UNIVERSITY OF CALGARY

Computer Model for Emission Estimation and Control of Natural Gas, Spark Ignited, Stationary Internal Combustion Engines

by

Gabriel Alberto Cuenca Suarez

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA

JUNE, 2001

ABSTRACT

The objectives of this research were (1) to investigate suitable techniques for modeling and controlling gaseous emissions emanating from stationary internal combustion engines; (2) to apply these techniques in the design of a parametric emission monitoring and control system; (3) to test these techniques using a computer model; and (4) to show the feasibility of integrating them with an existing engine management system, REMVue, provided by REM Technology Inc. These objectives were accomplished in 5 stages: (1) development of a parametric emission modeling system using multidimensional arrays; (2) development of a parametric emission modeling system using artificial neural networks; (3) development of an emission control modeling system using artificial neural networks; (4) quantification of performance and comparison with traditional emission monitoring systems; (5) evaluation of the existing engine management system for integration with the neural network parametric emission monitoring and control system. Parametric emission modeling using both techniques was feasible. The neural network approach proved to be the most suitable technique for future real-time operation within the existing engine management system.

ACKNOWLEDGEMENTS

I wish to express my appreciation to my supervisor Dr. Martin P. Mintchev for his guidance, support and valuable suggestions throughout the course of this research.

I am specially obligated to Dr. Howard Malm for his excellent technical support and for his diligence in scheduling the different visits to the North Caroline Gas Plant.

I would like to thank Mr. Les Keast and Mr. Daniel Chow for their support during the engine data collection process. Without their support the project would not have been successfully completed.

The work on this thesis was assigned and supported by REM Technology Inc., a company owned by Spartan Controls Ltd.

The author of this thesis was supported in part by full-time educational loan granted by "Fundacion Gran Mariscal de Ayacucho", Venezuela.

The research presented in this thesis was supported in part by the University of Calgary, and the Alberta Software Engineering Research Consortium.

A mifamilia

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER I

CHAPTER III

CHAPTER IV

LIST OF FIGURES

CHAPTER I

CHAPTER II

CHAPTER III

CHAPTER IV

CHAPTER V

LIST OF SYMBOLS, NOMENCLATURE AND ABBREVIATIONS

<h	Fuel/air equivalence ratio
X	Lambda factor
	Actual Lambda factor
Xe	Estimated Lambda factor
r|	Rate of learning
y	Performance ratio
£	Symbol of summation
m	Air-mass flow-rate
m	Fuel-mass flow rate
f	Transfer function
W	Weight matrix
E(n)	Cost function
$e_{k}(_{n})$	Error signal
ykOO	Neural network output signal
dk(n)	Desired output signal
	Three dimensions
JSJQ	Analog-Digital
ACEMS	Alternative CEMS
AEmi	Actual Emissions
AMP	Air Manifold Pressure
AMT	Air Manifold Temperature
BACT	Best Available Emission Control Technology
BC	Before Top Center
BFGS	Broyden, Fletcher, Goldfarb, and Shanno
BHP	Brake Horse Power
BIOS	Basic Input Output System
BMEP	Brake Mean Effective Pressure
BP	Brake Power or Backpropagation
BSFC	Brake Specific Fuel Consumption
bTC	Before Top Center
CAAA	Clear Air Act Amendments
CAD	Computer Aid Design

CAM	Compliance Assurance Monitoring
CEMS	Continuous Emission Monitoring System
CEPA	Canadian Environmental Protection Agency
CFR	Code of Federal Regulations
CH_4	Methane
CO	Carbon Monoxide
$C0_2$	Carbon Dioxide
D/A	Digital-Analog
DC	Direct Current
DCPL	Demonstrated Compliance Parameter Limit
DCS	Distributed Control System
Dp	Discharge pressure (Compressor)
EEmi	Estimated Emission levels
EPA	Environmetal Protection Agency (United States)
ExhT	Exhaust Temperatures
GUI	Graphic User Intergace
HC	Hydrocarbons
HEGO	Heated Exhaust Gas Oxygen
HP	Horse power
I/O	Input-Output
LExht	Left Exhaust Temperature
LM	Levenberg Maquardt
Loada	Actual Load
Loade	Estimated Load
MEX	MATLAB binary files
MRE	Maximal Relative Error
MSE	Minimal Square Error
NMHC	Non-Methane hydrocarbons
NN	Neural Network
NO	Nitric oxide
$N0_2$	Nitrogen dioxide
NO_x	Oxides of nitrogen
NVM	Non-Volatile memory
O3	Ozone
PCx	External personal computer
PEMS	Parametric Emission Monitoring System
PID	Proportional-Integral-Derivative

PLC	Programmable Logic Controller
ppm	Parts per million
PPMV	Parts Per Million based on Volume
PRET	Periodic Routine Emission Testing
RACT	Reasonable Available Control Technology
RAM	Random Access Memory
RExhT	Right Exhaust Temperature
ROM	Read Only Memory
RPM	Revolutions Per Minute
SI	Spark Ignition
SISIC	Spark Ignition Stationary Internal Combustion
SO2	Sulfur dioxide
SO_x	Oxides of Sulfur
Sp	Suction pressure (Compressor)
SparkTa	Actual Spark Timing
SparkTe	Estimated Spark Timing
Speeda	Actual Speed
Speede	Estimated Speed
St	Spark timing
TC	Top Center
THC	Total Hydrocarbons
TOC	Total Organic Compounds
VEMS	Virtual Emission Monitoring System
VOC	Volatile Organic Compounds

INTRODUCTION TO THE PROBLEM OF GAS EMISSION MONITORING AND CONTROL OF SPARK IGNITION STATIONARY INTERNAL COMBUSTION ENGINES

Deterioration of the atmosphere from gaseous pollutants is an important environmental issue. Local, state and national governments have enacted stricter exhaust emission legislation to reduce and possibly reverse atmospheric deterioration [1].

One particular area that has received a great deal of attention in the last few years is utilizing gaseous fuels to power spark-ignition, stationary internal combustion (SISIC) engines for driving compressors used in the oil and gas industry [2]. Environmental legislation has affected these installations by limiting the horsepower allowed, or by requiring very low emission levels out of these engine-compressor units. Because of this situation, natural gas internal combustion engine manufacturers continue to develop products which help to meet these environmental requirements. In addition, exhaust treatment companies have developed processes which reduce pollutants by converting them into safe, naturally occurring compounds that are not damaging to the atmosphere [3].

Environmental regulations, specifically those enacted by the Environmental Protection Agency and the Canadian Environmental Protection Agency [1], have stimulated an increased interest in monitoring and control of actual exhaust emission output from SISIC engine units. Access to actual levels of exhaust emissions assists in verifying operating permit compliance, facilitates computation of emission fees or trading credits, and enables a more comprehensive assessment of national emission inventory levels.

Several methods for controlling the performance of spark ignition internal combustion (SISIC) engines have been suggested [2,4-6]. Few of them have attempted to model, monitor, and control toxic emissions such as greenhouse gases (CO, CO2 and CH_4), nitrogen oxides (NO_x), unburned hydrocarbons (HC), and volatile organic compounds generated during the combustion process [2, 7-10].

Presently, there are several different techniques for engine exhaust gas quantification: (1) using a collection of expensive equipment (gas analyzers and toxic gas sensors) to sample, analyze and provide a continuous record of emission rates with the possibility for integration and forming the so-called Continuous Emission Monitoring Systems (CEMS) [11]; (2) empirically calculating these emissions using very complex chemical and thermodynamic equations [12]; (3) using Demonstrated Compliance Parameter Limit (DCPL) systems, which give no indication what the actual emissions are, but rather deliver an assurance that emissions are below a target range providing the engine is operated within a defined operational envelope [11]; (4) using empirical model-based Parametric Emission Monitoring Systems (PEMS), which are typically multidimensional arrays or look-up tables containing data for several operational parameters that describe engine performance [8,11]; and (5) designing and implementing Virtual Emission Monitoring Systems (VEMS) to model the engine performance and to estimate the emission rates based on different engine operational conditions [9].

Emission control has been a difficult task to achieve due to a lack of expertise. Few experts have studied and published optimal engine conditions that might be used to tune the engine performance for compliance with the legally required emission levels [2,9]. Moreover, there are no available tools on the market that can provide an engine-compressor unit operator with the required information to manipulate the engine operation and control the emission levels as desired.

SCOPE OF THE THESIS

As public awareness of environmental issues increases, a significant amount of research is being done to find practical solutions for continuously monitoring and controlling the different toxic emissions produced by oil and gas facilities.

The aim of this thesis is to suggest a method for estimating selected gas emissions (NO_x, HC, CO and CO2) in oil and gas facilities that are generated as products of combustion. Design and implementation of two computer-based models for emission estimation are presented. The models are capable of estimating different gas emissions produced by natural gas, stationary internal combustion engines without the use of expensive gas analyzers or toxic gas sensors. Furthermore, an open-loop control of gas emissions is presented and is embedded into the second model, so that a reliable tool for the estimation and control of these emissions could be achieved. Finally, integration of the developed system with the existing engine management system *REMVue,* developed and provided by REM Technology Inc., is described.

The thesis is structured into five chapters.

Chapter I introduces the fundamentals of internal combustion engines and different types of emission monitoring systems. Their advantages and disadvantages for solving the present problem are outlined.

* REMVue is the given name to the engine management system provided by REM Technology Inc.

Chapter II describes the development of a parametric computer model for the estimation of exhaust emissions from spark-ignition, natural gas, stationary, internal combustion (SISIC) engines, based on multidimensional arrays. The implementation and results are discussed. This first approach was published in the Proceedings of the 5th Biennial Conference on Engineering Systems Design and Analysis, Montreux, Switzerland, July 10-13, 2000, and was presented at the same conference by the author.

Chapter III presents a second approach for emission estimation, based on artificial neural networks. An overview of artificial neural networks and their applicability to the present problem is presented. A neural network-based design modeling for estimation of gas emissions for SISIC engines used in the oilfield is outlined. The design, implementation and results are discussed in details. This second approach was published in the Proceedings of the 6th International Conference on Control, Automation, Robotics and Vision (ICARCV2000). Singapore, December 5-8, 2000, and was presented by the author.

Chapter IV describes an open-loop control system for the emission control using artificial neural networks. The overall experimental work for testing the two approaches for emission estimation and the open-loop emission control system is presented, including a detailed description of the data collection process.

Chapter V encompasses the integration of the developed estimation and control system with an existing engine management system. The features of the REMVue engine management system are provided, and a protocol to build a stand-alone application to be embedded into the present engine system is suggested.

Finally, a conclusion chapter discusses the limitations of the neural network-based emission monitoring and control system, and some operational details that the users must be aware of for further implementation of the proposed approach.

CHAPTER I: INTERNAL COMBUSTION ENGINES AND EMISSION TECHNOLOGIES

Some essential features of the combustion process of internal combustion engines must be briefly outlined to provide the reader with a background on the generation of exhaust pollutants, and on the factors which affect this generation. Furthermore, gas emission technologies are introduced to clearly understand the scope of the present work.

1.1. Engineering Fundamentals of Spark Ignition Internal Combustion Engines and their Emission Pollutants.

The purpose of internal combustion engines is the production of mechanical power from the transformation of chemical energy contained in the fuel used [13]. In a conventional spark-ignition (SI) engine the fuel and air mix together in the intake system. Then, they are inducted through an intake valve into the cylinders, where they get compressed. Under normal operating conditions, combustion is initiated towards the end of the compression stroke by an electric discharge at the spark plug. A turbulent flame develops, propagates through this pre-mixed fuel-air mixture until it reaches the combustion chamber walls, and then extinguishes. Figure 1.1 illustrates the combustion process in a four-stroke SISIC engine.

Basically, each cylinder requires four strokes of its piston, i.e., two revolutions of the crankshaft to complete the sequence of events, which produce one power stroke. This cycle is known as the Otto cycle [13].

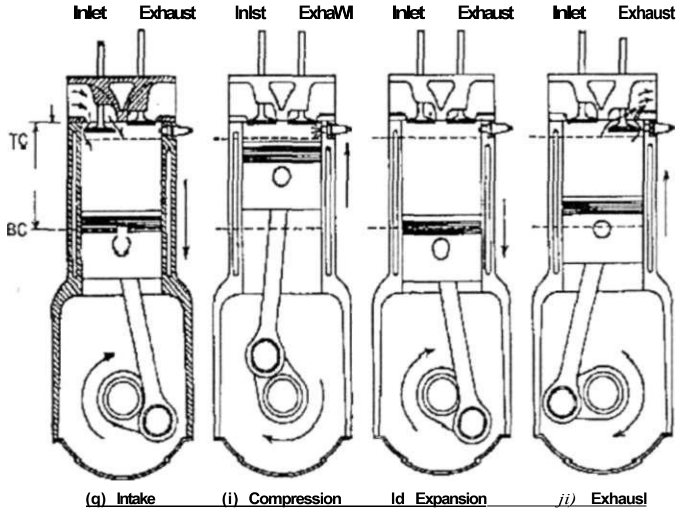


Figure 1.1. The four-stroke operating cycle (Otto cycle)

(1) The cycle commences with an intake stroke, which starts with the piston at top center (TC) and ends with the piston at bottom center (BC). During this first stroke the inlet valve opens to admit fresh air-fuel mixture into the cylinder.

(2) Then a compression stroke takes place. The inlet and exhaust valves are closed, and the mixture inside the cylinder is compressed toward the end of the cylinder. Shortly before TC the spark starts the combustion. The pressure of the gases resulting from the combustion process increases and reaches a maximum just after TC.

(3) A power stroke, or expansion stroke takes place, starting with the piston at TC and ending at BC as the high-pressure of the gases push the piston down and force the crank to rotate. As the piston approaches BC, the exhaust valve opens to initiate the exhaust process, and the cylinder pressure drops to the exhaust manifold pressure.

(4) Finally, an exhaust stroke occurs, when the remaining burned gases exit the cylinder through the exhaust valve, and then are released to the environment via the exhaust pipe.

Spark-ignition (SI) engines are a major source of air pollution. The SI engine exhaust gases contain oxides of nitrogen (ie., NO_x, including nitric oxide NO, and small amounts of nitrogen dioxide NO_2), carbon monoxide (CO), organic compounds which are unburned hydrocarbons (HC), carbon dioxide (CO_2) and oxides of sulfur (SO_x) [13,14].

Formation of oxides of nitrogen takes place during the high combustion temperatures in the combustion chamber containing atmospheric nitrogen, oxygen, and fuel. These gases harm humans and animals by reducing breathing capacity and limiting the blood's ability to carry oxygen [1,18]. In the lower atmosphere, when exposed to sunlight, they act as precursors in the formation of ozone (O_3) [1].

Carbon monoxide is formed by incomplete combustion. This occurs when there is insufficient oxygen near the fuel (hydrocarbon) for complete combustion or when combustion is quenched near a cold surface in the cylinder. CO is a poisonous gas, which causes nausea, headache and fatigue, and in heavy concentrations can cause even death. In addition, it reacts with O_3 in the upper atmosphere, producing carbon dioxide (CO_2), which depletes the ozone layer [1].

Natural gas is a fuel made up of several hydrocarbon gases including methane, ethane, propane, butane, and other heavier compounds. A small fraction of these hydrocarbons can pass through the combustion chamber or cylinder without reacting, therefore, they retain their form in the exhaust. These hydrocarbon emissions are commonly broken down into two categories: (1) Total hydrocarbons (THC) or Total Organic Compounds (TOC) which include all of the hydrocarbon gases found in the exhaust stream; and (2) Non-Methane Hydrocarbons (NMHC) or Volatile Organic compounds (VOC) containing the portion of the THC except methane. Methane (CH_4) is a greenhouse gas with a very high global warming potential. In addition, non-methane hydrocarbons can react with NO_x in the lower atmosphere, acting as the precursors of photochemical smog [1].

Oxides of sulfur are formed when sulfur compounds in the fuel are oxidized in the combustion chamber. Oxides of sulfur enter the atmosphere and combine with water in the air forming sulfurous acid and sulfuric acid. These acids return to Earth as acid rain [!]•

1.1.1. Emission regulations and systems

Regulations, governing the quantity of emission levels which a gas engine can discharge, vary between different regions (countries, cities and towns) due to the air quality in these regions, and political factors [1,18]. Usually, regions with poor air quality have much tighter restrictions on exhaust emissions than areas where air quality is relatively good. For this reason, the local air quality board must be contacted to determine if a project is in compliance when engines are considered for new projects or re-powers [!]•

In North America, there are two major organizations that enforce emission monitoring and control from various sources. The Environmental Protection Agency (EPA) in the United States, and the Canadian Environmental Protection Agency (CEPA) are responsible for regulating gaseous emissions from stationary and non-stationary sources.

In Canada, a new Canadian Environmental Protection Act was enacted in 1999. It appears to be the most advanced environmental law of its kind in the developed world and will be the societal most important tool in preventing the release of toxic substances not only into our air but also into the water. These new and existing laws promise to guarantee Canadians the highest standards of environmental and health protection, since they emphasize the importance of preventing pollution rather than cleaning it up after the fact.

