most about 1% in average and for about 9% in only few points. As the control action is concerned, the improvement in performance was obtained with an action comparable with that requested by the TFC: this is a consequence of the penalty terms introduced in the training error function. To compare the control actions, we refer to the second experiment. Fig. 11 shows the evolution over time of t_j as provided by the neural controller (continuous line) and the TFC (dashed line).

VI. CONCLUSION

The paper presents an application of neural techniques to the automotive field. Design of a neural controller of the AF ratio has been proposed which aims at minimizing the exhaust emissions in fuel injection engines. To improve the equation-based description of the process, we considered feedforward neural networks to model some relevant nonlinear parameters describing the fuel film dynamics. Since the reduced number of data prevented the use of cross validation techniques, we selected the best neural model with the FPEB criterion. The neural controller was then obtained by referring to an indirect control scheme, which required a preliminary identification of the process. The final neural controller has been designed to optimize performance, limits the necessary control actions, and allows for an on-line training. Encouraging results have been obtained on data coming from a real engine.

ACKNOWLEDGMENT

The authors wish to thank N. Dell'Oro, F. Savi, and L. Sala, for their support in the experimental phase.

REFERENCES

[1] T. Y. Chang *et al.*, "Urban and regional ozone air quality: Issues relevant to automobile industry," *Crit. Rev. Environ. Control*, vol. 22, no. 1–2, pp. 27–66, 1992.
[2] W. W. Yuen and H. Servati, "A Mathematical Engine Model Including the Effect of Engine Emissions," Rep., SAE TP 840 036, 1984.
[3] *Carisma 1.8 GDI LX Focus*, 1997–1998.
[4] H. Shirashi, S. L. Ipri, and D. D. Cho, "CMAC neural controller for fuel-injection systems," *IEEE Trans. Contr. Syst. Technol.*, vol. 3, pp. 32–38, Mar. 1995.
[5] J. B. Heywood, *Internal Combustion Engine Fundamentals*. New York: McGraw–Hill, 1988.
[6] C. D. Falck and J. J. Money, "Three-Way Conversion Catalyst: Effect of Closed-Loop Feedback Control and Other Parameters on Catalyst Efficiency," Report, SAE TP 800 462, 1980.
[7] C. F. Aquino, "Transient A/F Control Characteristics of the 5 Liters Central Fuel Injection Engine," Rep., SAE TP 810 494, 1981.
[8] E. Hendricks and T. Vesterholm, "Non-Linear Transient Fuel Film Compensation," Rep., SAE TP 930 767, 1993.
[9] S. D. Hires and M. T. Overington, "Transient Mixture Strength Excursion: An Investigation on Their Causes and the Development of a Constant Mixture Strength Fuelling Strategy," Rep., SAE TP 810 495, 1981.
[10] P. E. Moraal, "Adaptive compensation of fuel dynamics in a SI engine using a switching EGO sensor," presented at the 34th Conf. Decision Control, New Orleans, LA, Dec. 1995.

