Original Research Article

Robust predictive control of lambda in internal combustion engines using neural networks

T. Sardarmehni\textsuperscript{a,}\textsuperscript{*}, J. Keighobadi\textsuperscript{a}, M.B. Menhaj\textsuperscript{b}, H. Rahmani\textsuperscript{a}

\textsuperscript{a}Faculty of Mechanical Engineering, University of Tabriz, Tabriz, Iran
\textsuperscript{b}Department of Electrical Engineering, Amirkabir University of Technology, Hafez Ave., Hafez Street, PO Box 15875-4413, Tehran, Iran

\textbf{A B S T R A C T}

Stoichiometric air-to-fuel ratio (lambda) control plays a significant role on the performance of three way catalysts in the reduction of exhaust pollutants of Internal Combustion Engines (ICEs). The classic controllers, such as PI systems, could not result in robust control of lambda against exogenous disturbances and modeling uncertainties. Therefore, a Model Predictive Control (MPC) system is designed for robust control of lambda. As an accurate and control oriented model, a mean value model of a Spark Ignition (SI) engine is developed to generate simulation data of the engine’s subsystems. Based on the simulation data, two neural networks models of the engine are generated. The identified Multi-Layer Perceptron (MLP) neural network model yields small verification error compared with that of the adaptive Radial Base Function (RBF) neural network model. Consequently, based on the MLP engine’s model, the MPC system is performed through a nonlinear constrained optimization within gradient descent algorithm. The performance of the MPC system is compared with that of a first order Sliding Mode Control (SMC) system. According to simulation results, the tracking accuracy of lambda by the MPC system is close to that of the SMC system. However, the MPC system results in considerably smoother injected fuel signal.

\section{1. Introduction}

Internal combustion engines are one of the major causes of air pollution. National and international legislations, like On-Board Diagnostics II (OBD II), confine automotive industry to precise-predefined and strict eligible amount of emission pollutants due to their probable carcinogenic and destructive effects on human tissues [1]. Mostly, pollutant emissions of ICEs, in particular NO\textsubscript{x}, HC and CO, are cancerous gases which can also cause heart diseases and respiratory disorders. Fortunately, the amounts of these pollutants could be considerably decreased by use of a three way catalyst converter.
As the main concern, the catalyst converter functions effectively at its optimum state in which lambda should be controlled within \( \pm 1\% \) of the stoichiometric value \([1]\). PI-controllers, which are used in automotive industry for controlling lambda, require a long time calibration process as the state of the art in the control oriented models of lambda.\([2]\).

Nowadays, regarding the advancements in microcomputer programming, the intelligent and adaptive control algorithms of lambda could be implemented in Electronic Control Units (ECU) of engines. Besides the significant transient and steady-state performances, these controllers are adjusted simply and also are robust to the uncertainties \([2]\).

As the state of the art in the control oriented models of ICES, the mean value models of SI engines are developed. In the Mean Value Engine Model (MVEM), the dynamical equation for every state variable is fitted to the mean value of the corresponding state. In this way, the simulated dynamics of a real engine results in the state values which are as accurate as a typical dynamometer measurements in the steady-state mode, and yields \( \pm 2\% \) Mean Absolute Error (MAE) in the transient mode \([3]\). In the beginning, the MVEMs were not compatible to be applied to new SI engines because of the difficulties in modeling of volumetric efficiency of the engines. Hendricks et al. have improved the MVEM by introducing a simple model of the volumetric efficiency and therefore, the improved model is compatible for application in a wide range of SI engines \([4]\).

In the first developed neural network MPC of lambda by Wang and Yu in 2005, an adaptive MLP neural network system with both online and off-line back propagation and recursive least square training algorithms has been considered. Compared with the off-line training method, the online algorithm led to a smaller identification error in the sense of MAE. Since the neural network models have been trained for a bounded range of input variables, the predicted control effort of the MPC is obtained through constrained optimization algorithms. By assuming the injected fuel as a fixed value throughout the prediction horizon, a constrained optimization based on the reduced Hessian method could be performed. However, a heavy computation and quadratic programming is imposed on the MPC algorithm \([5]\). Wang et al. has introduced a second order RBF neural network model for identification and thereby the MPC of SI engines. Using the preceding reduced Hessian method for constrained optimization and under the constant fuel assumption throughout the prediction horizon, the RBF based MPC showed a high robustness to modeling uncertainties and also a better transient performance compared with the PI-controller \([6]\). Zhai and Yu have used the adaptive second order RBF neural network and have applied the recursive least square algorithm for online training of the RBF model. The optimization of the introduced cost function has been performed by both Secant and reduced Hessian methods under the fixed transient and steady-state mode in which lambda should be controlled within \( \pm 1\% \) of the stoichiometric value \([1]\).