On the other hand, in the United States, the EPA is reviewing the standards of performance for new stationary emission sources in order to prevent, and revert the effects that combustion related units have had during the past ten years [15]. The Clean Air Act Amendments of 1990 [16] called for enhanced monitoring of combustion emissions. Later, the monitoring portion of the amendment was withdrawn and replaced by the Compliance Assurance Monitoring (CAM) rule, which offered flexibility for monitoring programs that can comply with continuous compliance data requirements [1,15]. While the specific requirements for monitoring under CAM are less specific, the need to have good combustion practices, and to be emission compliant at all times has not been lessened. Additionally, some oil and gas sites are currently required to monitor emissions to show continuous compliance due to local air quality standards.

Best Available Emission Control Technology (BACT) is defined by the United States EPA and most of the other regulatory agencies as the technology capable of achieving the maximum degree of reduction of a given pollutant from an emitting facility on a case-by- case basis [1,15]. However, other factors such as energy, environmental and economic impact must be taken into account when evaluating the emission control technology used, thus limiting this concept. Therefore, the recently introduced Reasonable Available Control Technology (RACT, [1]) seems to be preferable by most of the emission technology manufacturers. This type of technology can achieve a fairly low emission limit applicable to a specific source using reasonably available and economically feasible control equipment [1].

Typically, some regions limit production from a natural gas source based on the amount of useful energy it is producing. For natural gas engines, the useful energy is given in Kilowatt-hours (kW-hr) or Brake horsepower-hours (BHP-hr) of mechanical energy. Therefore the emission rates are regulated in grams/Kilowatt-hours (or grams/BHP-hr) per engine. This method is called "Pollutant Per Energy Unit Generated". Alternately, exhaust emissions are regulated based on the volume of exhaust produced. The common unit is Parts Per Million based on Volume (PPMV), or parts-per-million (ppm). This method is known as " Pollutant Per Unit Volume of Exhaust". Both methods are used within the present work.

A. **Emission inventories**

NO_x, greenhouse gases (CO2, CH4), CO and VOCs are the main pollutants addressed in this subsection. NO_x and SO2 emissions are the dominant precursors of acidic deposition; NO_x and VOCs are primarily contributors to the formation of ground-level ozone; NO_x, SO2 and VOCs, together, contribute to particulate matter (PM) formation. Atmospheric methane (CH_4) is an integral component of the greenhouse effect, second only to CO2 as a contributor to anthropogenic greenhouse gas emissions. The overall contribution of methane to global warming is significant because it is estimated to be 21 times more effective at trapping heat in the atmosphere than CO2 [18].

Carbon monoxide is a byproduct of highway vehicle exhaust, which contributes about 60 percent of all CO emissions nationwide [17]. In cities, automobile exhaust can cause as much as 95 percent of all CO emissions. These emissions can result in high concentrations of CO, particularly in local areas with heavy traffic congestion. Other sources of CO emissions include industrial processes and fuel combustion in internal combustion engines, boilers and incinerators. The overall CO concentrations in North America is decreasing; however, some metropolitan and rural areas are still experiencing high levels of CO [17].

A.1. Nitrogen oxides (NO_x)

The principal anthropogenic source of NO_x emissions is the combustion of fuels in stationary and mobile sources. This occurs in motor vehicles, residential and commercial furnaces, industrial and electrical utility boilers and engines, and other equipment. Table 1.1.a presents the overall estimated trends of NO_x levels in Canada and the United States from 1980 to 2010. Table 1.1.b reflects the NO_x emissions by source category for 1995 [17].

NO_x Emissions	Canada (million tons)	U.S (million tons)
1980	5.2	22.1
1985	4.1	20.5
1990	4.3	20.8
1995	3.0	21.0
2000	3.1	19.0
2005*	3.0	18.5
2010*	2.8	18.2

Table J.La. Canada and U.S overall NO_x emissions for 1980-2010 (^Projected emissions)[17]

Source categories	**Canada (1995)**	**U.S (1995)**
Fuel Combustion (%)	23	19
Transportation (%)	59	50
Electric utilities (%)	11	27
Other industrial sources (%)	7	4
Canadian TOTAL (million tons)	3.0	21

Table 1. Lb. Canada and U.S. NO_x Emissions for 1995 [17]

The largest contributor of NO_x in Canada is the transportation sector, which accounts for about 60% of all emissions. Improvements are expected by 2010, with an anticipated decline in NO_x emissions of 10% from the 1990 levels. For stationary sources, Canada is on target to meet its commitment to reduce national stationary source NO_x emissions by the year 2000, with expected reductions well in excess of the required 100 kilo-tons (kt). Reductions are in place at major combustion sources, power plants, and metal smelting facilities. Stricter emission limits for reducing NO_x emissions from new power plants were established in 1995, and further tightening of the post 2000 emission limits is in progress. New guidelines have also been developed for reducing NO_x emissions from new gas-fired reciprocating compressor engines, gas turbines, modified commercial and industrial boilers, process heaters and cement kilns [17].

The United States continues to address NO_x emissions from stationary and mobile sources under the 1990 Clear Air Act Amendments (CAAA), which mandated a two million ton reduction in NO_x emissions for 2001.

A.2. Volatile Organic Compounds (VOC)

VOCs contribute to the formation of ground-level ozone. Anthropogenic emissions of VOCs come from wide variety of sources, such as mobile sources (transportation), stationary sources, and industrial sources (e.g., chemical manufacturing and production of petroleum products). VOC emissions in both countries are expected to decline in the year 2000, and remain stable until 2010. Overall estimated trends in anthropogenic VOC emissions from 1980 to 2010 for Canada and the United States are shown in Table 1.1.c [17].

VOCs Emissions	Canada (million tons)	U.S (million tons)
1980	2.0	24.0
1985	3.5	21.8
1990	3.0	18.5
1995	3.2	18.6
2000	3.1	14.5
2005*	3.0	13.8
2010*	3.1	14.0

Table I.I.e. Canada and U.S overall VOCs emissionsfor 1980-2010 (^Projected emissions) [17]

A.3. Greenhouse gases: Carbon Dioxide and Methane

The global carbon cycle is made up of large carbon flows and reservoirs. Hundreds of billions of tons of carbon in the form of CO_2 are absorbed by oceans and living biomass (sinks) and are emitted to the atmosphere annually through natural processes(sources).

Since the Industrial Revolution, the equilibrium of atmospheric carbon has been altered. According to emission inventories recorded by the EPA, atmospheric concentrations of CO_2 have risen about 28 percent [17] in the last 15 years, principally because of fossil fuel combustion, which accounted for almost 99 percent of the total U.S. and Canadian CO_2 emissions in 1997.

Over the last two centuries, methane concentration in the atmosphere has more than doubled [18]. Scientists [17,18] believe these atmospheric increases were due largely to increasing emissions from anthropogenic sources, such as landfills, natural gas and petroleum systems, agricultural activities, coal mining, fossil fuel combustion, wastewater treatment, and certain industrial processes [17,18].

Even though other pollutants such as NO_x, SO2 and VOCs have decreased dramatically over the last decade, a remarkable contrast is reflected on CO2 and CH4 targets. Table 1.1.d summarizes U.S. and Canadian CO2 and CH_4 levels for 1990-1995. Additionally, Table 1.1.e, shows CO2 levels generated from the energy and non-energy sector.

Emission	**1990**	**1991**	**1992**	**1993**	**1994**	**1995**
$C0_2$	463,200	453,700	467,300	469,000	482,400	499,600
CH_4	66,300	68,300	70,400	72,600	75,600	78,600

Table lid. Greenhouse Gases, 1990-1995 (Kilotons)[17]

Year	**Energy**	**Non-energy**	**Total**
1990	468	97.7	567
1995	517	102.3	619
2000	536.7	73.2	609.9
2005	557.5	79.0	636.5
2010	583.0	85.7	668.7

Table lie COj equivalent basis Megatons (Mt)[17]

Today, organizations such as Environment Canada and the CEPA are trying to enforce new targets for CO2 and CH4 emissions for the next ten years. However, it is not very clear how the Canadian government will deliver this commitment. Meanwhile, both

Canadian and U.S environmental organizations are promoting new technologies to quantify and control these critical pollutants. Despite the fact that the greenhouse gas levels have decreased at monitoring stations [17], concern for quantifying uncertain emission sources (e.g., new stationary combustion related units) is increasing.

A.4. Carbon Monoxide (CO)

Table 1.1.f summarizes U.S national CO emission levels for 1992-1998 [17]. Clearly, the overall CO levels decreased 10% since 1994. However, CO emissions generated from stationary combustion related units have increased since 1992 due to the increased natural gas production.

Source	1992	1993	1994	1995	1996	1997	1998
Fuel Combustion	6,155	5,587	5,519	5,934	6,148	5,423	5.374
(a) Electricity	47	51	55	58	54	56	57
(b) Oil & Gas	227	268	251	245	301	295	289
Industrial Processes	5,683	5,898	5,838	5,790	4,692	4,844	4,860
Transportation	78,858	79,593	81,629	74,331	73,494	71,980	70,300
Miscellaneous	6,934	7,082	9,657	7,298	11,144	12,164	8,920
TOTAL	97,630	98,160	102,643	93,353	95,479	94,410	89,454

Table 1.1.f. U.S National Carbon Monoxide Emissions Estimates, 1992-1998 (thousand of tons), (a) & (b) represent CO levels from internal combustion engines used in electric utilities and oil and gas industry [177.

A.5 Sulfur dioxide (SO2)

The principal anthropogenic sources of SO2 are coal and oil combustion, smelting and few industrial processes. $S0_2$ emissions are declining in Canada and the United

States. Overall trends in emissions levels from 1980 to 2010 are shown in Table 1.1.g [17].

SO_2 Emissions	Canada (million tons)	U.S (million tons)
1980	5.1	23.5
1985	4.2	20.8
1990	4.3	20.5
1995	3.2	17.5
2000	3.3	15.0
2005*	3.2	15.2
2010*	3.2	15.1

Table 1. l.g. Canada and U.S overall SO2 emissions for 1980-2010 (^Projected emissions) [17]

Canada has surpassed its international and domestic commitments to reduce emissions of SO_2. In 1997, national emissions were approximately 2.6 million tons, or 18% below the cap of 3.2 million tones. Furthermore, forecasts of emissions of SO_2 up to the year 2010 indicate that Canadian emissions will remain well below these caps.

Despite meeting and exceeding its commitments, Canada remains concerned about acid rain and depletion of the ozone layer. As a result, Environment Canada is forcing some provinces such as Alberta, and Ontario, to develop strategies that would lead to the establishment of new $S0_2$ targets and reduction schedules. Recent reports [17] concluded that emission reductions commitments of up to 75% beyond current commitments would be required in specific target regions of Canada (e.g., Alberta, Ontario) and the United States to prevent damage to sensitive ecosystems in Canada.

Natural gas, stationary internal combustion engines do not contribute to SO_2 generation due to the fact that the fuel (natural gas in the present case) is not expected to contain sulfur. However, as mentioned before, SO_2 contributes to other effects when reacting with pollutants such as NO_x and UHC (unburned hydrocarbons), which are obviously generated from these units.

B. Emission Monitoring Systems

Various systems can be selected to indicate emission compliance. Some of these compliance-monitoring technologies are still maturing, while some of them have evolved to a more useful state. At present, the four main options for emission monitoring are Continuous Emission Monitoring Systems (CEMS), Demonstrated Compliance Parameter Limit (DCPL) systems, Periodic Routine Emission Testing Systems (PRETS), and Parametric Emission Monitoring Systems (PEMS) [11].

Canada and the United States differ in the compliance monitoring systems that they use to measure gaseous emissions from utilities. Due to high installation and maintenance costs, plus relatively fair air quality, Continuous Emission Monitoring Systems (CEMS) are not yet fully utilized in Canada as a tool applicable to all major NOx. CO, HC and CO_2 sources. Instead, methods of comparable effectiveness to CEMS are utilized, even though research into more reliable and inexpensive methods has been progressing in the recent years. CEMS are currently used for NO_x, CO, HC and SO_2 emission measurements in units greater than 25MW, utility boilers, and cogeneration facilities, due to the fact that they are potential contributors of these pollutants, and that no effective method to estimate this emission is available on the market. Other sources such as small cogenerators and pipeline compressors still depend on alternative methods (e.g., annual sampling, parametric performance analysis).

B.1. Continuous Emission Monitoring Systems

Continuous Emission Monitoring Systems incorporate advanced sensor technology to detect and quantify emission concentrations [11,18]. CEMS manufacturers continue working this technology while improving the economics and maintainability.

Presently, the NO_x and CO CEMS are among the most relevant emission monitoring systems due to their potential to deteriorate the atmosphere [18]. Figure 1.2 shows a diagram of a particular continuous emission monitoring system using two NO_x sensors, and an oxygen sensor. Additionally, a three-way catalytic converter, and a NO_x catalyst are utilized to enhance NO_x reduction [2].

The CEM system measures the following variables: (1) Supplied air by using a mass-air flow sensor; (2) Supplied fuel, measured by a fuel flow sensor; (3) Oxygen concentration in the exhaust system using a Heated Exhaust Gas Oxygen (HEGO) sensor; (4) Speed or RPMs; and (5) NO_x concentrations in the exhaust system, before and after the NO_x catalyst.

As described in section 1.1, air is supplied and controlled by three different valves not shown in Figure 1.2. Air/fuel mixture pressure is controlled by a throttle valve, which has an attached throttle position sensor for proper control. Thus, the air/fuel mixture enters the intake manifold of the engine. Spark timing is controlled by a spark ignition system, and an air assist system (turbocharger) is present to boost the air pressure as desired. Two NO_x sensors are used simply to determine whether the three-way catalytic converter or the NO_x catalyst requires maintenance.

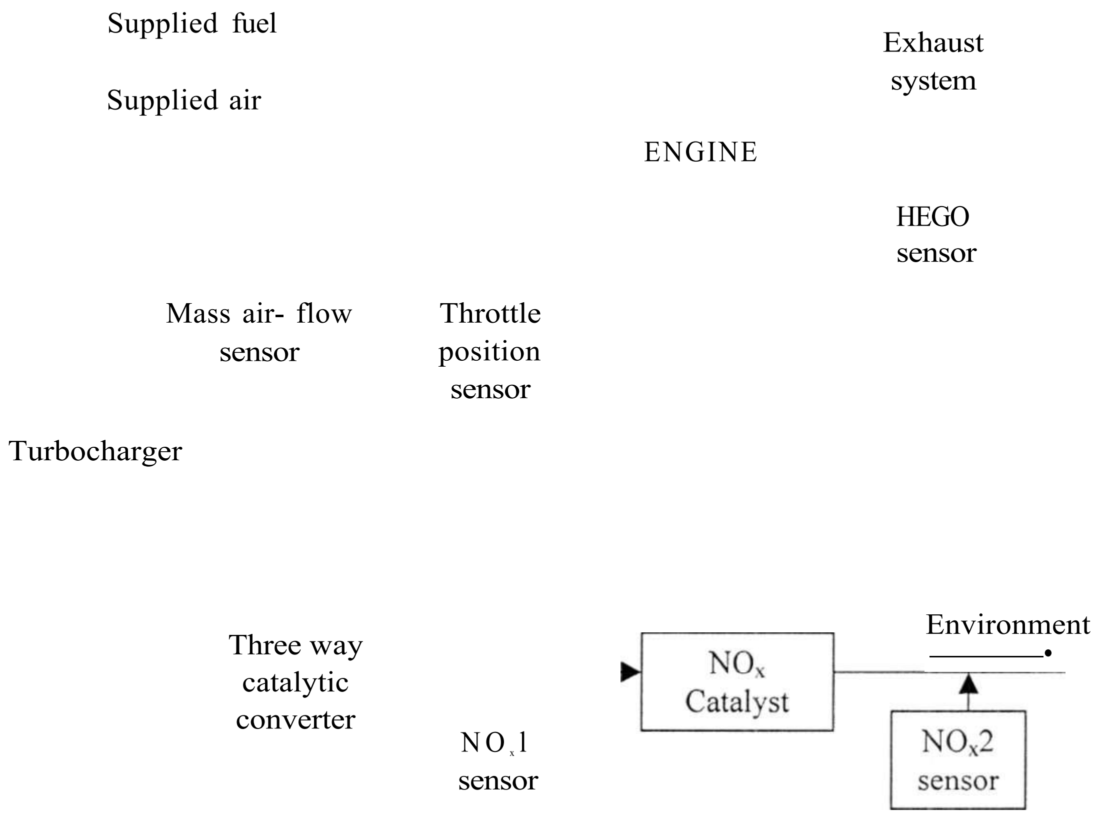


Figure 1.2. CEM system using O2 and NO_x sensors for automotive exhaust emission systems [2J.

A closed loop system to control the air/fuel ratio requires quantification of the oxygen concentration in the exhaust. Therefore, an HEGO sensor is placed in the exhaust pipe of this particular system.

In some CEMS, gas sensors are replaced by a gas analyzer which is directly connected to the exhaust pipe of the engine. The analyzer is interconnected to a personal computer (PC) where emission levels are recorded and stored in a database for later report generation. The use of gas analyzer involves high maintenance costs due to frequent calibration, and short life span of its gas cells (1-2 years).

Presently, the automotive industry is leading the edge of gas sensor technology [19]. It continues using and improving NO_x and O2 sensors. Considerable work is also being done in infrared (IR) technology to detect all chemical components in the automotive exhaust emission systems. However, in the majority of oil and gas facilities, users and experts view the purchase, operation, and maintenance costs of CEMS as prohibitive considering the life span of these systems (1-2 years).