[11] L. I. Davis Jr., G. V. Puskorius, F. Yuan, and L. A. Feldkamp, "Neural network modeling and control of an anti-lock brake system," in *Proc. Intelligent Vehicles'92 Symp.*, Detroit, MI, 1992, pp. 179–184.
[12] G. V. Puskorius and L. A. Feldkamp, "Automotive engine idle speed control with recurrent neural networks," in *Proc. 1993 American Control Conf.*, San Francisco, CA, June 1993, pp. 311–316.
[13] M. Majors, J. Stori, and D. I. Cho, "Neural network control of automotive fuel-injection systems," *IEEE Contr. Syst. Technol. Mag.*, vol. 14, pp. 31–36, June 1994.
[14] P. J. Shayler and M. S. Goodman, "Transient Air/Fuel Ratio Control of an S. I. Engine Using Neural Networks," Rep., SAE TP no. 960 326, 1996.
[15] R. C. Turin and H. P. Geering, "On-Line Identification of Air-Fuel Ratio Dynamics in a Sequentially Injected SI Engine,", SAE TP 930 857, 1993.
[16] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4–27, Jan. 1990.
[17] K. S. Narendra and A. U. Levin, "Control of nonlinear dynamical systems using neural networks: Controllability and stabilization," *IEEE Trans. Neural Networks*, vol. 4, pp. 192–206, Mar. 1993.
[18] G. V. Puskorius, L. A. Feldkamp, and L. I. Davis, Trained Neural Network Air/Fuel Control Systems, U.S. Patent 5 781 700, 1998.
[19] C. Alippi and V. Piuri, "Experimenting neural networks for prediction and identification," *IEEE Trans. Instrum. Meas.*, vol. 45, pp. 670–676, Apr. 1996.
[20] ——, "Topological minimization of multilayered feed-forward neural networks by spectral decomposition," *Proc. 1992 IEEE Int. Joint Conf. Neural Networks*, vol. 2, pp. 805–810, Nov. 1992.
[21] Y. Le Cun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems*. San Mateo, CA: Morgan Kaufmann, 1989, vol. 2, pp. 598–606.
[22] B. Hassibi and D. G. Stork, "Second order derivative for network pruning: Optimal brain surgeon," in *Advances in Neural Information Processing Systems*. San Mateo, CA: Morgan Kaufmann, 1992, vol. 5, pp. 164–171.
[23] S. J. Nowlan and G. E. Hinton, "Simplifying neural networks by soft weight-sharing," *Neural Comput.*, vol. 4, no. 4, pp. 473–493, 1992.
[24] C. Wang, S. S. Venkatesh, and J. S. Judd, "Optimal stopping and effective machine complexity in learning," in *Advances in Neural Information Processing Systems*. San Mateo, CA: Morgan Kaufmann, 1993, vol. 6, pp. 303–310.
[25] A. S. Weigend, D. E. Rumelhart, and B. A. Huberman, "Generalization by weight elimination with application to forecasting," in *Advances in Neural Information Processing Systems*. San Mateo, CA: Morgan Kaufmann, 1990, vol. 3, pp. 875–882.
[26] N. Murata, S. Yoshizawa, and S. Amari, "Network information criterion—Determining the number of hidden units for an artificial neural network model," *IEEE Trans. Neural Networks*, vol. 5, pp. 865–872, Nov. 1994.
[27] C. Alippi, "FPE-Based Criteria to Dimension Feedforward Neural Topologies, IEEE-TCAS1," National Research Council, vol. 46, Internal Rep., Aug. 1999.
[28] J. Moody, "The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems," in *Advances in Neural Information Processing Systems*. San Mateo, CA: Morgan Kaufmann, 1992, vol. 4, pp. 847–854.
[29] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes*. Cambridge, MA: Cambridge Univ. Press, 1986.
[30] C. Alippi and V. Piuri, "The use of neural technologies for prediction and identification of nonlinear dynamic systems," *Proc. 1996 IEEE Int. Workshop Emergent Technologies Instrumentation Measurement*, pp. 1–9, June 1996.
[31] *IEEE Trans. Neural Networks*, Mar. 1994.
[32] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, 1989.
[33] B. Pearlmutter, "Gradient calculations for dynamic recurrent neural networks," in *A Field Guide to Dynamical Recurrent Networks*, J. F. Kolen and S. C. Kremer, Eds. New York: Wiley, 2001.

Cesare Alippi (SM'99) received the Dr.Ing. degree in electronic engineering (summacumlaude) in 1990 and the Ph.D. degree in computer engineering in 1995, both from Politecnico di Milano, Milano, Italy.

He is a Full Professor in information processing systems at the Politecnico di Milano. His further education includes research work in computer sciences carried out at the University College London, London, U.K., and the Massachusetts Institute of Technology, Cambridge. His interests include neural networks (learning theories, implementation issues and applications), composite systems and high level analysis and design methodologies for embedded systems. His research results have been published in more than 100 technical papers in international journals and conference proceedings.

Cosimo de Russis was born in Turin, Italy, in 1966. He received the B.E.E. degree in electronic engineering from the Politecnico di Torino, Torino, Italy in 1990. From 1991 to 1995, he continued his education and researches on control systems, both in the USA and Italy at the Department of Automatic Controls, FIAT Research Center, Orbassano, Italy.

Since 1998, he has been an Associate Director at Altran, the international technological consultancy group. He is now managing one of its subsidiaries, based in Rome, Italy, specialized in engineering consultancy for telecom, automotive, and avionic sectors. His interest was mainly on robust control theory application in robotics and automotive field. Some of his results have been patented or have been published in journals and conference proceedings.

Vincenzo Piuri (F'01) received the Ph.D. degree in computer engineering in 1989 from Politecnico di Milano, Milano, Italy. He is Associate Editor for the Journal of Systems Architecture.

Since October 2000, he has been a Full Professor of computer engineering at the University of Milano. His research interests include distributed and parallel computing systems, computer arithmetic, application-specific processing architectures, digital signal processing architectures, fault tolerance, neural network architectures, theory and industrial applications of neural techniques for identification, prediction, control, signal and image processing. Original results have been published in more than 150 papers in book chapters, international journals, and proceedings of international conferences.

Dr. Piuri is a member of ACM, IMACS, INNS, and AEI. He is Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS. He is Vice President for Publications of the IEEE Instrumentation and Measurement Society and Vice President for Member Activities of the IEEE Neural Networks Society. He is member of the IMACS Technical Committee on Neural Networks.