### Nomenclature

- \( m_i \): air mass in intake manifold [kg]
- \( \dot{m}_{air} \): air mass flow passing the throttle plate [kg/s]
- \( \dot{m}_{lep} \): air mass flow entering the combustion chamber [kg/s]
- \( P_i \): manifold air pressure [bar]
- \( V_i \): intake manifold and port passage volume [m\(^3\)]
- \( T_i \): intake manifold temperature [K]
- \( R \): gas global constant [kJ/kg K]
- \( V_d \): engine displacement volume [l]
- \( \rho_i \): density of air [kg/m\(^3\)]
- \( \eta_v \): volumetric efficiency
- \( s_i \): slope of the normalized air charge
- \( y_i \): y-intercept of the normalized air charge
- \( T_a \): ambient temperature [K]
- \( \alpha \): throttle angle [deg]
- \( P_f \): engine mechanical friction loss power [kW]
- \( P_p \): engine pumping loss power [kW]
- \( n \): crankshaft speed [rpm/1000 or krpm]
- \( I \): total engine’s moment of inertia \([\text{kg} \times \text{m}^2 / (\text{rpm}^2 / 3000)^2] \)
- \( H_a \): fuel heating value [kJ/kg]
- \( \eta_i \): indicated efficiency
- \( \tau_{inj} \): time delay for the injected fuel to make power [s]
- \( \dot{m}_{inj} \): injected fuel mass flow rate [kg/s]
- \( X_f \): ratio of injected fuel deposits in the intake manifold
- \( \dot{m}_f \): fuel flow into combustion chamber [kg/s]
- \( \dot{m}_{ff} \): fuel film mass flow [kg/s]
- \( \dot{m}_{v} \): fuel vapor mass flow [kg/s]
- \( \tau_f \): fuel film evaporation time constant [s]
- \( \lambda_{sensor} \): measured lambda by lambda sensor
- \( \tau_e \): time delay in measurements of lambda sensor [ms or \( 10^{-3}\) s]
- \( \lambda_m \): lambda which is directly calculated from the MVEM
- \( \lambda_d \): desired value of lambda equals to 1
- \( AFR_{st} \): stoichiometric value of air-to-fuel ratio equals 14.86
- \( \delta \): total delay in lambda sensor measurements [ms or \( 10^{-3}\) s]
- \( \theta_{EVC} \): angle of crankshaft at which exhaust valve opens [deg]
- \( n_{cyl} \): number of engine’s cylinders
- \( \tau_d \): time constant delay for the exhaust gas to reach the lambda sensor [ms or \( 10^{-3}\) s]
- \( \beta \): tuning parameter in MLP network’s activation function
- \( \eta \): learning rate of back propagation and gradient descent algorithm
- \( t \): time [s]
- \( l \): number of in process training pattern
- \( k \): iteration number in training of neural network and optimization.
assumptions for the injected fuel throughout the prediction horizon. The simulation results showed that the Secant method is more time efficient than the reduced Hessian method [2]. Zhai et al. have developed a diagonal recurrent neural network model for MPC of air-to-fuel ratio. The presented neural network model was adapted on-line to deal with the engine’s time-varying dynamics. Furthermore, the constrained optimization algorithm was carried out through “fmincon” function of MATLAB software. As the main result, the robust performance of the control system has been improved compared to that of the PI-controller [7].

As a model-based control system for SI engines, the SMC systems are preferred due to their rapidity and the ease of their application on the nonlinear model of the engine. Besides the possibility of fast implementation, the robustness and the stability of SMC could be guaranteed in most applications [8]. In the first documented SMCs applied to the experimental control of lambda by Kaidantzis et al. and Carnevale et al., the performance of the SMC has been compared to that of the PI-controller. During the experiments of SMC, the variation of lambda has been maintained within ±2% of the stoichiometric value; however, the PI-controller could only regulate lambda within ±12% of the stoichiometric value [9,10]. In a dynamic SMC system, two second order RBF neural networks were designed for online estimation of time-varying gain parameters of the SMC system. Although analytical methods could be used to obtain the gain parameters of the SMC system, see for example [4], the RBF neural network based SMC system has resulted in less chattering error and more robustness against uncertainties [11].