B.2. Demonstrated Compliance Parameter Limit (DCPL) systems

DCPL systems give no indication of what the actual emissions are, but rather give an assurance that emissions are below certain target when the engine is operated within a define operational window [11]. DCPL systems can be applied to situations where unit operation is stable, meaning that the engine is base-loaded, and controlled parameters such as air manifold pressure (AMP), air manifold temperature (AMT), oil and coolant temperatures, etc., are invariant.

At first glance, the simplicity of DCPL systems looks attractive. The downsides of this simple solution are (1) that the engine typically is constrained to a narrow operational range, which can restrict its usefulness, and (2) since the actual emission levels are not known, they can not be inventoried for fee assessment.

B.3. Periodic Routine Emission Testing (PRET)

Scheduled portable or reference lab emissions testing can also serve to fulfill certain emission monitoring requirements. PRET systems refer to emission analyzers that are relatively affordable [11,20]. To collect emission data accurately, the analyzer needs a well-designed sample conditioning system that disallows water from condensing in the sample line and has a low volume to detect transient emission variations. In order to utilize the analyzer it must have a battery of test gases available for the entire range of engines to be tested and calibrated. In other words, before using the analyzer, it must be calibrated at different gas concentrations provided that the battery of gases is utilized as a certified reference.

Scheduled routine emission testing apparently seems to be very appealing option for emission monitoring, particularly when the testing can be done with low cost portable units. However, the portable analyzer must have a battery of test gases for the entire range of engines to be tested, typically with at least five gases. This battery of tested gases ages with time, with a typical lifetime between two or three years. Therefore, for an analyzer measuring NO, NO_2 and CO on a range of engines, a minimum of fifteen different calibration gases are required. It is worth noting that even though the emission analyzer might be very portable, the calibration gases are not.

Another disadvantage of scheduled routine emission testing is that if an engine is found out of compliance, fines can be levied back to the last defensible in-compliance condition, which corresponds to the last emission test. As a result, if the last emission test were six months earlier, then the fines could accrue back to that time and be significant.

1.1.2. Engine parameters related to pollutant generation

Variation of the emission levels from SI engines depends on the engine operating parameters. Basically, there are four major operating variables that affect not only spark-ignition engine emissions but also the performance and the efficiency of the unit. They are the (1) Air/fuel ratio, (2) Speed, (3) Load and (4) Spark timing [12-14]. These parameters can be defined as engine control parameters due to the fact that they can be manipulated to achieve a specific engine performance.

The following is a discussion of how variations of these engine operational variables influence gas emission levels generated by the SI engines.

A. Air/fuel ratio

In engine operations, both the air mass flow-rate *(m_a),* and the fuel mass flow-rate *(rrif)* are routinely measured. The ratio of these flow rates defines the so-called air/fuel ratio [12]:

$$Air\ I fuel\_ratio = \frac{m}{m_f} \tag{1.1}$$

Another way in to represent the air/fuel ratio is with the excess air ratio referred to as the **Lambda* factor *(k).* Excess air ratio is defined as the ratio between the operating air/fuel ratio and the stoichiometric air/fuel ratio. A stoichiometric air/fuel ratio or mixture indicates a chemically correct mixture of air and fuel which makes possible all molecules of each fuel component to be completely burned during the combustion process [14].

$$X = \frac{\mathrm{Air\,/\,fuel\_\ ratio(operating)}}{\mathrm{Air\,/\,fuel\_ratio(stoichiometric)}} \tag{1.2}$$

where $X = 1$ at the stoichiometric air/fuel ratio.

The fuel/air equivalence ratio (O) is inversely proportional to the Lambda factor *X*. ***A** rich* air-fuel mixture (O >1, *X* <1) characterizes a mixture with absence of oxygen in the exhaust system after combustion, while a *lean* air-fuel mixture (O < 1, *X* > 1) means that there is an amount of oxygen present in the exhaust stream after combustion [14]. Emission levels from SI engines are extremely sensitive to the variation of O. Figure 1.3 illustrates NO_x, CO and HC exhaust gases compared to O.

The fuel/air equivalence ratio is an important parameter controlling SI engine emissions. The critical factors affecting emissions that are governed by the fuel/air equivalence ratio are the oxygen concentration and the temperature of the burned gases [2,13,14]. The maximum exhaust gas temperatures occur when the engine operates slightly rich at O =1.1. As the mixture becomes lean, the increasing oxygen concentration initially offsets the falling gas temperatures. However, NO_x peaks abruptly at O =0.95. After this point, with the decrement of the cylinder temperature, NO_x emissions decrease to low levels.

Figure 1.3 also shows the effect of variations in O on the HC emissions. For rich mixtures, HC are very high due to the lack of oxygen for burning the residual unburned hydrocarbons that escape the primary combustion process within the cylinder and the exhaust system [14].

* Lambda factor *(X)* is the actual parameter controlled by the exisitng engine management system

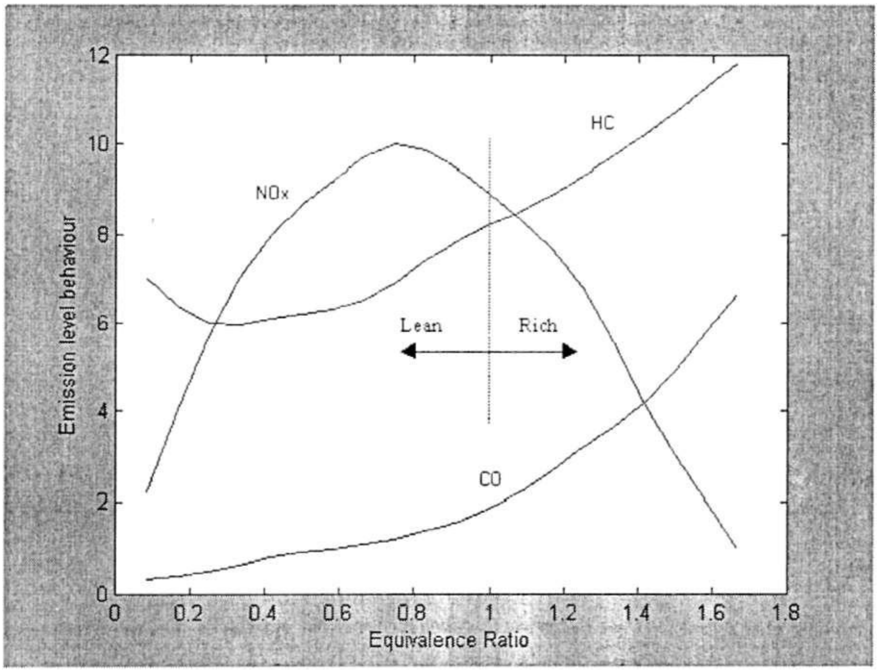


Figure 1.3. Effect ofthe variation on the equivalence ratio (0) on the engine exhausts gases (NOx, HCand CO) [14].

For moderately lean mixtures (O < 1). HC emission levels decrease until a certain limit (Φ =0.8). When Φ is approaching the lean limit (Φ < 0.8). combustion quality deteriorates, and HC emissions start to rise again due to the occurrence of misfiring cycles (incomplete combustion).

Considering the CO emissions, it is obvious (see Figure 1**.3)** that for rich mixtures CO levels are high because complete oxidation of the fuel carbon to $CO2$ is not possible due to insufficient oxygen. For lean mixtures, CO levels are approximately constant at very low levels.

(REMVue) used in the present work.

B. Speed and Load

These two engine parameters are always used to analyze the engine performance by determining the highest power an engine is able to develop at a certain speed [13].

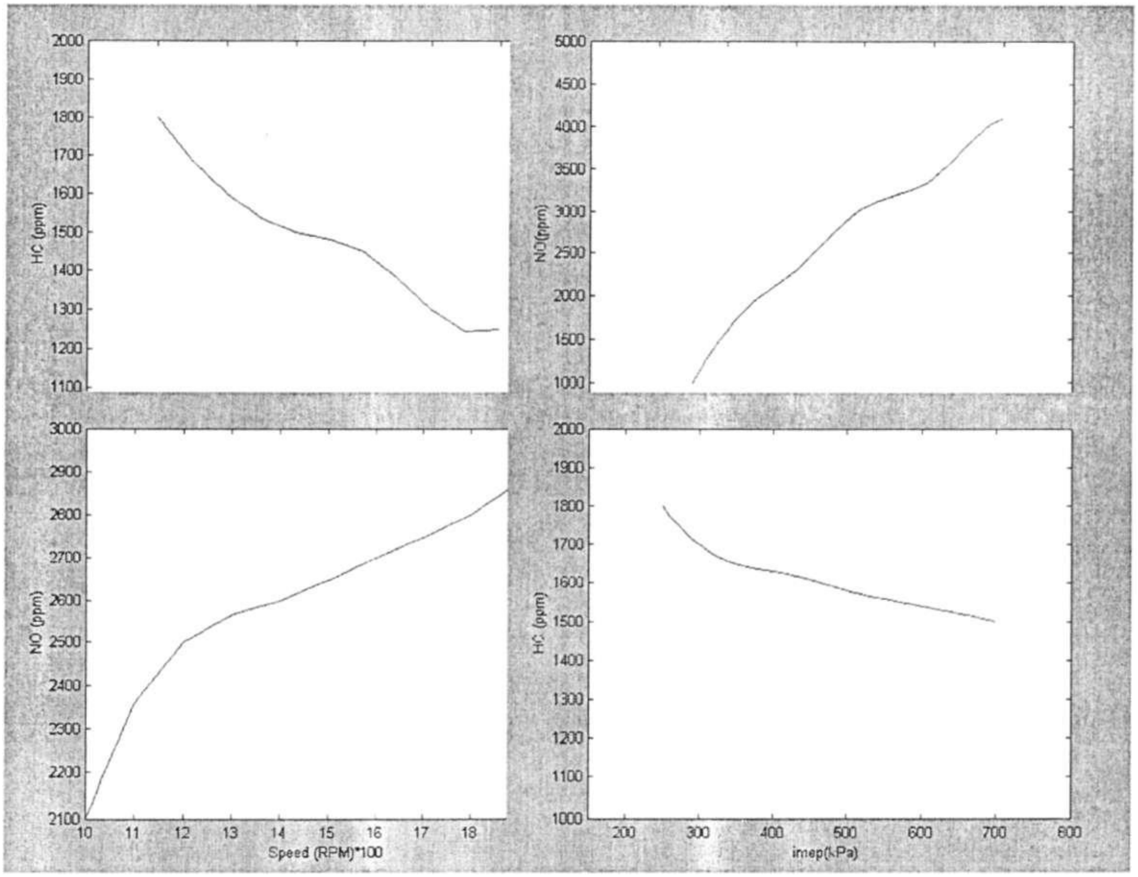
Rated speed is a performance definition commonly used when referring to the crankshaft rotational speed at which a rated power is developed. It is given in revolutions per minute (RPM), and for SISIC engines, it varies from 250 to 1800 RPM. In addition, engine speed determines the amount of time that the combustible mixture of fuel and air is subjected to relatively higher engine cylinder pressures or complete combustion (this is known as "residence time") [14].

On the other hand, two concepts can describe the engine load. (1) Torque, which is a measure of an engine's ability to do work, and (2) power, which is the rate at which work is done. For stationary, spark-ignition engines, pressure data for the gas in the engine cylinders can be used to calculate the work transfer from the gas to the piston. Having this relationship, calculation of the useful power delivered by the engine to the load is obtained [13,14]. Commonly, the engine load can be expressed as power (Horsepower or Kilowatts) or as the Brake Mean Effective Pressure (BMEP) obtained with the following equation [14]:

$$BMEP\wedge \quad Vd \qquad (L\ 3)$$

where r is the brake output torque, and *Vd* is the displacement per cylinder.

The effects of speed and load variations on NO_x and HC emissions are shown in Figure 1.4.



(a) *(b)*

Figure 1.4. Variation ofHC andNO$_x$ emissions in a SI engine with (a) engine speed at load=380 kPa, and (b)load or indicated mean pressure (imep) at speed=1250 RPM. The equivalence ratio is 0=0.9 [13].

Figure 1.4.a. shows how NO_x concentrations increase moderately with increasing the speed at constant load. NO_x concentrations increase dramatically when increasing the load at constant speed as depicted in Figure 1,4.b. Conversely, HC concentration changes are opposite to the NO_x concentration trends.

Engine speed and load affect CO concentrations by varying the cylinder temperatures during combustion. The importance of combustion temperatures to CO and HC emission levels is illustrated on Figure 1.5.

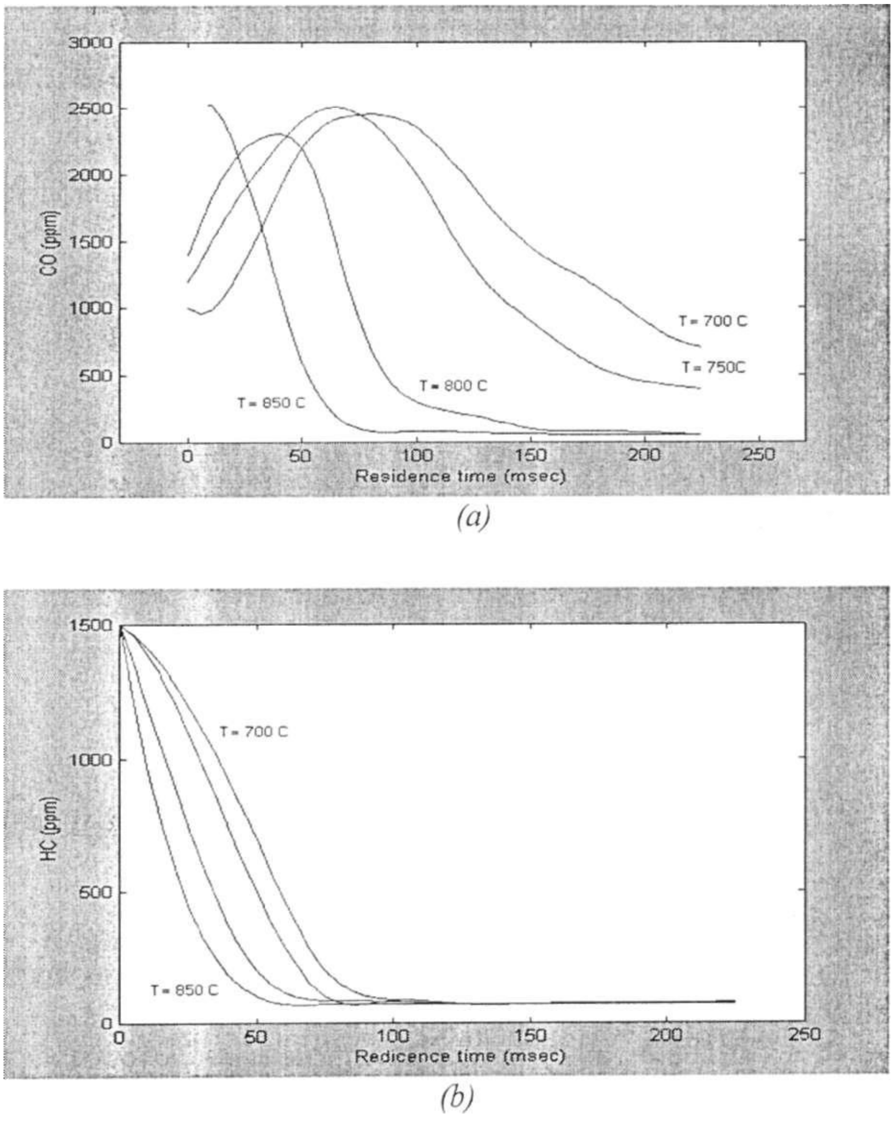


Figure 1.5. Effect of exhaust gas temperature on (a) CO and (b) HC emission levels [13].

CO and HC concentrations are presented as functions of residence time at different exhaust gas temperatures. With higher combustion temperatures, the residence time required for complete combustion decreases. As a result, CO concentrations approach zero in a shorter period of time [13].

C. Spark timing

Spark timing is defined as the time at which the ignition system provides the electrical energy to the spark plugs into the engine cylinders to start the combustion process per operating cycle [2]. It is expressed in degrees with respect to TC (before TC or after TC). Spark timing significantly affects NO_x emission levels. Advancing the timing so that combustion occurs earlier in the cycle increases the peak cylinder pressure. On the other hand, retarding the timing decreases the peak cylinder pressure. Higher peak pressures result in higher peak combustion temperatures, and hence higher NO_x emission rates. For lower peak cylinder pressures, lower NO_x formation rates result. Figure 1.6 shows typical NO_x emission data for a SI engine as a function of spark timing [14]. NO levels correspond to more than 90% of NO_x levels.

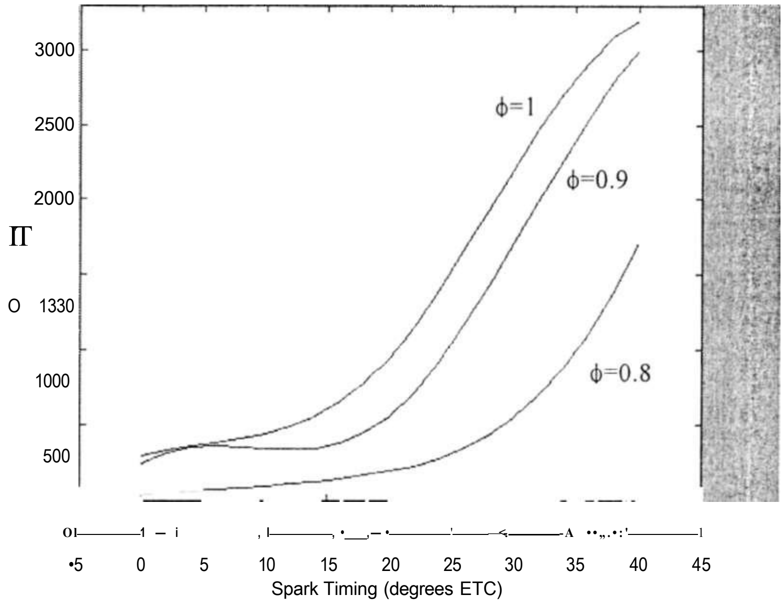


Figure 1.6. Variation of NO concentration with spark timing retard. Three curves represent variation of NO levels at three different ϕ=1, 0.9, 0.8 [14].

As mentioned before, spark timing affects combustion temperatures. Therefore, HC and CO concentrations vary depending on the timing. When timing is advanced, HC and CO concentrations decrease because more time is allowed for complete combustion, and therefore higher combustion temperatures are experienced. Conversely, when the timing is retarded, lower combustion temperatures are experienced, and HC and CO concentrations augment. This effect is observed on Figure 1.5. At lower combustion temperatures (T= 700 C), HC and CO emission levels are higher than at higher combustion temperatures (T=850 C).

1.2. Gas Emission Technology

This section discusses primarily the fuel and air engine systems which are responsible for the correct air/fuel ratio control. Strategies to achieve a reliable air/fuel control are briefly described. In addition, an introduction and brief description of the spark timing system is provided. Some other effects that various control strategies might have on the emission level concentrations, and on the overall engine performance, are also mentioned.

1.2.1. Engine control systems

A. Fuel and Air system

The fuel system is responsible for delivering the correct amount of fuel to the engine to achieve the desired engine speed [2]. Commonly, a fuel governor, which can be mechanical or electronic, controls a fuel valve that allows the fuel to enter into the combustion chamber or cylinders. In engines with carburetor(s) *, however, the governor controls the amount of air/fuel mixture delivered to the intake manifold, so air and fuel are mixed before reaching the cylinders.

During starting, the fuel must be controlled carefully to ensure reliable starting. Measurement of the fuel is necessary to determine the amount of air needed to reach a specific air/fuel ratio.

The air system is in charge of delivering the correct amount of air into the cylinders [2]. It is directly related to the load driven by the engine. A simplified air/fuel system is shown in Figure 1.7. A brief general description of the operation of the overall system is provided here for a carbureted system.

(1) Air is filtered before entering to the turbocharger .

* In the present work, a fuel/air system using a carburetor was used.

* A turbocharger is a unit supplied with energy by a turbine driven by the engine waste gases. It is used to boost the air pressure supplied to the intake manifold, and it is driven the engine exhaust gas pressures.

(2) The fuel is supplied and controlled by a fuel pressure regulator based on the air/fuel ratio and speed desired.

(3) Both air and fuel are mixed in the carburetor.

(4) The governor regulates the amount of air/fuel mixture entering the intake manifold.

(5) Air/fuel mixture is equally distributed into the engine cylinders, combustion takes place and residual gases leave the combustion chamber.

(6) Parts of these residual or waste gases are used to drive the turbine of the compressor, and all the gases are released into the environment.

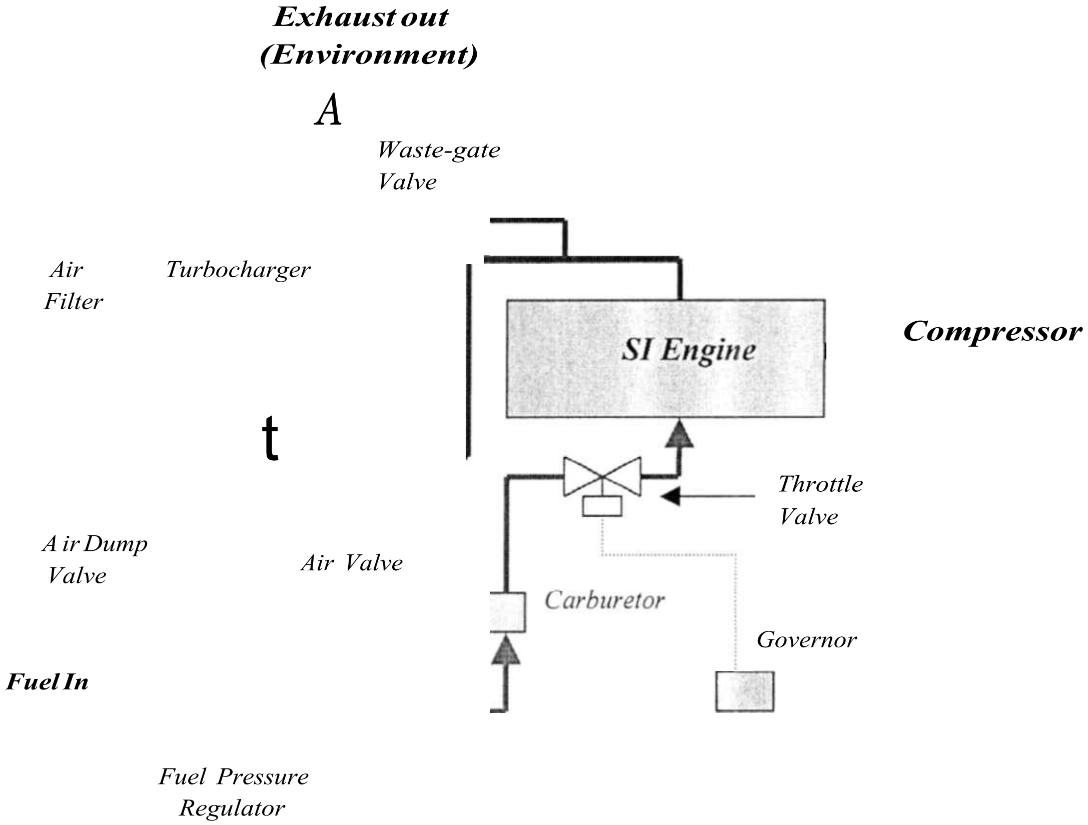


Figure 1.7. A simplified air/fuel system

There are two basic control strategies for air/fuel control: (1) closed loop; and (2) open loop control.

The closed loop control strategy is based on an oxygen sensor located in the exhaust system as shown in Figure 1.8. This control method is normally used for control of engines equipped with catalytic converters, which are active materials in a specially designed metal casing that reduce NO_x, HC or CO emissions.

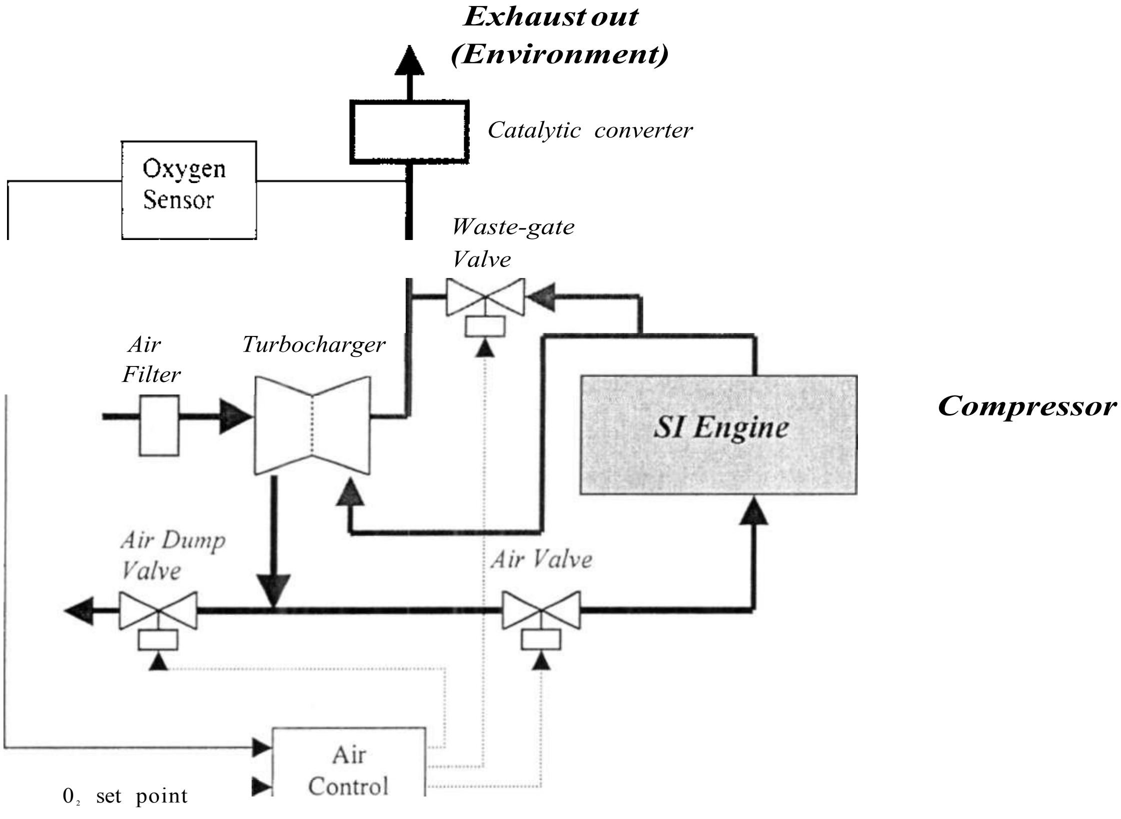


Figure 1.8. Closed loop control ofthe air system [2]

The oxygen sensor measures the amount of oxygen present in the exhaust system after combustion occurs. Therefore, having this amount of O2, the air/fuel ratio can be carefully controlled. However, it is important to note that current heated exhaust gas oxygen (HEGO) sensors limit the control to equivalence ratios very close to the stoichiometric point [2]. Problems are often encountered due to the unreliability of the HEGO sensors used in heavy-duty SISIC engines. In addition, operating the catalytic converter outside the stoichiometric point results in serious loss of efficiency as illustrated in Figure 1.9.

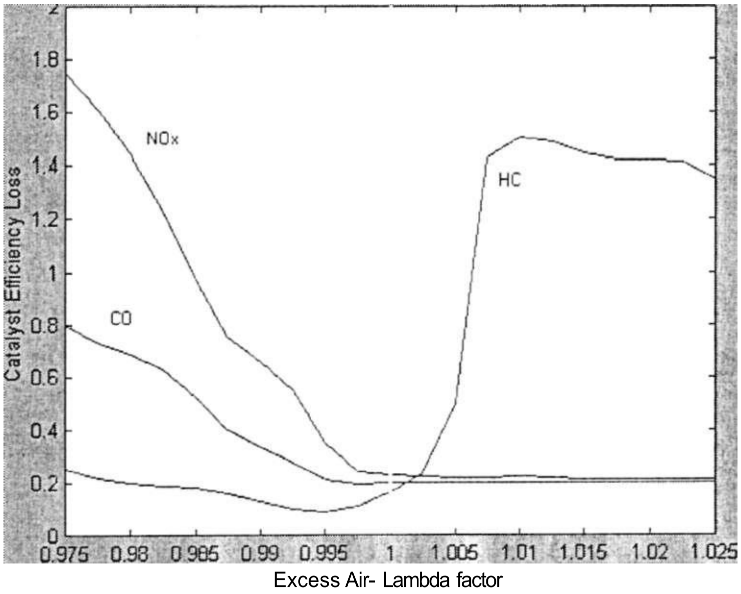


Figure 1.9. Three-way catalytic converter efficiency [14]

For the open loop control solution, an air set point is established either according to a calculation or a performance map based on a range of input sensors. As stated before, the amount of air required depends on the amount and type of fuel, the desired air/fuel ratio and the load. If the fuel composition is constant, the fuel amount may be determined from the volumetric flow, the heat content, and the fuel temperature [1,14]. If the ratio of combustible elements in the fuel varies, a better alternative is to use the fuel mass measurement [13]. If the fuel is mixed with a variable amount of non-combustible gases such as N_2 or CO_2, either these elements must be measured online or closed loop control is needed. The amount of air may be determined either with an air flow meter or by calculations from the air manifold pressure and temperature [1]. Figure 1.10 shows an air/fuel ratio control based on the open loop concept. The amount of air is calculated by measuring the fuel flow or pressure, the fuel temperature and air parameters such as the air manifold temperature and pressure. Based on the air set point, the amount of air is controlled by regulating one or more of the air valves, air dump valve and waste-gate valve, previously discussed.

Some systems do not use catalyst for emission reduction . Instead, the engine control system is designed to operate the engine near the lean limit, and some additional factors need to be considered. At lower engine load and speed, when the fuel concentration is lower, the lean limit occurs at lower value of *X,* so the excess air amount must be adjusted according to the load.

The main advantage of using the described open loop control system is its independence on utilizing HEGO sensors to set the appropriate air pressure to reach the

The present engine management system (REMVue) operates the engine in the lean area.

required air/fuel ratio. However, field experience shows that present HEGO sensors fail within 3 months when used with SISIC engines [Malm, H, REM Technology Inc., B.C. Canada].

Therefore, combining open and closed loop control offers the advantages of both approaches while lessening the total dependence on a single oxygen sensor.

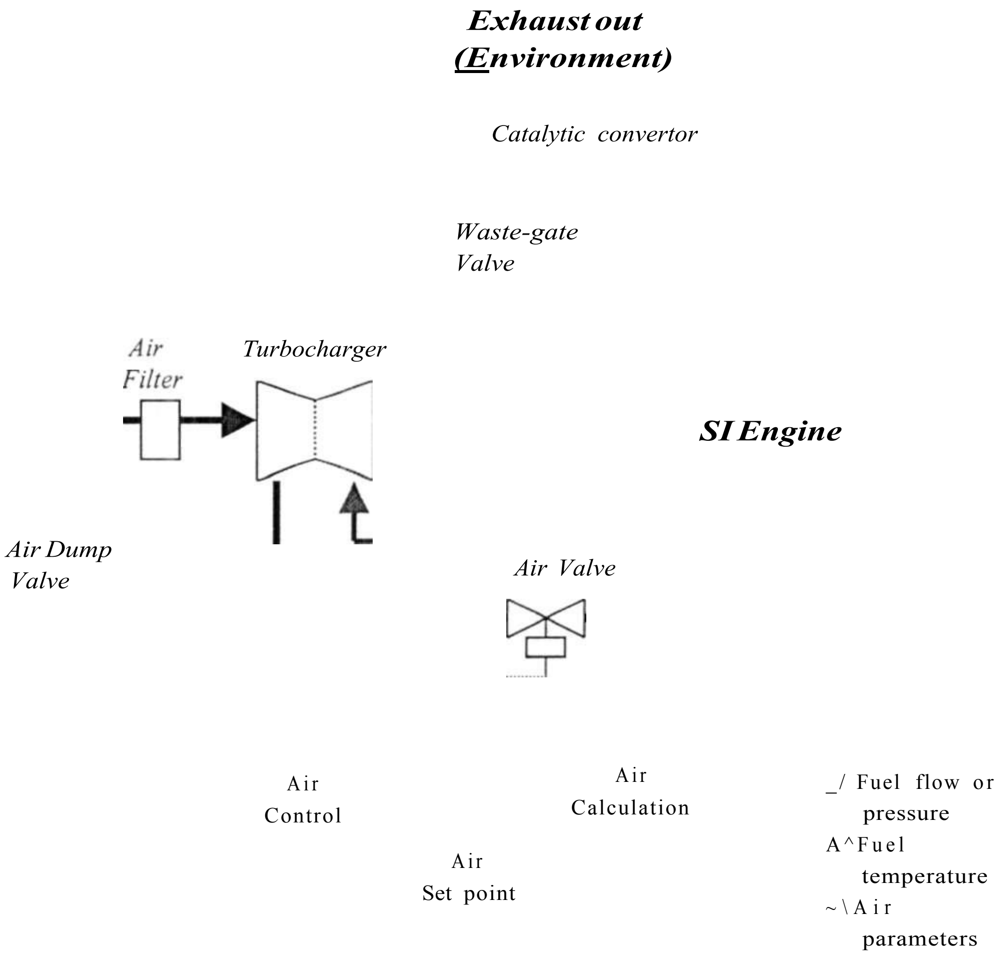


Figure 1.10. Open loop control of the air system

Additional strategies to reduce these effects are: (1) utilization of second oxygen sensor after the catalytic converter, and (2) combination of open and closed loop control where the air/fuel ratio is controlled according to an equation or performance map and trimmed by the oxygen sensor reading [2]

B. Ignition system

The ignition system provides the electrical energy to the spark plugs at the correct time to initiate combustion. Commonly, ignition timing is fixed except during starting the engine. Improvements in fuel economy, knock (uncontrolled combustion or auto-ignition) and/or emissions can be achieved with the optimization of ignition timing; however, not all of these conditions can be optimized at the same time. The ignition timing depends on speed, load, fuel-air mixture and cylinder temperatures. In order to find an optimal timing, a timing map is needed, i.e., variation of the spark timing at different engine loads, speed and air/fuel ratio must be performed, and the final decision is taken based on which parameter needs priority optimization [1]. Since knock can damage the engine, either a timing map must be constructed to avoid autoignition or the inputs from the knock sensing system must have priority control [1]. However, the importance of emission control is of great concern, and a timing map has to include this factor. Figure 1.11 shows a typical ignition control system.