In this paper, the MVEM and a first-order linear model of lambda sensor (oxygen sensor) are used to generate simulation data of the SI engine. Using the simulation data, two different adaptive RBF and MLP neural network models are trained. In order to decrease the neural networks’ identification error, data collection sample time for neural network training is selected to be smaller than what has been recorded in recent research documents [2,5–7]. The adaptive RBF network’s computation and training time is more than that of the MLP model. Furthermore, the identification error of the RBF model is larger than that of the MLP model. Consequently, the MLP neural network model is preferred rather than the RBF kind as the base of the MPC. The identification error of the MLP neural network model is considerably smaller than what has been recorded in the state of the art works published recently [2,5–7]. Since the smaller sample time for data collection makes the entire feedback control system slow, the prediction horizon is shortened in the MPC to accelerate computations. In the proposed constrained optimization method, the simple and fast gradient descent method is used instead of the preceding Secant or reduced Hessian methods. In this way, the slow functioning of the MPC due to reduction of the data collection sample time is compensated. Application of neural network methods in MPC of lambda in internal combustion engines has been introduced in [2,5–7]. In the referenced documents, due to complex structures of proposed neural network models as well as the optimization process of predictive control methods, a considerable computational burden suffers online identification and control of the engine lambda system. However, in this research work, by decreasing the computational burdens the performance of the proposed online control system has been improved and the convergence rate of the optimization algorithm is increased. Furthermore, as shown by simulation results, the accuracy and the convergence rate of the proposed identification algorithm for off-line training of designed neural network model are significantly superior with respect to the recently documented results in [2,5–7]. Owing to the optimization of MPC based on the gradient descent method, a faster tracking performance is obtained compared with those reported in [2,5,6]. As a result, the proposed method is more suitable for real time applications.

The robustness of the designed MPC system against modeling uncertainty is evaluated by imposing rough throttle plate angles with 10% and 25% Gaussian noises. The simulation results show that the introduced MPC system is satisfactorily robust to modeling uncertainties. For a better assessment, the overall performance of the proposed MPC system is compared with that of a first order SMC system. The simulation results show that the designed predictive controller has a smoother injected fuel signal than the SMC system. On the other hand, though there is a better tracking performance, the control input of the sliding mode controller includes large oscillations whose production is difficult for implementation. The entire program is running in MATLAB/simulink software.

2. Mean Value Engine Model (MVEM)

As shown in Fig. 1, MVEM in its basic form consisted of three sub-systems which are intake manifold air mass flow, crankshaft/loading and fuel vapor/film sub-system. State variables are fuel mass flow deposited in the intake manifold, manifold pressure and crankshaft speed [3]. The measured lambda by sensor is considered as the 4th state of the MVEM. The inputs in the MVEM are injected fuel mass flow, throttle plate angle and ignition advance. The ignition advance is supposed to be

![Fig. 1 – Block diagram of mean value model of engine.](image-url)
a fixed value and the throttle plate angle is regarded as a noise signal. Therefore, the control commands could be imposed on the engine merely through the injected fuel mass flow.

2.1. Intake manifold air mass flow sub-system

The first state equation is obtained from applying conservation of mass on the intake manifold which is considered as a control volume.

\[
\frac{d}{dt}m_i = \dot{m}_{at} - \dot{m}_{ap}
\]

(1)

Assuming air as an ideal gas gives

\[
m_i = \frac{P_i V_i}{RT_i}
\]

(2)

Substitution of (2) in (1) results in the first state equation as

\[
P_i = \frac{RT_i (m_{at} - \dot{m}_{ap})}{V_i}
\]

(3)

in (3), \( \dot{m}_{ap} \) is a function of crankshaft speed and manifold pressure. Hendricks has obtained the following modified dynamics for \( \dot{m}_{ap} \) [3]:

\[
\dot{m}_{ap} = \frac{1}{2} \frac{n}{60} V_d p_i
\]

(4)

Assuming air as an ideal gas, one has

\[
\dot{\rho} = \frac{P_i}{RT_i}
\]

(5)

Replacing (5) in (4) generates the normalized engine air charge \( P_i q_i \) which is defined as follows.

\[
P_i q_i = \frac{s_i(n) P_i - y_i(n)}{\delta}
\]

(6)

where \( s_i \) and \( y_i \) are considered as the constants for typical 4-stroke SI engines according to [4]. \( m_{at} \) is a function of the throttle plate angle and the manifold pressure which can be defined as follows.

\[
m_{at}(\alpha, P_t) = m_{at1} \frac{P_a}{P_t} \beta_1(\alpha) \beta_2(P_t) + m_{at0}
\]

(7)

where \( m_{at1} \) and \( m_{at0} \) are the constants to be defined. The intermediate terms of \( \beta_1 \) and \( \beta_2 \) are considered as

\[
\beta_1(\alpha) = 1 - \cos(\alpha) - \frac{\alpha^2}{2}
\]

(8)

\[
P_t = \frac{P_t}{P_a}
\]

(9)

\[
\beta_2(P_t) = \begin{cases} \sqrt{1 - \left(\frac{P_t - P_c}{1 - P_c}\right)^2} & \text{if } P_t \geq P_c \\ 1 & \text{if } P_t < P_c \\ \end{cases}
\]

(10)

In (8), \( \alpha_0 \) is the closed throttle plate angle and \( P_c \) is a constant scalar in (10).

2.2. Crankshaft and loading sub-system

The second state space equation is obtained through the energy conservation principle for the crankshaft rotation.

\[
\dot{n} = \frac{-1}{\ln(n) P_r(n) + P_c(n) + P_s(n)) + \frac{1}{\ln H_0/\ln(n, P_r, \lambda)} \mu_i(t - \tau_d)}
\]

(11)

2.3. Fuel vapor and fuel film sub-system

From all the injected fuel in the intake manifold, a \( X_f \) ratio deposits as liquid in the intake manifold after reaching the cold air in the manifold, and \( 1 - X_f \) ratio remains as fuel vapor. The condensed fuel film gradually vaporizes to be mixed with the fuel vapor and makes \( \dot{m}_f \) to enter the combustion chamber. Accordingly, the dynamics of the fuel sub-system could be summarized as

\[
\dot{m}_{ff} = \frac{1}{s_i} (\dot{m}_{ff} + X_f \dot{m}_f)
\]

(12)

\[
\dot{m}_f = (1 - X_f) \dot{m}_f
\]

(13)

\[
\dot{m}_f = \dot{m}_f + \dot{m}_{ff}
\]

(14)

2.4. Lambda sensor model

The 4th state equation determines the dynamics of a first order lambda sensor which is summarized as

\[
\lambda_{sensor} = \lambda_{sensor} - \lambda_m(t - \tau)
\]

(15)

where \( \lambda_{sensor} \) is the measured lambda by lambda sensor; and \( \tau \) defines the time delay in the measurements made by the sensor. The computed lambda from the MVEM, \( \lambda_m \) is defined as

\[
\lambda_m = \frac{\dot{m}_{ap}}{AFR_{est} \dot{m}_f}
\]

(16)

In (15), \( \delta \) is the total delay of the lambda sensor defined as

\[
\delta = \left( \frac{\lambda_{EVD}}{220} \right) \left( \frac{120}{\lambda_{Sensor}} \right) + \delta_d
\]

(17)

3. Multilayer perceptron (MLP) modeling of the SI engine

3.1. Structure of MLP neural networks

Block diagram of a multilayer perceptron neural network is shown in Fig. 2. The three layers MLP neural network model

![Fig. 2 - Block diagram of MLP neural networks.](image-url)
of the engine comprises of the input layer, hidden layer and output layer. The first layer receives the input data and distributes them to all the nodes of the hidden layer. The nonlinear activation function, \( f^1 \) in every neuron of the hidden layer receives the weighted sum of the all input data plus the bias term of corresponding node. In the output layer, the weighted outputs of the hidden layer linearly sum up to the output layer's bias and make the output of network.

Assuming the number of nodes in the input layer as \( p \), the number of neurons in the hidden layer as \( q \) and the dimension of the network's output as 1, the attributes of the neural network model become as

\[
W_1 \in \mathbb{R}^{q \times p}, b_1 \in \mathbb{R}^{q \times 1}, net_1 \in \mathbb{R}^{q \times 1}, a_1 \in \mathbb{R}^{q \times 1}, W_2 \in \mathbb{R}^{1 \times q} \text{ and } b_2 \in \mathbb{R}^{1 \times 1}.
\]

The activation function \( f^1 \) is a bipolar function as

\[
f^1(x) = \frac{2}{1 + \exp(-\beta x)} - 1 \quad (18)
\]

where \( 0 < /\beta < 1 \) is a tuning parameter and could be adjusted by experience or trial and error. Now, the weighted input vector, \( P \), is gathered with the neuron's bias term to obtain \( net_1 \) as

\[
net_1 = W_1 P + b_1 \quad (19)
\]

Or in more details

\[
net_1 = \sum_{j=1}^{p} (W_1(i,j)P(j,1) + b_1(i,1)) \quad i = 1, \ldots, q; \ j = 1, \ldots, p \quad (20)
\]

Therefore, the output of the hidden layer, \( a_1 \) is obtained as

\[
a_1 = f^1(net_1) \quad (21)
\]

The network’s overall output can now be obtained as follows [12].