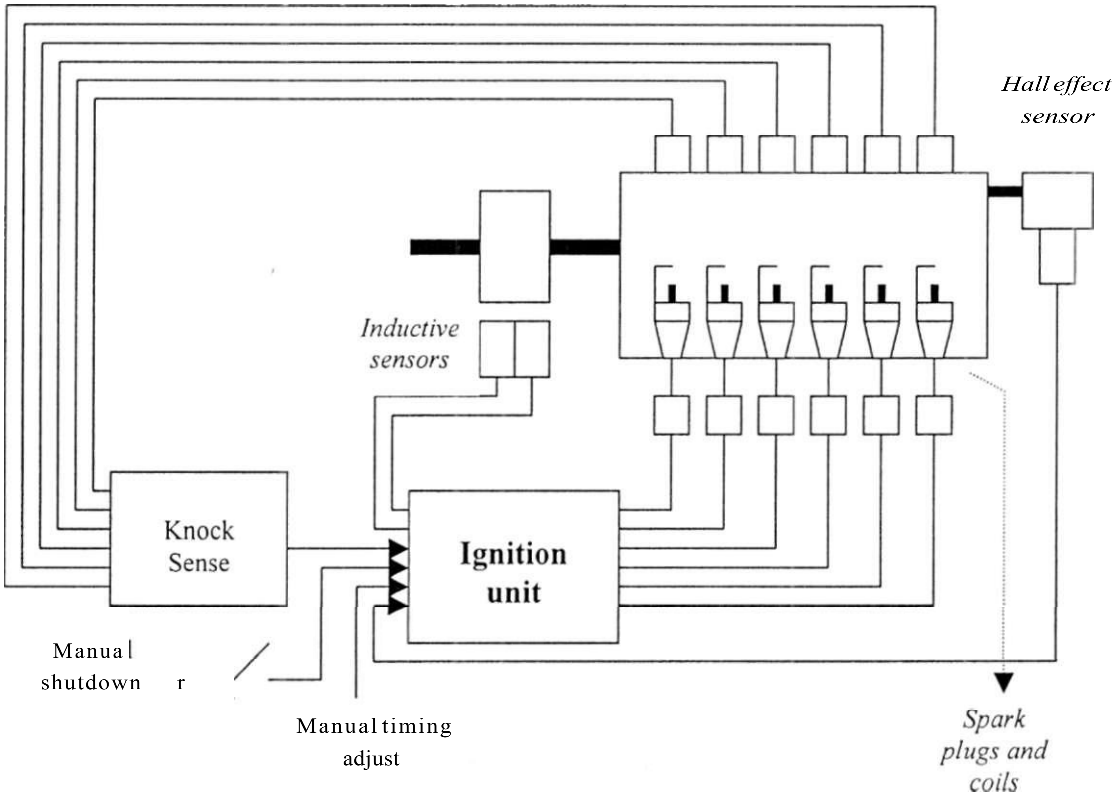


Figure 1.11. Typical ignition unit for SI engines [2]

The ignition unit provides the energy to the spark plugs based on the conditions detected by the knock system, the signals coming from (a) the inductive sensors, (b) the Hall-effect sensors, and (c) the engine management system corresponding to a possible manual shutdown, and a continuous manual timing adjustment.

Basically, the ignition system needs timing marks to function. The inductive and Hall effect sensors are used to obtain signals from the engine-rotating shaft. These signals provide (1) a reference angular position; (2) a desired timing mark for the ignition signal; and (3) a signal to distinguish between the power revolution (compression and combustion) and the breathing revolution (exhaust and intake) where no spark is required (Malm, H., R E M Technology Inc., Port Coquitlam, B.C., Canada).

1.2.2. Gas Sensors

Contemporary toxic gas sensors, known as electrochemical sensors [21,22], are capable of monitoring specific toxic gases (e.g., NO_x, HC, CO). Electrochemical sensors can be classified as (1) fuel cell; (2) semiconductor; and (3) galvanic sensors. They all work based on the principle illustrated in Figure 1.12. The gas being measured diffuses through a porous membrane and reacts with the reactive electrode (WE). Then, oxidation or reduction of the measured gas generates DC electricity. The amount of current generated is proportional to the amount of gas consumed.

Fuel cell sensors are miniaturized fuel cells mainly made of hydrogen or methanol that react to low concentrations of gas (0-100 ppm). Semiconductor sensors are based on metal oxide, which is heated from 300 to 400 °C by a heated coil. In the presence of gases the conductivity of the metal oxide changes and quantification of the gas is possible. The average life of these sensors, the fuel cell and the semiconductors, is between 3 and 4 years [22,23].

Galvanic sensor cells are made of metal and their lifetime is governed by the amount of gas they absorb [23]

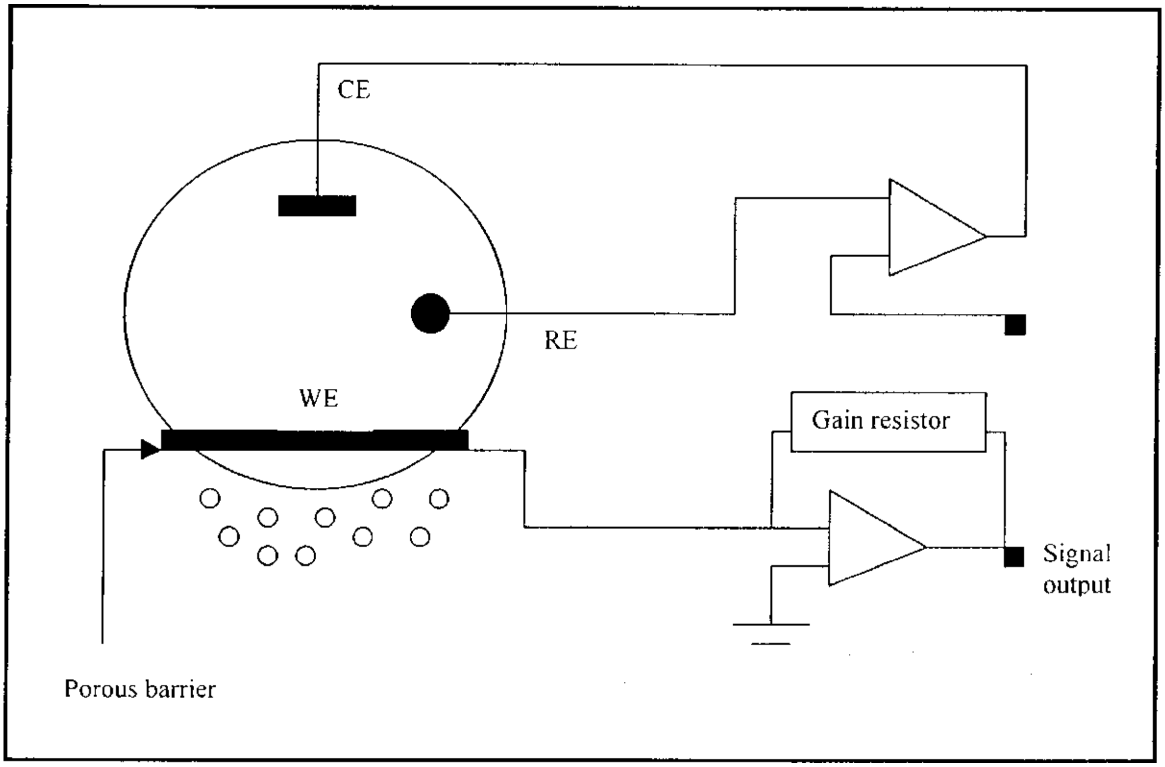


Figure LI 2. Operationalprinciple ofan electrochemical sensor [23]

Presently, gas sensor technology is capable of monitoring a wide range of gas concentrations (0-4000 ppm) in environment temperatures ranging from -10 to 40 °C [22, 23]. The automotive industry has always been on the leading of gas sensor technology, due to the fact that it contributes significantly to air pollution. Based on the requirements to monitor and control transportation emission levels, gas sensors capable of measuring concentrations from 0 to 1000 ppm at high temperatures (100-300C) have been launched into the market [23]. However, none of the sensors is able to operate reliably at higher temperatures (400 to 600°C), which are normal cylinder and exhaust temperature in stationary industrial engines.

A great amount of research has been focused on developing micro-fabricated and micro-machined semiconductor sensors to minimize size, weight, and power consumption while providing a reliable, accurate operation in hostile conditions [19]. A promising technique that has been investigated to quantify NO_x and HC gases is using Schottky diode sensor structures for aeronautic and aerospace applications [24].

CHAPTER II: PARAMETRIC EMISSION MONITORING SYSTEM BASED ON MULTIDIMENSIONAL ARRAYS

2. Parametric Emission Monitoring Systems (PEMS)

PEMS have their roots not just in the gas and oil industry, but in the manufacturing industry as well. Pulp and paper mills, as well as chemical plants, employ PEMS in combustion and non-combustion processes to calculate emissions [11]. The term PEMS originally stood for Predictive Emission Monitoring/Measuring Systems. Later, the term Predictive was dropped in favor of the more correct term "Parametric". Furthermore, the Environmental Protection Agency in its well-known Method 19 for Engine Exhaust calculation uses this term [25].

PEMS applied to natural gas-fueled engines are somewhat new, but the concept is mature enough and established in terms of its effectiveness. A PEM system, sometimes called an Alternative CEM system (ACEMS), is based on key measurements that affect and control the combustion process, and thus define the combustion emissions [11]. PEMS can be designed to monitor few or many parameters, depending on the emission species of interest, the type of PEMS, the accuracy of control, and the broadness of the engine's operational envelope.

There are two types of PEMS: empirical and model based. The empirical type, usually called matrix-based, is based on look-up tables (matrices) containing data of measured engine performance. The model type of PEMS generally requires parameters which serve as inputs into equations derived from the laws of chemistry, physics and thermodynamics, to calculate engine parameters related to emission generation.

The empirical types of PEMS are developed by collecting key data (mapping) from each engine type over its entire operating range, both when it is meeting the control algorithm set points and when it is not, i.e., when the engine is running tuned or non-tuned.

The model-based PEMS involve few key engine measurements that compute variables such as air/fuel ratio, combustion temperature, etc [8,11]. A bulk combustion temperature model-based PEMS uses an in-cylinder combustion pressure sensor as an input to the ideal gas law and chemical kinetics equations to compute NO_x, CO and HC concentrations [11]. Although model-based PEMS require fewer parameter inputs, they do not provide diagnostic capabilities (see example below) that empirically-based PEMS do, and profound combustion knowledge is required for their development.

Most contemporary PEMS require diagnostic capabilities, i.e., detection of any failure in the engine instrumentation system directly related to the emission control. A cylinder temperature might dramatically decrease, and a model-based PEM system will compute and obtain lower NO_x concentrations due to the fact that no previous engine information is provided, while an empirical PEM system will immediately notify the user that failure of the spark ignition system has been experienced due to the deviation between the expected (engine information is available due to mapping) and the actual cylinder temperature.

In a pure model-based PEM system, a single parameter can be changed without affecting any other parameters and a pure cause and effect noted from the emissions can be obtained. For example, in a turbocharged-engine the air manifold pressure (AMP) can be boosted by 50%, while the air manifold temperature (AMT), the exhaust pressure, and air/fuel ratio remain unchanged. In a real engine, this scenario is unachievable since AMP, AMT, air/fuel ratio and exhaust pressure are all somewhat linked and pure singular cause and effect relationship to emissions cannot be easily determined. Conversely, an empirical type of PEM system is capable of simulating these secondary effects, thus reflecting a more realistic behavior of the engine [8,11].

The overall goal of any PEMS development is to invent systems that use as few inputs as possible, while maximizing the amount of information possible, yet making the system universally applicable to any engine.

2.1. PEMS: Advantages and Disadvantages

The main difference between PEMS and CEMS is that PEMS are intelligence (software) based and CEMS are hardware based.

Some other advantages and disadvantages that PEMS present can be listed as follows:

- **PEMS Disadvantages**

The first area of concern is the cost to install and maintain PEMS. PEMS are considered more costly and complicated than Demonstrated Compliance (DCPL) systems or portable analyzer emission testing programs. It is usually the engine mapping that drives up the PEMS cost. However, the more similar the engine types, the lower per unit cost for the mapping. They cost less than CEMS, but are substantially more expensive than portable emission analyzer programs. In addition, PEMS require Quality Assurance/Quality Control (QA/QC) certification to verify correctness, and a rigorous control system [11].

In the future, low cost CEMS sensors might become available and might replace PEMS. However, experts comment that sensor technology will enhance not only CEMS but also PEMS, offering more powerful tools for emission monitoring and control [11,26].

- **PEMS Advantages**

PEMS are not limited by the lifetime of chemical sensors. They can be implemented by using a gas analyzer to record the different emission levels only when engine mapping is performed. Most PEMS are commonly applied to NO_x measurements, but in theory, one could create a PEM system for any chemical species of concern, such as CO, or HC.

PEMS can be potential candidates for replacing existing emission monitoring system for all engine types, including gas turbines. As a result, PEMS are candidates for monitoring other processes associated e.g. with boilers or gas dehydrators, among others. Secondly, they have a low maintenance cost once installed. Additionally, PEMS demonstrate continuous compliance, and can provide a standard format report indicating emission, engine operation and instrumentation system diagnostics.

PEMS can be configured to generate accurate reports of emission levels both when the engine is operating at set point conditions and at offset point conditions. Furthermore, they can provide data on NO_x, CO and HC along with the ability to provide data on other selected emissions.

Since PEMS are parameter-based, when a PEM system reports a specific emission level, it can also in varying degrees be able to indicate the parameter or parameters driving emissions out of the normal range. The greater the number of parameters, the more powerful the diagnostic capability is [11].

The ability to diagnose an engine remotely, via a computer and a modem, is becoming a necessity. Specialists are looking for monitoring engine performance, and adjusting certain key control parameters to comply with the emission requirements in real-time. Therefore, PEMS might be the solution for making this feasible at very low costs.

2.2. PEMS standards

Some of the PEMS practitioners are of the opinion that each and every engine must be mapped [11]. For the engines that have multitude parts or components that no longer meet design specifications, this might be the case. However, this approach leads to profound complications, because it means that a new engine map is required every time an engine component is changed. Nevertheless, a strong case can be made for bringing all similar engine types to the same pre-defined mechanical condition, so that their combustion fingerprint is the same. This does not mean that the engine must be working perfectly, but it does require that the engines should be maintained to a reasonable and achievable pre-defined mechanical standard. Thus, if the mechanical condition of the engine is known and the control system accurately controls key engine parameters, the emissions can be accurately predicted.

2.3. Parametric Emission Monitoring System for Estimation of Gas Emissions for Stationary Internal Combustion Engines based on multidimensional arrays.

2.3.1. Introduction

Typically, quantification of emission pollutants from SISIC engines has been accomplished using a collection of expensive equipment to sample, analyze and provide a continuous record of emission rates [1]. On the other hand, various PEM models have been developed to calculate these emissions based on chemical and thermodynamic equations, which are too complex to be practically applicable for continuous monitoring [12,27,28]. As mentioned before, PEMS can be designed to estimate the distinct emission levels based on a number of operational parameters already recorded from these machines for control compliance.

Heater et al [8] proposed a parameter-based method for calculating exhaust emissions from reciprocating natural gas transmission engines, achieving relative accuracy derived from CEMS standards due to the fact that standards for testing and evaluating PEMS are not yet available. Buchop et al. [7] presented a patent providing a reliable and accurate PEMS for stationary engine/compressor units coupled to a pipeline based on an emission matrix primarily as a function of engine speed and engine torque.

This section describes the development of an empirical-based PEM system capable of estimating emission levels from SISIC engines based on multidimensional arrays. The present system differs from the existing PEMS for SISIC engines since it utilizes different engine parameters, and compares estimated and actual values between some pre-selected engine parameters (see Section 2.3.3) for estimation adjustments. Thus, a universal design of a parametric model, which is capable of calculating emissions produced by natural gas SISIC engines without the need of expensive gas sensors, can be implemented. In the development of the model control parameters, monitoring parameters, and calculated parameters were utilized. Using data from a tuned engine*, 4 multidimensional arrays were created containing emission levels as functions of selected operational parameters. Interpolation methods were used to fill these arrays, thus increasing model resolution.

Recorded data from a non-tuned engine was used to expand the model. The approach can dynamically estimate harmful emissions from SISIC engines and offers the potential to eliminate the need for site compliance monitoring.

2.3.2. **Methods**

A. **Parameter definition**

Based on the theory of PEMS [11], which stipulates that there is a relationship between emissions produced by an engine and certain key engine operating parameters, identification of these important parameters was performed, involving engineering fundamentals of the internal combustion engines, existing data from different SISIC engines, and experience provided by those skilled in the art.