\[
y = W_2 a_1 + b_2 = \sum_{j=1}^{q} (W_2(j,1) \times a_1(j,1)) + b_2 \quad (22)
\]

### 3.2. Training algorithm of the MLP network

The parameters of MLP model, including \( b_1, b_2, W_1 \) and \( W_2 \), are updated by the back propagation algorithm based on the following error, \( e_i \).

\[
e_i = y - y_d \quad (23)
\]

where \( y \) and \( y_d \) are the MLP’s and desired outputs, respectively. The total error can now be defined as

\[
E = \frac{1}{2} \sum_{l=1}^{Q} (y(l) - y_d(l))^2 \quad (24)
\]

where \( Q \) is the total number of training patterns and \( l \) defines the in process training pattern. Based on the back propagation algorithm, the bias term of (22) is updated as

\[
b_2(k+1) = b_2(k) - \eta \frac{\partial E(k)}{\partial b_2(k)} \quad (25)
\]

**Fig. 3** – (a) Normalized injected fuel used for training and (b) normalized throttle plate angle used for training.
where $\eta$ is a small positive number called the learning rate. More precisely, it can be shown

$$b^2(k+1) = b^2(k) - \eta \sum_{l=1}^{Q} \frac{\partial E(k)}{\partial y(l)} \frac{\partial y(l)}{\partial b^2(k)} = b^2(k)$$

Moreover, $W^2$ is updated by the back propagation with the same procedure \(25\) as follows.

$$W^2(k+1) = W^2(k) - \eta \frac{\partial E(k)}{\partial W^2(k)}$$ \(26\)

In complete details, \(27\) is represented as

$$W^2(k+1) = W^2(k) - \eta \sum_{l=1}^{Q} \frac{\partial E(k)}{\partial y(l)} \frac{\partial y(l)}{\partial b^2(k)} \frac{\partial b^2(k)}{\partial W^2(k)} = b^2(k) - \eta \sum_{l=1}^{Q} e_l(l) f'(W^1l + b^1)$$ \(28\)

In every iteration $k$, $b^1$ is updated as follows.

$$b^1(k+1) = b^1(k) - \eta \frac{\partial E(k)}{\partial b^1(k)}$$ \(29\)

Or in more details as

---

**Fig. 4** – (a) Normalized injected fuel used for validation, (b) normalized throttle plate angle used for validation and (c) validation data $MAE = 0.0025$. 

---
thereby the neural network parameters are speci-
work system. As Fig. 3a and b shows, the training patterns of
have a wide pulse width, i.e. 1.5 s. By imposing the input
patterns of the recent data are required for training the
Training is done by the back propagation algorithm described
Corresponding to
and the injected fuel mass
(normalized into the interval [0, 1]. A fixed 0.01 s sample time
is considered 2 and the learning rate in the back propagation
algorithm for the first 500,000 iterations is 0.01 and for the
iterations 500,000 through 1,000,000 is 0.001. After one million
iterations, the accuracy of the trained MLP model is evaluated
by two new sets of 10,000 bounded random samples for
the throttle plate angle and the injected fuel mass flow. The resulted 0.0025 Mean Absolute Error (MAE) corresponding
to the evaluation data of Fig. 4 is significantly smaller than what has been recorded in [2,5–7].

The converged weight and biases of the MLP neural net-
work model of the engine are represented in Table 1.

3.3. The MLP model training and validation data
Training is done by the back propagation algorithm described in Section 3.2. Since MVEM is a second order system, two past samples of the recent data are required for training the engine model. Considering \( \dot{\mathbf{i}}(t) \) as the output of the MLP network at the current sample time, \( t \), the input vector of the training process consisted of the fuel mass flow and the lambda at \((t-2)\) and \((t-1)\) as

\[
\mathbf{P} = [m_f(t), m_f(t-1), \dot{\lambda}(t-1), \lambda(t-2)]^T
\]

Through widespread simulation data of the MLP engine model, 4 neurons are designed in the hidden layer and thereby the neural network parameters are specified as \( \mathbf{W}^{1,2,3} \in \mathbb{R}^{2 \times 4} \), \( \mathbf{b}^{1,2,3} \in \mathbb{R}^{4 \times 1} \), \( \mathbf{W}^{2,3} \in \mathbb{R}^{1 \times 4} \) and \( \mathbf{b}^{2,3} \in \mathbb{R}^{1 \times 1} \). Corresponding to the two inputs of the MVEM, two 1000-tuple random signals with bounded amplitude are imposed to the MVEM. The throttle plate as the first input is bounded to the interval 20–40 (deg) and the 2nd input, the injected fuel mass flow, is bounded to the interval 0.001–0.0079 (kg/s).