Three categories of operational parameters were established: (1) control parameters, aiming at obtaining a specific performance of the engine, (2) monitoring parameters, resulting from the response of the engine to a set of control parameters, and (3) calculated parameters resulting from empirical calculation of various combinations of both the control and the monitoring parameters.

* Waukesha, 16 cylinder, Model P9390GSIU, Rated power: 1800HP

Engine parameters selected for the estimation of the emission levels were (a) control parameters: speed, air-fuel ratio (lambda factor), load, and spark timing (ST); (b) monitoring parameters: fuel flow rate and exhaust temperature (or combustion temperature); and (c) calculated parameters: brake specific fuel consumption (BSFC). In some cases, engine load cannot be controlled or measured. For such situations, required engine parameters should encompass air intake manifold pressure, fuel flow, speed, and exhaust temperature.

It is important to monitor the fuel flow rate, since it allows for the calculation of the brake specific fuel consumption (BSFC), which determines whether the engine is operating tuned or non-tuned. Ideally, an engine is expected to operate at the most efficient consumption rate calculated at specific load, speed, and air/fuel ratio.

On the other hand, NO_x, HC and CO emission rates are affected by different operational parameters, which directly affect the temperature inside the combustion chamber. In the present engine system, exhaust temperature is measured and used to estimate the combustion temperature. Therefore, exhaust temperature is included as a relevant monitoring parameter that allows for adjustment of the emission level estimation.

Brake-power (BP) is a calculated parameter, which is related with the engine load. Although it can be set to obtain a specific response from the engine, it is empirically calculated using data from the compressor unit, which the engine drives.

B. **Multidimensional arrays**

Multidimensional arrays are an extension of the ordinary two-dimensional matrix. Matrices have two dimensions: the row dimension and the column dimension. Access to a two-dimensional matrix element is possible by using two subscripts: the first representing the row index, and the second representing the column index. Multidimensional arrays use additional subscripts for indexing. A three-dimensional array, for example, uses three subscripts. The first subscript references array dimension 1 (the row). The second references dimension 2 (the column), and the third references dimension 3 (the depth). Figure 2.1 uses the concept of a page to represent dimensions 3 and higher.

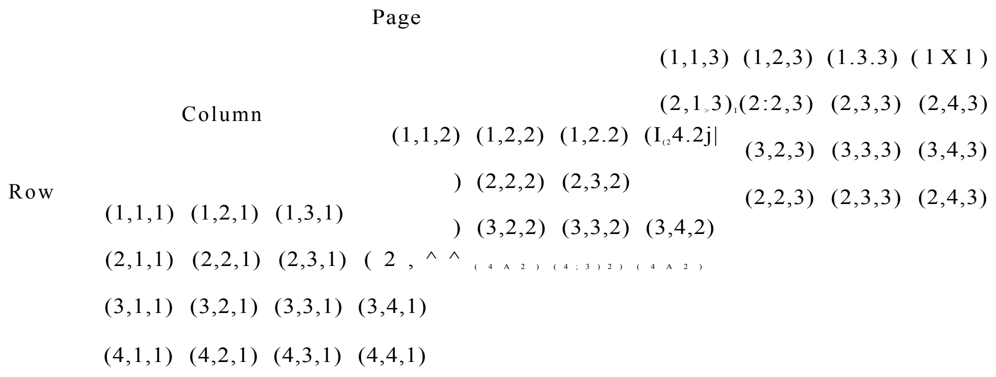


Figure 2.1. Multidimensional (3D) Array

Multidimensional arrays have been used in several applications [29,30], which require building multidimensional and comprehensive databases. Such arrays are directly applicable in the present project, creating the opportunity for inexpensive feasibility testing of the suggested parametric model. However, their cumbersome interaction with the computer memory and the need for intervalue interpolation in order to improve the resolution of the model might jeopardize the practical implementation of this approach, limiting the number of parameters in the model and/or affecting the real-time requirements. Figure 2.1 presents an indexing diagram of a three-dimensional (3D) array.

2.3.3. Implementation

A. **Multidimensional array approach**

A practical approach, which offered simplicity and low cost, was to construct a number of multidimensional arrays, which represented the emission levels in response to manual control of the crucial operational parameters mentioned before. Following the idea given by Heater et al. [8], the first step was to develop a model with fewest number of parameter inputs, while achieving a given accuracy. As explained earlier, three types of parameters were defined: (1) control parameters, (2) monitoring parameters, and (3) calculated parameters. Subsequently, 3 multidimensional arrays were created using data from a tuned engine, containing (1) emission levels as functions of a predetermined set of control parameters, (2) calculated parameters as functions of a given set of monitoring and control parameters; and (3) monitoring parameters as functions of a given set of control, monitoring, and/or calculated parameters. In order to fill these arrays, a recording-data protocol was established.

Four basic operation intervals or ranges for setting a natural gas SISIC engine were established. (1) Speed: from 700 to 1200 RPM, (2) Load: from 100 HP (just sufficient to overcome friction and accessory load) to 1200 HP, (3) Lambda factor: from 0.95(rich burn) to 1.75(lean limit), and (4) Spark timing: from 20° before Top Center (bTC) to 27°bTC (See Appendix B). The decision to operate the engine with an overall lean mixture was made due to the facts that NO_x production is strongly temperature dependent, and that the exisitng engine management system was designed to achieve engine control within the lean range. Therefore, working in a lean zone causes combustion chamber temperature to decrease and leads to the reduction of the NO_x levels. No catalyst was used to reduce NO_x levels.

Having the possibility of setting these four operating parameters manually, and working with a tuned engine, a fixed number of combinations were set and data corresponding to these pre-selected monitoring parameters were recorded. Data, including the different values of the recorded sets of control and monitoring parameters, was entered into the model. Calculated parameters were computed using empirical equations defined in the literature [14] making this PEM system not only empirical-based but also model-based [11]. After this calculation, a complete set of data representing the overall behavior of the engine was obtained. In order to increase the resolution of the model, well-known interpolation methods for building the multidimensional arrays were used. Extension of the model was performed by recording data from a non-tuned engine so that emission estimation can be achieved for any engine condition, and diagnostic capabilities to detect failures in the engine instrumentation system can be included in the future.

The first multidimensional array provided preliminary emission estimation as a result of manually setting four control parameters: lambda factor *(k),* speed, load or brake power (BP), and spark timing (St). These arrays can be represented as follows:

$$\boldsymbol{NO_x(X, Speed, BP, St)}$$

$$\boldsymbol{HC\,(X, Speed, BP, St)}$$

$$\boldsymbol{CO\,(X, Speed, BP, St)}$$

The second multidimensional array contained data representing calculated parameters such as BSFC and BP as functions of the control parameters: lambda factor, speed and ignition timing, and monitoring parameters such as compressor discharge (Dp) and suction pressure (Sp). These arrays are represented as follows:

***BSFC(k,* Speed, BP, St)**

***BP (I,* Speed, Sp, Dp)**

The third multidimensional array incorporated monitoring parameters, e.g. average exhaust temperature (Avg. ExhT), as a function of control parameters and calculated parameters, e.g., lambda factor, speed, load, and ignition timing. Right and Left Exhaust temperatures corresponding to the right and left engine cylinder banks were averaged:

***Avg. Right ExhT (k,* Speed, BP, St)**

***Avg. Left ExhT (I,* Speed, BP, St)**

The whole PEM model was developed using the CAD system MATLAB 5.3.1 (The Math Works, Inc., Natik, MA, USA). Simulation of the complete target operational environment was performed. After building the multidimensional arrays, a set of data that represented a tuned or non-tuned engine was simulated. This data was continuously entered into the model and dynamic estimation of the emission levels was obtained. The PEM model compared the data of some operating parameters to decide whether the engine was operating tuned or non-tuned. When the engine was working tuned, direct estimation was accomplished by presenting the resulting emission levels that corresponded to a specific parameter combination. Alternatively, when the engine worked non-tuned a preliminary estimation was obtained using the previously recorded data representing a tuned engine. Then, comparison between the actual and the expected operational parameters was performed. The differences (Differ1 and Differ2) were used to calculate the final emission levels. Previously recorded data from a non-tuned engine were used to increase the accuracy of the model.

Figure 2.2 illustrates a diagram describing the different steps performed by the developed algorithm that calculate the selected emission levels generated from a SISIC engine after the different multidimensional arrays have been built.

Figure 2.3 shows the graphic user interface developed, in which the user is able to continuously monitor the values for both the actual and the expected operating parameters, as well as the emission (NO_x, CO and THC) levels. The model can generate a three-dimensional representation of any emission level as a function of any control parameter. Figure 2.4 illustrates a three-dimensional representation of the NO_x levels as a function of engine speed and load.

In addition, the system is capable of indicating excess levels if emission limits are reached, and offers the flexibility to add other operational parameters thus increasing the accuracy of the model. It also includes self-tutorial for new users, and a calculator, which allows the user to obtain emission levels based on Method 19 of the Code of Federal Regulations (CRF) for emission pollutant calculation [25].

Read :

(1) ***I*** (2) Speed (3) St (4) CiaP (5) CoaP

(6) Avg. ExhT

Estimate:

(1) BP *(X,* Speed, CiaP,CoaP)

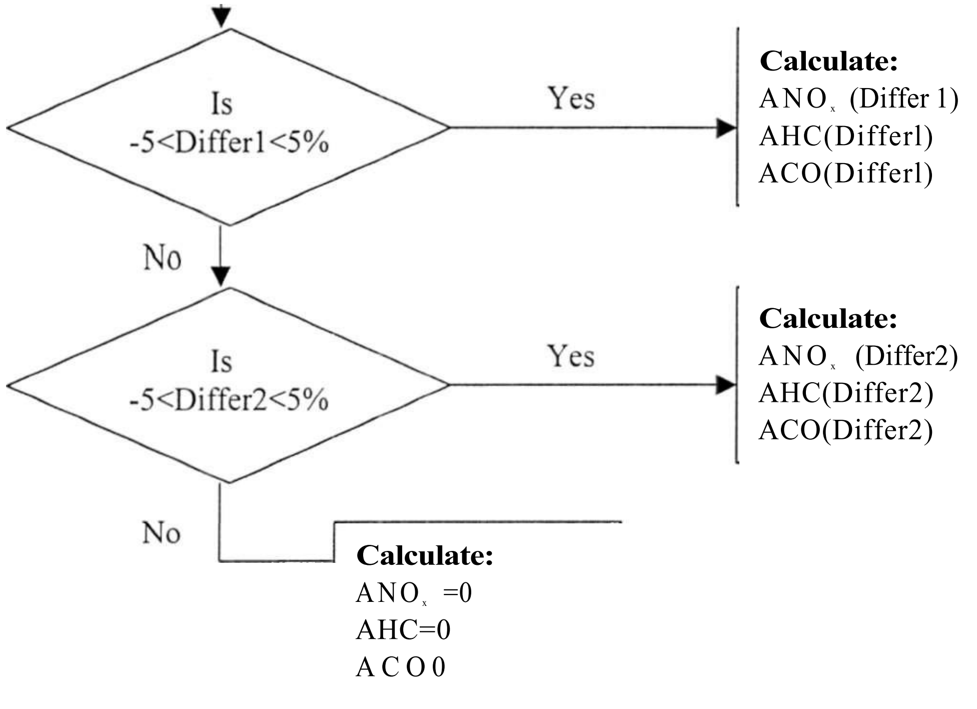
(2) Avg. Exhaust T(1, Speed, BP, St)

(3) *BS¥C(k,* Speed, load, St)

Calculate Error

(1) Differ 1= $\frac{\text{(Estimated Avg. ExhT - Actual Avg. ExhtT)*100}}{\text{Estimated Avg. ExhT}}$

(2) Differ2 = $\frac{\text{(Estimated BSFC - Actual BSFC)* 100}}{\text{Estimated BSFC}}$



Calculate:

ANO_x (Differ 1)

AHC(Differl)

ACO(Differl)

Calculate:

ANO_x (Differ2)

AHC(Differ2)

ACO(Differ2)

Calculate:

ANO_x =0

AHC=0

ACO 0

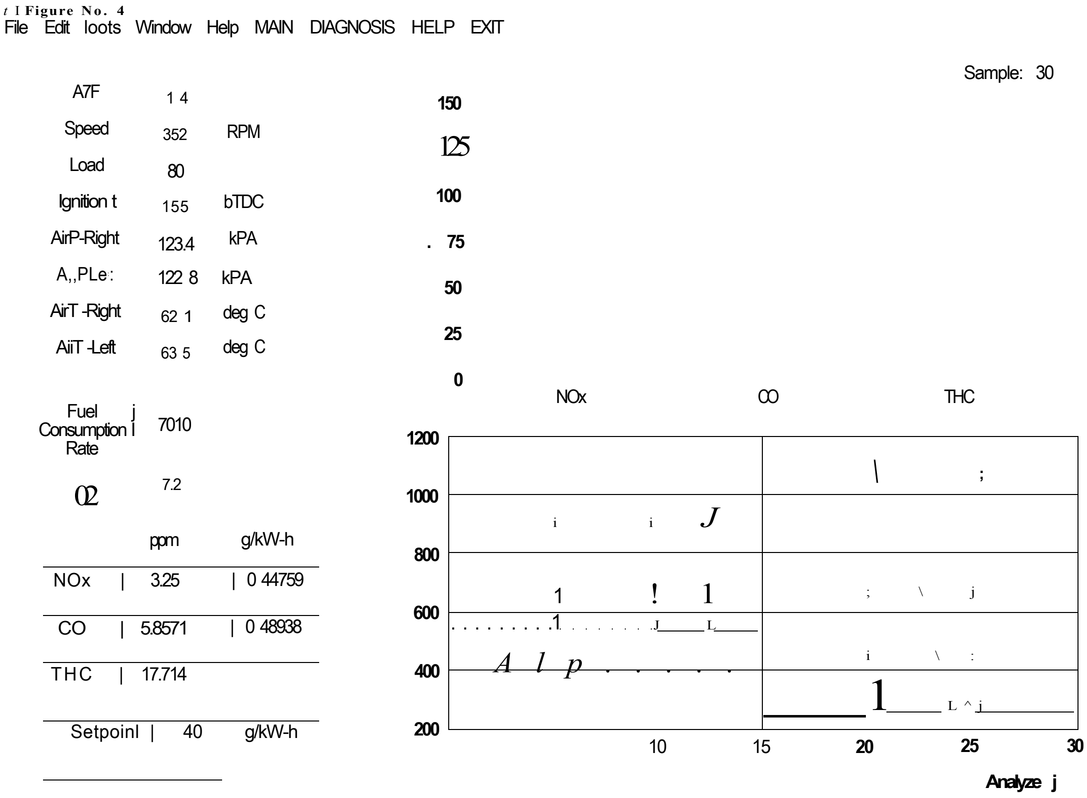
Final calculation:

NOx = NOx (X, Speed, BP, St) + ANO_x

HC = HC *(k,* Speed, BP, St) + AHC

CO = CO *(k,* Speed, BP, St) + ACO

Figure 2.2. Flow chart for emission estimation



A7F	1 4	
Speed	352	RPM
Load	80	
Ignition t	155	bTDC
AirP-Right	123.4	kPA
A,,PLe:	122 8	kPA
AirT -Right	62 1	deg C
AiiT -Left	63 5	deg C
Fuel j Consumption I Rate	7010	
02	7.2	

	ppm	g/kW-h
NOx	3.25	0 44759
CO	5.8571	0 48938
THC	17.714	
Setpoinl	40	g/kW-h

Figure 2.3. PEMS graphic user interface developed in MA TLAB 5.3.1.

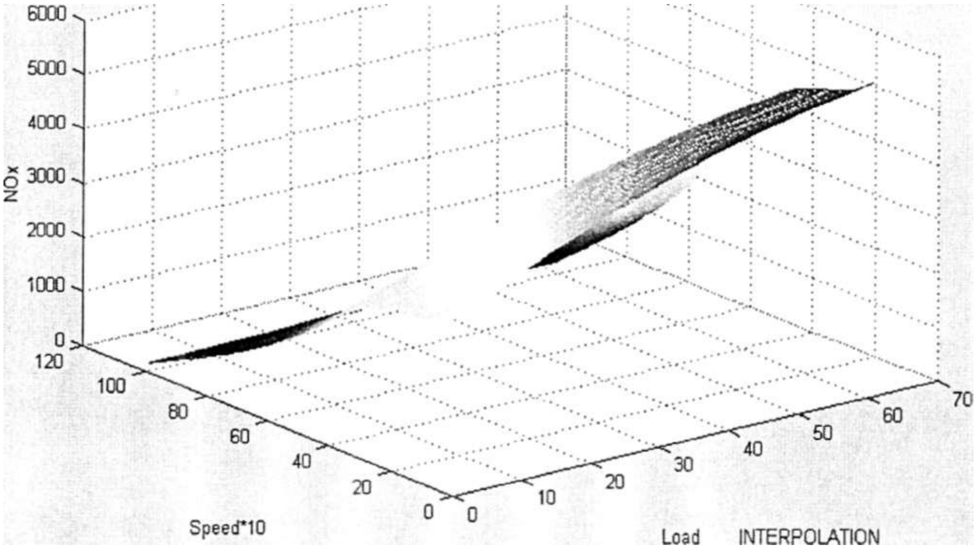


Figure 2.4. Three-dimensional representation of NO_x *estimation as a function of the control parameters using multidimensional arrays.*

2.3.4. Discussion

A universal approach for the development of a Parametric Emission Monitoring System for natural gas SISIC engines was presented. This approach can estimate harmful emissions from SISIC engines in both tuned and non-tuned states, and offers the potential to eliminate the need for site compliance monitoring.