To consider both the transient and the steady state performance of the MVEM, one half of the training patterns are generated with a pulse width of 0.1 s and the other half have a wide pulse width, i.e. 1.5 s. By imposing the input training data on the MVEM, 1000 output values for lambda are generated which are used in training process of neural network system. As Fig. 3a and b shows, the training patterns of
the injected fuel mass flow and the throttle plate angle are normalized into the interval [0, 1]. A fixed 0.01 s sample time is used in the generation of the training data. \( \beta \) in (18) is considered 2 and the learning rate in the back propagation algorithm for the first 500,000 iterations is 0.01 and for the iterations 500,000 through 1,000,000 is 0.001. After one million iterations, the accuracy of the trained MLP model is evaluated by two new sets of 10,000 bounded random samples for
the throttle plate angle and the injected fuel mass flow. The resulted 0.0025 Mean Absolute Error (MAE) corresponding
to the evaluation data of Fig. 4 is significantly smaller than what has been recorded in [2,5–7].

The converged weight and biases of the MLP neural net-
work model of the engine are represented in Table 1.

4. Radial Base Function Neural network (RBFN)
4.1. Structure of RBFN
In the three layer RBFN model of the engine, the first layer simply receives the input data and transfers it to the second/ hidden layer. The hidden layer consisted of several clusters with individual centers. The following Gaussian activation function is considered in the designed clusters.

\[
h_i(x) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2\sigma^2}(x-c_i)^2}, \quad i = 1, \ldots, q
\]

where \( c_i \) is the center for each cluster; the positive scalar, \( \sigma \), is called the width of the activation function and defines a range over which, the Gaussian function has a significant output [2]; and \( q \) is the number of clusters in the hidden layer. Through the third/output layer, the weighted outputs of the hidden layer are linearly summed up to form the final output of the RBFN as

\[
y = \mathbf{W} \times \mathbf{h}(x)
\]
weight matrix of the output layer is updated as follows [13].

$$\mathbf{W}_i(k+1) = \mathbf{W}_i(k) - \eta \frac{\partial E(k)}{\partial \mathbf{W}_i(k)}$$  \hspace{1cm} (38)

where \(i=1,\ldots,q\) and \(j=1,\ldots,l\) show the number of clusters and network’s outputs, respectively. \(\eta\) is the learning rate and \(k\) is the iteration number in the training process. Eq. (38) is simplified as

$$\frac{\partial E(k)}{\partial \mathbf{W}_i(k)} = \sum_{i=1}^{Q} e_i \mathbf{h}_i(k)$$ \hspace{1cm} (39)

Using (39) in (38) completes the training algorithm.

4.3. Validation data for the RBFN model of the engine

The same training patterns used in the MLP system are considered for training the RBFN system. Through widespread simulation results, 14 clusters are determined in the hidden layer. The center of each cluster is defined by K-means clustering algorithm. For the widths of Gaussian functions, \(\rho\)-nearest neighbor method for 5 nearest neighbors is used. Afterwards, the back propagation algorithm with \(\eta=0.001\) is used for updating the weight matrix of the output layer. Using the same validation data of the MLP system in the RBFN system yields \(\text{MAE}=0.0051\). Although this tracking error is significantly less than what has been recorded in the recent documented research works, the tracking error is still considerably more than that of the MLP system. Therefore, the MLP neural network model is preferred in the identification of the engine.

5. Model Predictive Control (MPC)

The schematic of the proposed MPC system is shown in Fig. 5. In each sample time, the system predicts \(N_u\) number of constrained control inputs for considering in the designed cost function of the MPC system.

In Fig. 5, \(\hat{m}_f(t_0)\) and \(\hat{\lambda}(t_0)\) show the injected fuel mass flow and lambda at time, \(t_0\), respectively. Therefore, in the time interval \(t_0\) through \(t_{0+\Delta t}\), all the components of the following vector, \(\mathbf{P}_0\), are known for \(i=0\).

\[
\mathbf{P}_0 = [\hat{m}_f(t_0 + i\Delta t), \hat{m}_f(t_0 + (i-1)\Delta t), \hat{\lambda}(t_0 + (i-1)\Delta t)]
\]

As shown in Fig. 5, the first component of \(\mathbf{P}_1, U_1\) is used as the control input. Similarly, the remaining \(U_s\) are considered as the initial values for the future time steps. The online gradient descent algorithm results in optimized values of all \(U_s\) in each sample time. Therefore, using the preceding optimized values as the initial values in the current sample time will result in fast convergence of the MPC algorithm.

5.1. Optimization

According to gradient descent algorithm, the under optimization vector, \(\mathbf{U}=[U_1, U_2, \ldots, U_{Nu}]\) is updated as

\[
U(k+1) = U(k) - \eta \frac{\partial f(k)}{\partial U(k)} = U(k)
\]

In the MPC system, the vector \(\mathbf{P}_i\) (40) is given to the neural network model of the engine for predicting lambda. Therefore, in the interval, \(t_0\) to \(t_{0+\Delta t}\), MPC requires the predicted values of \(\hat{m}_f\) and \(\hat{\lambda}\) at time samples \(t_0+i\Delta t\) \((i=0,\ldots,N_u)\). The values of \(\hat{\lambda}(t_0+i\Delta t)\) for each \(P_i\) in Fig. 5 are predicted through the trained MLP model of the engine using \(\mathbf{P}_{i-1}\) for \(i=1,\ldots,N_u\). The optimal values of \(\hat{m}_f(t_0+i\Delta t)\) are obtained through minimization of the following MPC cost function based on gradient descent method.

\[
J = \frac{1}{2} \sum_{i=1}^{N_u} (\hat{\lambda}(i)-\lambda_d(i))^2
\]

where \(\lambda_d\) is the predicted lambda at time \(t_0+(i+1)\Delta t\). \(\lambda_d\) is ideal or stoichiometric value of lambda and it equals 1 which is the desired value for lambda. Regarding Eq. (16), one can see that the value of lambda as the normalized air-to-fuel ratio tends to 1 whenever the combustion process has the same air-to-fuel ratio of an ideal combustion. In stoichiometric combustion of a typical SI-engine, the value of air-to-fuel ratio is 14.86 which is shown by AFRst in (16). The value of AFRst is used to normalize the air-to-fuel ratio of real engines which simply shows lambda. The goal of the whole control system is the regulation of injected fuel such that lambda would be controlled within \(\pm 1\%\) of its stoichiometric value 1. In this case, as mentioned before, the amount of pollutant emissions leaving the exhaust pipe can be considerably reduced by the catalyst converter.

Fig. 5 – Schematic of MPC system.
where $\eta$ is the learning rate and $k$ shows the iteration number. The partial derivative in the last term of (42) is computed as follows.

$$
\frac{\partial \hat{d}(i)}{\partial U(k)} = \frac{\partial \hat{d}(i)}{\partial P_{f}(j, 1)} = \sum_{j=1}^{q} (W_{j}^{f}(\text{net}_{j})) W_{j}(i, i);
$$

where $i = 1, \ldots, q$; $j = 1, \ldots, p$

As Fig. 5 shows, $U_{1}$ directly affects the prediction of $\hat{d}(t_{0}+\Delta t)$ and $\hat{d}(t_{0}+2\Delta t)$. Besides, due to the effect of $\hat{d}(t_{0}+\Delta t)$ and $\hat{d}(t_{0}+2\Delta t)$ on the prediction of the next $\hat{d}(t_{0}+(i+1)\Delta t)$, the role of $U_{1}$ on reducing the future lambdas in the prediction horizon is vivid. For, $N_{u}=1$, (42) decreases to

$$
U_{1}(k+1) = U_{1}(k) - \eta \frac{\partial (k)}{\partial U_{1}(k)} = U_{1}(k) - \eta \hat{d}(1) \frac{\partial \hat{d}(1)}{\partial U_{1}(k)}
$$

in which,

$$
\frac{\partial \hat{d}(1)}{\partial U_{1}(k)} = \frac{\partial \hat{d}(1)}{\partial P_{f}(2, 1)}
$$

6. Sliding mode controller

In this section, a SMC system is designed to compare with the newly proposed MPC. The sliding surface of the SMC system is defined as follows.

$$
S(t) = \lambda_{\text{sensor}} - \lambda_{d}
$$

In (46), $S$ is the sliding surface and $\lambda_{\text{sensor}}$ is the measured lambda by the sensor. The time derivative of (46) yields

$$
\dot{S}(t) = \dot{\lambda}_{\text{sensor}}
$$

The SMC system determines the control input (46) and (47) satisfying the following reaching condition:

$$
S(t) = 0 \Rightarrow \dot{\lambda}_{\text{sensor}} = \dot{\lambda}_{d}
$$

$$
\dot{S}(t) = 0 \Rightarrow \dot{\lambda}_{\text{sensor}} = 0
$$

Using (48) and (49) in (15) gives

$$
\dot{\lambda}_{\text{sensor}} = \frac{1}{\tau_{e}} (\dot{\lambda}_{\text{sensor}} + \lambda_{m}(1-\delta)) \bigg|_{\lambda_{\text{sensor}}=0} \dot{\lambda}_{d} = \frac{m_{a}}{\lambda_{d} \lambda_{AFR_{st}}}
$$