Real-time emission estimation is limited by the extensive amount of tabular data that the system might contain depending on the required system resolution. For example, assume that an engine is desired to be mapped at the following operational conditions: (a) Lambda factor from 1.1 to 1.7, (b) Speed from 300 to 1200 RPM, (c) load 500 to 1200 HP, and (d) spark timing 19° to 27° bTC. Few samples would be enough to build the different multidimensional arrays for emission estimation. However, the size of the arrays would depend on the desired system resolution (e.g., Lambda Factor 1.1:0.01:1.7, Speed =300:0.5:1200, and Load=500:0.5:1200HP, and Spark timing=18°:0.1:27°bTC). As a result, a multidimensional array of 60x1800x1400x90 is needed in order to cover the selected range for estimating one emission level. Furthermore, the system accuracy will be jeopardized when the actual resolution of some of the input parameters surpasses the restricted system resolution (e.g., Lambda Factor= 1.452).

In contrast, this approach seems to be a very good avenue for quasi-real time emission estimation. Therefore, users can analyze in an offline fashion the emission levels for a specific SISIC engine. Resulting emission surfaces resulted to be very smooth, and resolution was improved by selecting a smaller interpolation step for each parameter. However, memory size augmented but it was not a limitation when using the system for offline emission modeling.

Based on this limitation, investigation into a different technique to estimate emission concentrations for SISIC engines was pursued, and it is presented in the following chapters.

CHAPTER III: EMISSION MONITORING SYSTEM BASED ON ARTIFICIAL NEURAL NETWORKS

3.1. Overview

In the previous chapter, a PEM system design based on multidimensional arrays was suggested and described. Limitations were encountered when evaluating the system for real-time emission estimation implementation. Because of these limitations, investigation into a new approach for developing an emission monitoring system was pursued.

The preliminary work carried by Keeler et al [9], which introduced the use of artificial neural networks (ANN) in estimating and controlling emissions produced by manufacturing plants, motivated a detailed study on the feasibility of utilizing neural networks for the present application.

Artificial neural networks have been used in various applications including pattern recognition, identification, classification, speech, vision and control. Special focus has been placed on utilizing their operational capabilities for estimation and control of nonlinear processes. Due to the fact that the combustion itself and the products of combustion (emissions) are nonlinear with respect to most measured parameters [13,14], analysis of the neural network operational principles, as well as comparison between various architectures and learning algorithms, were pursued. Design and implementation of a neural network-based model for emission estimation for SISIC engines was carried out.

3.2. Artificial neural networks

3.2.1. Definition

A neural network can be defined as a massively parallel distributed process made up of simple processing units, which has a natural propensity for storing experimental knowledge and making it available for use [31]. Its knowledge is acquired from its environment through learning processes. Inter-neuron connection strengths (known as synaptic weights) are used to store the acquired knowledge.

A neuron with a single scalar input and bias is shown on Figure 3.1.

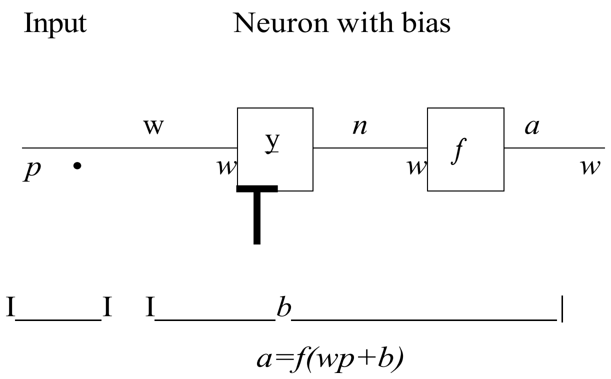
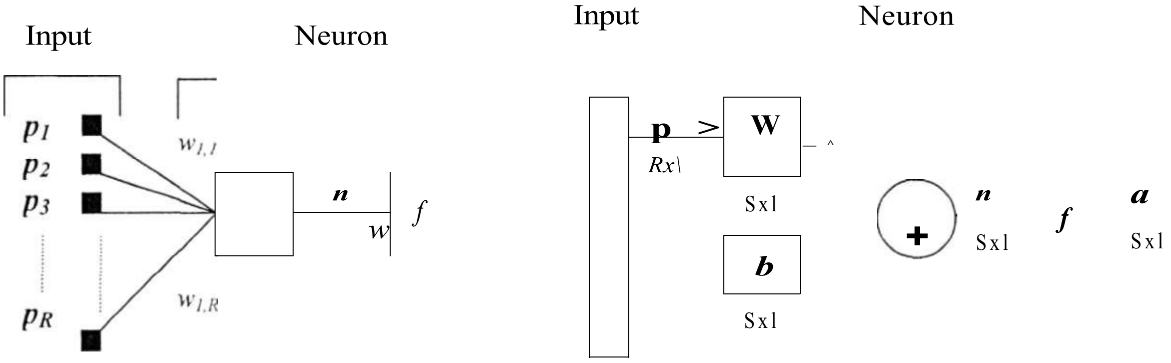


Figure 3.L Schematic of simple neuron

The scalar input *p* is transmitted through a connection that multiplies its strength by the scalar weight *w,* forming the product *wp,* a scalar as well. The bias *b,* an additional weight, is added to the product *wp.* The resulting scalar *n* = *wp+b,* is the argument of the transfer function/ which produces a scalar output *a.* Typically,/can be a step, linear or nonlinear function. The main idea of neural networks is that parameters, such as *w* and *b,* can be adjusted so that the neural network can exhibit some desired behavior [31,32].

Two representations of a neuron with a single input vector of R elements are shown on Figure 3.2.a and 3.2.b. The individual element inputs *pi, P2---PR* are multiplied by weights *wij,* ***Wij, ...WI.R,.*** The sum is simply *Wp,* the scalar product of the weight matrix *W* and the input vector *p.* The neuron has a bias *b,* which is summed with the weighted inputs to form the net input *n.* Then, $n = w/j\ p/ + w/jp2 + ... + w_{/R}PR + b$ becomes the argument of/



a=f(Wp+b)

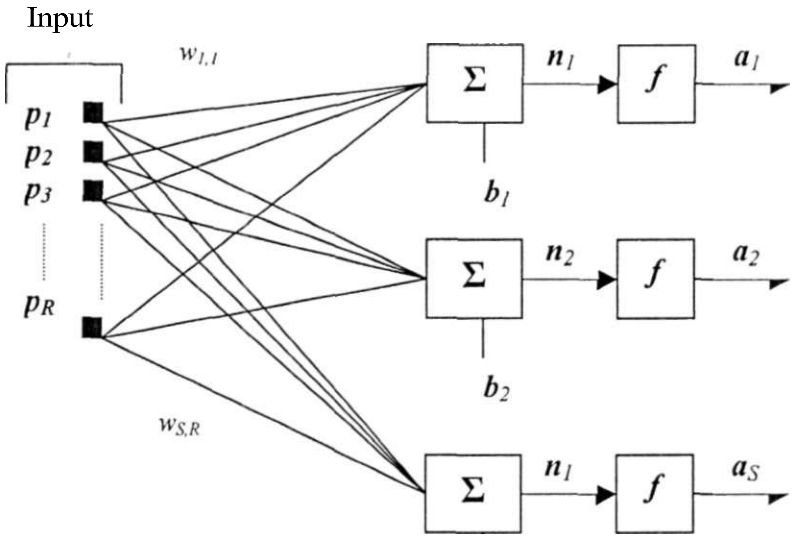
(a)

a=f(W$_v$+b)

(b)

Figure 3.2. a and 3.2.b represent two ways for representing a neuron with vector input. R is the number of elements in input vector, and S is the number of neurons in layer

A neural network contains one or more layers, each encompassing two or more neurons. Figure 3.3 presents a one-layer network with *R* input elements and *S* neurons. In this network, each element of the input vector *p* is connected to each neuron input through the weight matrix *W.* The i$^{'h}$ neuron has a summation that gathers its weighted inputs and biases to form its own scalar output *n(i).* The different *n(i)* taken together form an S-element net input vector ***n.*** Finally, the neuron layer outputs (ai,fl2,-fls)***form a*** column vector ***a*** [33].



bs

$$a=/rw_{P}+b)$$

Figure 3.3 Single-layer neural network

A neural network layer is not constrained to have the number of its inputs equal to the number of its neurons. Therefore, putting two networks in parallel can create a single layer of neurons with different transfer functions. Both networks would have the same inputs, and each network would create a specific output.

A neural network can also have several layers. Each layer has a weight matrix ***W,*** a bias vector ***b,*** and an output vector ***a.*** Figure 3.4 illustrates a three-layer neural network. This network has R^1 inputs, S^1 neurons in the first layer, S^2 neurons in the second layer, and S^3 neurons in the last layer. The outputs of each intermediate layer are the inputs to the following layer. Thus, layer 2 can be analyzed as a one-layer network with S^1 inputs, S^2 neurons, a $S'xS^2$ weight matrix W^2, an input vector a^1, and an output vector a^2.

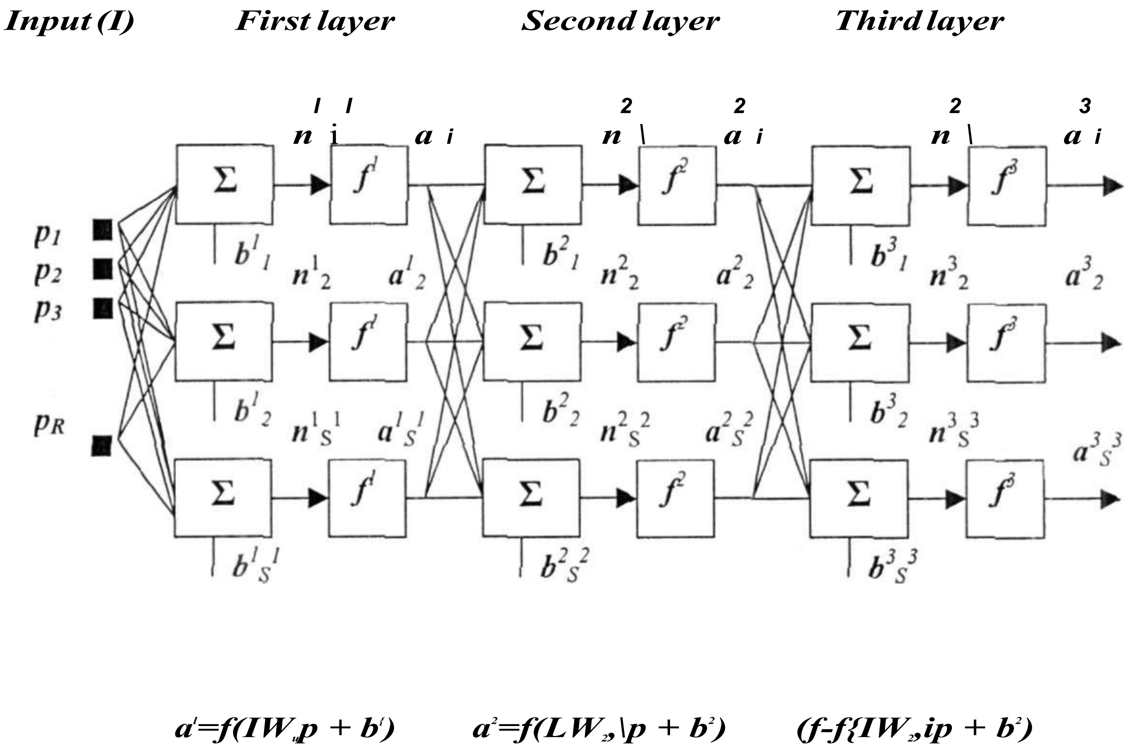


Figure 3.4. Three- layer feedforward neural network

3.2.2. Neural network knowledge representation

Knowledge refers to stored information or models used by a person or machine to interpret, predict, and appropriately respond to the outside world [31].

Knowledge representation can be considered as (1) the information that actually is made explicit or (2) the manner the information is physically encoded for subsequent use. In real-world applications of intelligent machines, it can be said that a good solution depends on a good representation of knowledge [31]. Therefore, neural networks offer the required capabilities to represent an intelligent machine. However, the possible forms of neural network representations are highly diverse, which tends to make the development of a satisfactory solution by means of a neural network a real design challenge.

The major task for a neural network is to learn a model of the world (environment) in which it is embedded, and to maintain the model sufficiently consistent with the real world so as to achieve the specified goals of the application of interest. Two types of information describe the knowledge of the world: (1) the known world state, represented by facts about what is and what has been known (prior information), and (2) observations (measurements) of the world, obtained by means of sensors designed to probe the environment in which a neural network is supposed to operate. These measurements provide the necessary information from which samples used to train the neural network are drawn.

The samples can be labeled or unlabeled. Labeled samples are represented by an input signal paired with a corresponding desired response (target output). Conversely, unlabeled samples consist of different realizations of the input signal itself. Both sets of samples represent knowledge about the environment of interest that a neural network can learn through training. A set of input-output pairs, with each pair consisting of an input signal and the corresponding desired response, is referred to as a set of training data or training sample [31].

In a neural network of specified architecture, knowledge representation of the surrounding environment is defined by the values taken on by the free parameters (i.e., synaptic weights and biases) of the network. The form of this knowledge representation involves the design of the neural network, and therefore holds the key to its performance. Different rules for knowledge representation have been extensively described in the literature [31,34]. The following four rules summarize the most important rules, which are employed in this work.

- Rule 1: Similar inputs from similar classes should usually produce similar representations inside the network, and should therefore be classified as belonging to the same category.
- Rule 2: Items to be categorized as separate classes should be given widely different representations in the network.
- Rule 3: If a particular feature is important, then there should be a large number of neurons involved in the representation of that item in the network.
- Rule 4: Prior information and invariance should be built into the design of a neural network, thereby simplifying the network design by not having to learn them.

These rules were implemented in the present approach by implementing initial preprocessing steps on the network inputs (see Section 3.3.3, Part B).

3.2.3. Artificial neural networks features and their applicability for estimating gaseous emissions for SISIC engines

Artificial neural networks offer some useful properties that match the characteristics required for modeling the emission levels generated from SISIC engines. These properties are *Nonlinearity, Input-Output Mapping, Generalization,* and *Adaptivity* [31-34] and are briefly discussed.

A. Nonlinearity: A neural network designed as an interconnection of nonlinear neurons is nonlinear itself. Nonlinear neurons are established by defining nonlinear transfer functions between these interconnections. This nonlinearity is distributed throughout the network, and is very important in the present application due to the fact that the physical mechanism (generation of combustion products) responsible for the generation of the input signals (control and monitoring parameters), and output signals (emission levels) is inherently nonlinear [14].

B. Input-Output Mapping: Adjustment of the weights and biases during the neural network training process is performed by supplying a particular set of input samples to the network, so that they produce specific targets or desired outputs [33]. In the present study, the performance goal was achieved by the minimization of the mean square error (MSE) between the desired outputs (emission levels directly recorded from a SISIC engine) and the actual network response. Figure 3.5 depicts the input-output mapping property.

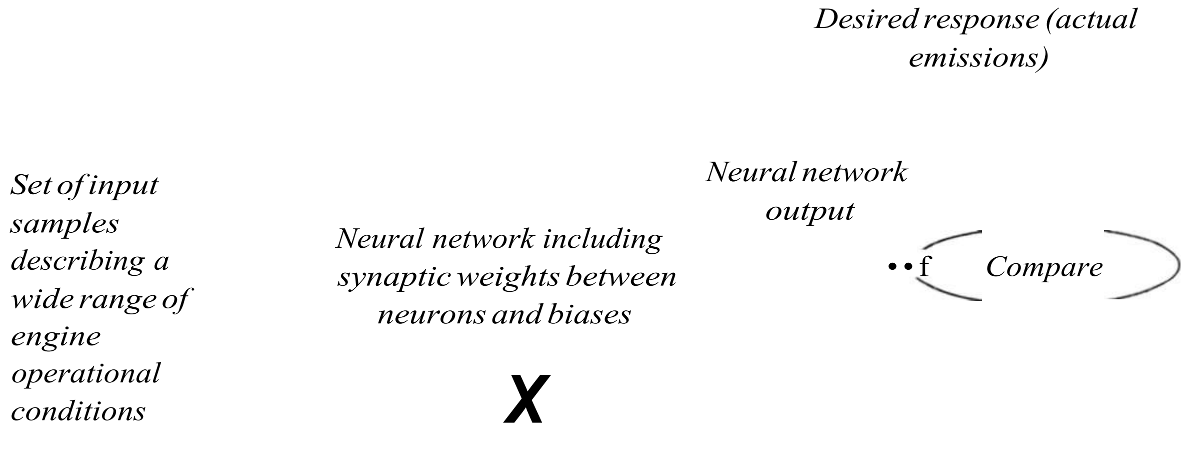

Desired response (actual emissions)
Set of input samples describing a wide range of engine operational conditions
Neural network including synaptic weights between neurons and biases
X
Neural network output
• • f
Compare

Figure 3.5. Block diagram illustrating the input-output mapping property of neural networks for this application.