Substituting (50) in (16) yields

$$
u_{eq} = m_{f} = \frac{1}{1-X_{f}} \left( \frac{m_{a}}{\lambda_{d} \lambda_{AFR_{st}}} - m_{f} \right)
$$

The equivalent term of the injected fuel mass flow, $\nu_{eq}$, is obtained considering no uncertainty in the system. The robust SMC system to probable uncertainties is completed as $u(t) = \nu_{eq}(t) - K \text{sgn}(S(t))$

where $K$ is a positive real number and sgn stands for the sign function. The robust performance and the exponential stability of the SMC system (52) have been evaluated through Lyapunov’s direct method [14].

7. Simulation results of the MPC and the SMC systems

As shown in Sections 3.3, the identification error of the MLP is very small compared with that of the RBFN model. However, this error may accumulate in an enlarged prediction horizon, $N_{u}$. Since the variations of modified desired lambda are smooth and small, a short prediction horizon as $N_{u}=1$ is used to decrease the computational burden which is important for real-time applications. The iterations of the gradient descent algorithm in (42) is continued while the pre-defined threshold value, 0.01, for absolute tracking error is fulfilled. In addition, $U_{1}$ in (42) should satisfy its upper and lower bounds which are considered 1 and 0, respectively. The learning rate of the gradient descent optimization is 0.001 in (42) and the modified desired lambda is obtained as follows.

$$
\lambda_{d} = 1 - \exp(-Ct)
$$

In this paper, $C=5$. For a better comparison of results with that of the recent applications, the same profile of throttle plate angle in [2] subjected to 10% uncertainty is used in the simulations. Figs. 6 and 7 show the profile of the throttle plate angle with 10% uncertainty and the simulation results of the newly proposed MPC within the lambda window, respectively. The mean absolute tracking error here is 0.0081 which is significantly better than 0.4464 which has been recorded in [2]. The MAE between the MLP and MVEM response is 0.0057. The mean time cost for calculation in each sample time
Fig. 7 – Simulation results of the MPC with 10% input uncertainty.

Fig. 8 – Throttle plate angle profile with 25% uncertainty.

Fig. 9 – The simulation results of the MPC with 25% input uncertainty.
$\Delta t = 0.01 \, [s]$ is 0.0062 \, [s]. The short mean time cost for calculations in each sample time assures the possibility of real time implementation of the new MPC system.

To evaluate the robustness of the MPC system to larger uncertainties, the uncertainty on the throttle plate angle is increased to 25% as shown in Fig. 8.

In spite of the large uncertainty imposed to the proposed MPC system, the robust performance of the control system results in a small 0.0085 mean absolute tracking error as shown in Fig. 9. The MAE between the MLP and MVEM response is 0.0069.

The tracking performances of the SMC and the MPC systems are compared in Fig. 10. The Mean Absolute Error of the tracked lambda by the sliding mode controller is 0.0017 which is considerably smaller than 0.0081 of the MPC system. The trajectory of the injected fuel mass flow in both the MPC and SMC systems are shown in Fig. 11. The oscillations of injected fuel in the SMC system are considerably larger than those of the MPC. The harsh signal of injected fuel by the SMC system may be difficult to generate by real actuators. Besides, for implementation of the proposed SMC system the value of fuel film mass flow should be known which requires designing an observer or using extra sensor. It is also worthy of attention that the proposed MPC system based on MLP neural network is a model-free system and can be applied on wide range of dynamical systems without imposing extra cost for sensors.

8. Conclusion

As an alternative to classic PI-controllers, a new nonlinear MPC system based on MLP neural network identification has been proposed for stoichiometric air-to-fuel ratio control of SI engines. Considering modeling uncertainties and time varying dynamics of SI engines, the new MPC system has been
developed for robust control of lambda. As an accurate and control-oriented model of SI engines, the MVEM has been simulated to generate the required data for identification of two different MLP and RBF neural network systems. According to the validated results, the reduction of data collection sample time in training process would significantly increase the identification accuracy. Besides, a better control performance in the sense of MAE of lambda tracking is obtained through smaller sample times compared with what has been recorded in this research field. Regarding real-time implementations, the prediction horizon of optimization cost function has been shortened to 1 and the gradient descent algorithm as a simple and fast optimization method has been used. The performance of the MPC system has been evaluated compared with a first-order SMC system. Compared to the SMC system, the proposed MPC system results in smoother variations of the injected fuel.

REFERENCES