Input-output mapping makes it possible to represent a SISIC engine performance by training a customized neural network system with certain number of training samples that correspond to specific engine operational conditions.

C. Generalization: The potential capabilities of neural networks are expanded by their ability for generalizing. This feature stems from the fact that neural networks can produce reasonable outputs for input samples not encountered in the set of training samples [31-34]. Therefore, when mapping an engine, not all-possible operational conditions must be set to ensure maximal training effectiveness.

D. Adaptivity: A neural network trained to operate and represent a specific process can be easily retrained to adapt its weights and biases to minor changes in the operating environmental conditions [31]. Therefore, engine performance variations due to lifetime degradation can be taken into account by readjusting the different weights and biases of the neural network neurons (the neural network can be retrained).

3.2.4. Learning process

The most significance property of neural networks is their ability to learn from the environment, and to improve their performance through learning. A neural network learns about a specific environment through an interactive process of adjustments applied to its synaptic weights and biases, and ideally it becomes more knowledgeable about its environment after each iteration of the learning process. In the present context, learning can be defined as the process by which free parameters (weights and biases) are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which parameter changes take place [32-33].

In other words, the neural network is stimulated by the environment. It undergoes changes in its free parameters as a result of the stimulation, and finally it responds in a new way to the environment because of the changes that have occurred in its internal structure.

There are different prescribed sets of well-defined rules for solving the neural networks learning problem, known as learning algorithms [33]. They differ from each other in the way the adjustment to a synaptic weight of a neuron is formulated, and the manner in which its structure relates to its environment.

For the present work, error-correction, supervised learning is used. These concepts are discussed below.

- Error-Correction learning resembles the input-output mapping property of neural networks. This learning process aims at minimizing a cost function $E_k(n)$ which is defined in terms of the error signal e^$n)$ obtained when comparing the output signal of the neural network $y_k(n)$ to the desired response or target output $d_k(n)$**.** Equations 3.1, 3.2, 3.3, and 3.4 describe the general error-correction learning process.

$$e_k(n) = d_k\{n) - yiin) \tag{3.1}$$

$$E_k(n) = 0.5\ e^2_k(n) \tag{3.2}$$

$$\mathrm{Awyty(n)}{=}\mathrm{r|}\ e_k(n)Xj(n) \tag{3.3}$$

$$vj_{kj}(n+1\ ) = \mathrm{w*/n)} + \mathrm{Aw*,(n)} \tag{3.4}$$

where k identifies an specific neuron, r| is a positive constant that determines the rate of learning as the learning process proceeds from one step to another, x,(n) is a component of the input signal vector $x(n)$, $w_{kj}(n)$ is the value of an specific synaptic weight, Awj/n) is a synaptic weight adjustment, and w^(n+l) is the updated value of synaptic weight W/t/n).

Similar error correction process is utilized when adjusting synaptic weights of a neural network when applying learning algorithms such as the Levenberg Maquardt and the Bayesian Regularization, which are described later. (See Appendix A).

- Supervised learning: Given an algorithm designed to minimize the cost function ***Ek(n),*** an adequate set of input-output samples, and enough time permitted to do the neural network training, a supervised learning system is usually able to perform tasks such as function approximation [31]. It is the "teacher" involved in this process that provides the neural network with a desired response for a training vector. Indeed, the desired response represents the optimum action to be performed by the neural network. The network parameters are adjusted under the combined influence of the training vector and the error signal, and a step-by-step process takes place making the neural network to emulate the teacher. Thus, knowledge of the environment available to the teacher is transferred to the neural network through training as fully as possible. When this condition is reached, the teacher is dispensed and the neural network deals with the environment by itself.

As a performance measure for the system, mean-square error or the sum of squared errors over the training sample is used. This sum may be visualized as a multidimensional error-performance surface or error surface. For the system to improve performance over time and therefore learn from the teacher, the operating point has to move down successively toward a minimum point of the error surface. Thus, a supervised system is able to do this with the useful information it has about the gradient of the error surface corresponding to the current behavior of the system [31,33].

A. **Learning algorithms**

Different ANN topologies and training algorithms have been introduced [31-34] for various applications. The Backpropagation (BP) algorithm is one of the most common and reliable algorithms to train neural networks and is described in numerous articles [35,36]. It is a gradient descent algorithm, and its name refers to the manner in which the gradient is computed for non-linear feed-forward multilayer networks. There are two different ways for implementing this gradient descent algorithm: incremental mode and batch mode [31,33]. In the incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In the batch mode all of the inputs are applied to the network before the weights are updated. The goal of the algorithm is to minimize the mean square error (MSE) between the predicted and the desired outputs of the network by changing the synaptic weights and biases caused by back propagating errors. High performance BP training algorithms fall into two main categories. The first category, which encompasses the variable learning rate backpropagation and the resilient backpropagation algorithms [33], uses heuristic techniques, and they involve the momentum technique which is often too slow for practical problems [33]. The second category corresponds to standard numerical optimization techniques associated with the Quasi-Newton and the Levenberg-Maquardt conjugate gradient techniques [33,37,38].

Very often, it is very difficult to determine which training algorithm would be the fastest for a given problem. The algorithm speed depends on many factors, including the complexity of the problem, the number of data points in the training set, the number of weights and biases in the network, and the error goal. In general, for most situations it is recommended to try the Levenberg-Maquardt (LM) algorithm first when working with networks that contain a few hundred weights [33]. However, if this algorithm requires too much memory, then the Quasi-Newton Broyden, Fletcher, Goldfarb, and Shanno (BFGS) algorithm is the option to try [33,34]. Table 3.1 compares the performance of different learning algorithms based on convergence time, number of epochs, and required computational operations [33]. A neural network with three layers containing 1-10-1 neurons or weights, is trained on a data set with 41 input /output pairs until a MSE of 0.01 is achieved. The networks were trained by using the Neural Network Toolbox provided by the CAD system, Matlab 5.3.0.

Technique	Time of convergence (sec)	Epochs	**Mflops
Variable learning rate	57.71	980	2.5
Resilient BP	12.95	185	0.56
Fletcher-Powell	16.4	81	0.99
Polak-Riebiere	19.16	89	0.75
Powell-Beale	15.03	74	0.59
One Step-Secant	18.46	101	0.75
BFGS Quasi-Newton	10.86	44	0.59
Levenberg-Maquardt (LM)	1.87	6	0.46

Table 3.1. Comparison between learning algorithms

*Iterations. This term is used when working with neural networks

* Million floating-point operations

The reader might notice that there is not a clear relationship between the number of floating-point operations (Mflops) and the time required to reach convergence. This is because some of the algorithms can take advantage of efficient built-in MATLAB functions, especially when using the LM algorithm.

Comparison between the results presented in Table 3.1 confirms the fact that LM, and Quasi-Newton BFGS training algorithms are the preliminary optimal algorithms to use, and therefore are applied in the present study.

Detailed explanation of the development and implementation of a neural network design for estimating gas emissions for SISIC engines is provided in the following sections.

3.3. Neural Network-Based Design Modeling for Estimation of Gas Emissions for SISIC engines.

3.3.1. Introduction

An emission monitoring system based on ANN can be represented by a group of neural networks capable of modeling the operation of actual gas (NO_x, HC, CO and CO2) sensors. These neural network-based sensors must be able to virtually monitor different emission levels generated from heavy-duty, natural gas SISIC engines used in the oilfield at any operational condition.

Choosing an appropriate structure for the different neural networks encompassing the system was one of the most difficult tasks due to the fact that there are no standards for application-related neural network structuring [34].

Extensive training and validation were performed to show the capabilities and the possible limitations of using neural networks for this particular application. Two algorithms (LM and Quasi-Newton BFGS) were initially used to train each network, and comparisons between both techniques based on time of convergence and generalization were made.

3.3.2. Parameter definition

In previous chapters definitions of certain operational parameters directly related to the emission generation were introduced. Two categories were established, control and monitoring parameters.

Selected control parameters were: (1) Lambda factor*; (2) Speed; (3) Load; and (4) Spark timing. Selected monitoring parameters were: (1) Exhaust temperature, and (2) Fuel pressure or fuel flow (optional). For the present approach, the system was designed to estimate concentrations of NO_x, HC, CO, and CO2 after combustion.

3.3.3. Methodology

A. Neural network system architecture

The architecture of the neural network system is related to the number of neural networks to be designed and implemented. After defining the number of neural networks, determination of the number of inputs, the number of layers, the number of outputs, the type of transfer functions applied to each layer, and the neuron interconnection layout for each neural network was pursued. As mentioned before, presently there are no standard guidelines to define the architecture of a neural network. Therefore, proper structuring of each network to assure maximum training effectiveness was performed.

* Factor calculated by the existing engine management system. Factor $X = X$ (manifold pressure,RPM,fuel flow, atmosphere pressure)

The neural network system consisted of a number of neural networks working in parallel, each one designed to estimate a specific output. Due to the high nonlinearity of the combustion process [13,14], the decision to use a customized neural network for each output or emission level estimated was taken. Figure 3.6 presents a block diagram of the developed system involving estimation of three emission levels (NO_x, CO and CO2).

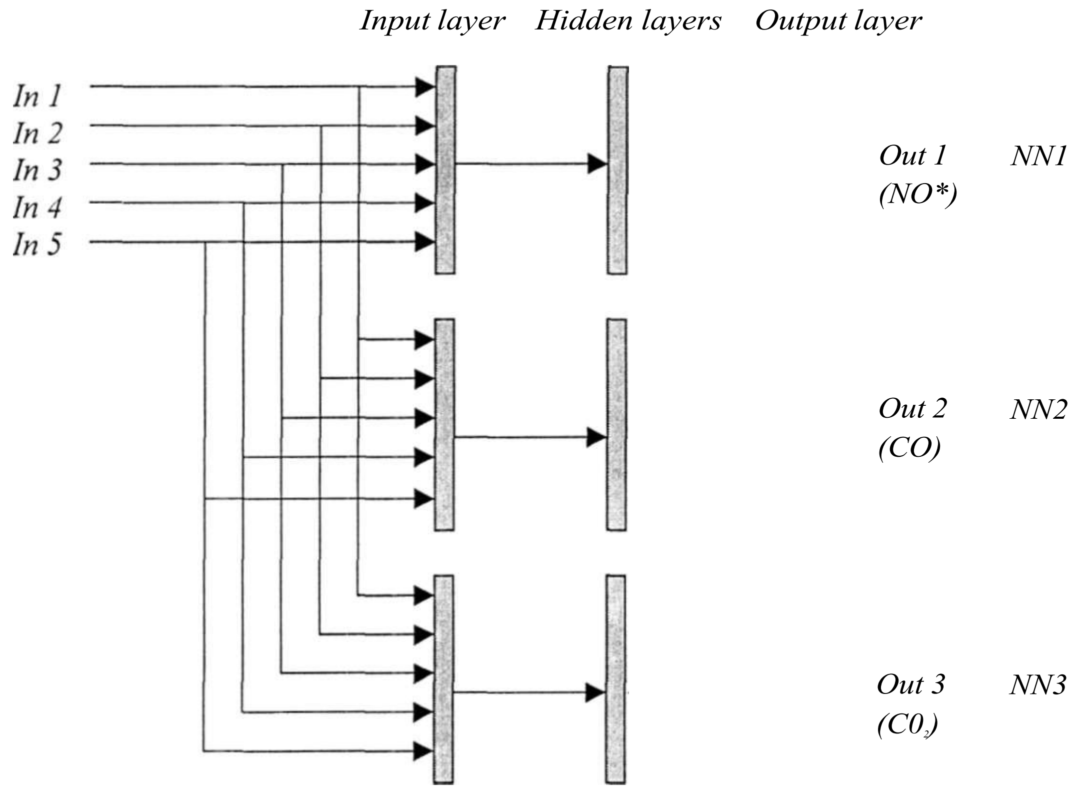


Figure 3.6. Block diagram illustrating the external architecture of the proposed neural network system. Three outputs were estimated based on the variation of five input parameters. Three neural networks (NN1 to NN3) operated in parallel.

B. **Neural network system training**

Two high performance algorithms, which can converge from ten to one hundred times faster than the traditional *Backpropagation* algorithms, were used to train the present system. They were (1) *Quasi-Newton BFGS,* and (2) *Levenberg-Maquardt* fZMj[34,37,38]*

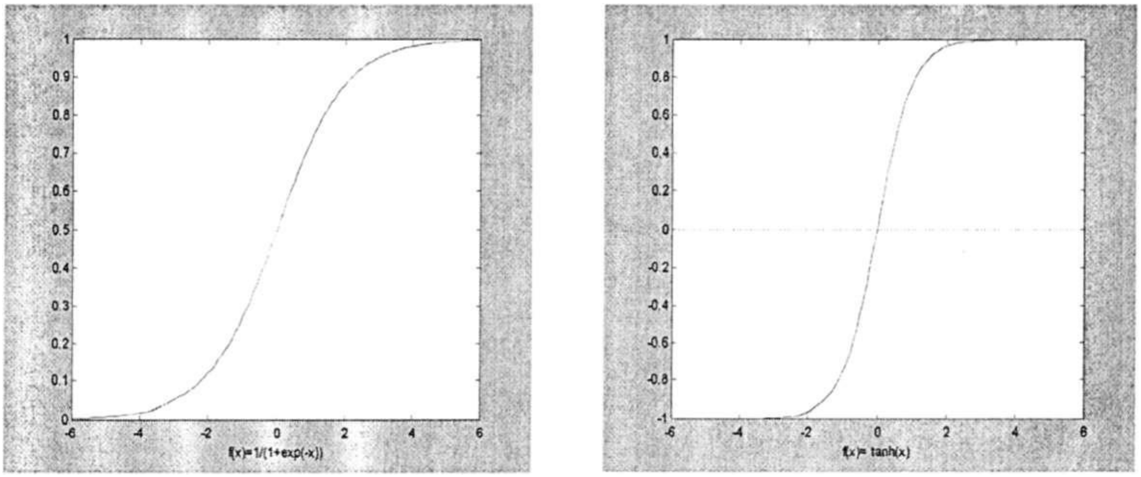
When training the neural networks, initial preprocessing steps on the network inputs and targets were carried out. First, all network inputs and targets were normalized within a specified range, enhancing training effectiveness. Second, due to the fact that in some situations the dimension of the input vector was large and its components or samples were highly correlated, it was useful to reduce its dimension using the so-called principal component analysis [33]. This technique orthogonalized the components of the input vector, so that they were uncorrelated with each other, ordered the resulting orthogonal (principal) components so that those with the largest variation came first, and finally eliminated the components that contributed the least to the variation of the data set.

As mentioned before, nonlinear functions give neural networks distinct nonlinear capabilities. One of the most common forms of activation function is the sigmoid which is a monotonically increasing function that asymptotes at some finite value as ± co is approached. The most common examples are the standard logistic function$_J$/(x)=(l+e"^{x})"^{1} and the hyperbolic tangent _/(x)=tanh(x) shown in Figure 3.7. Sigmoids that are symmetric about the origin are preferred so that they produce outputs (inputs to the next layer) that are on average close to zero due to normalization. Therefore, for the present

* See Appendix A for BFGS and LM description and implementation

approach hyperbolic tangents were applied to all layers in the networks excepting to the output layer, because it only contained one neuron.

Initial weights were chosen randomly but in such way that the sigmoid was primarily activated in its linear region. Thus, weights that ranged over the sigmoid linear region forced the network to learn the linear part of the mapping before the more difficult nonlinear part during training.



(a) *(b)*

Figure 3.7. (a) Logisticfunctionf(x)=(l+e~^{x}f ; (b) Hyperbolic tangentfunction f(x)=tanh(x)

On the other hand, *overfitting* is one of the problems that can occur during neural network training [34]. A neural network can memorize the training samples, but does not learn to generalize new situations, if it is not properly trained. Among the methods for improving generalization, the most common techniques encountered in the literature were Regularization, and Early Stopping [31,33,34].

Regularization involves the modification of the performance function, which is normally chosen to be the sum of squares of the network errors on the training set, i.e., the MSE represented by Equation 3.5. This modification *(msereg)* is performed by adding a term that corresponds to the mean of the sum of squares of the network weights *(msw),* as described in Equation 3.7:

$$\begin{array}{l} \qquad I\ \ N \qquad\qquad J\ \ N \qquad\qquad \wedge \\ mse = \text{—}\,\mathrm{V}\ (eS' = \text{—}\,\mathrm{V}\ (t,\ \text{-}\ a,)^{2} \end{array} \qquad (3.5)$$

$$msereg\ \text{-}\ /.mse + (1\ \text{-}\,y)ms\backslash v, \qquad (3.6)$$

where y is the performance ratio, n the number of weights, and

$$\begin{array}{l} \qquad\quad 1\ \ \wedge \\ msw \end{array}$$

Using this new performance function (Eq.3.7), the network can be forced to have smaller weights and biases. Therefore, a smoother network response can be observed. However, the problem with regularization boils down to the difficulty of determining the optimum value for the performance ratio parameter. Choosing a very large performance ratio caused overfitting. Conversely, a small value did not allow proper training. Therefore, decision to use automated Bayeisan regularization (see Appendix A), was made based on the capabilities provided by the MATLAB Neural Network Toolbox [33]. One feature of this regularization technique was that it provided a measure of how many network parameters (weights and biases) were effectively used by the network. Therefore, determination of these active parameters helped to decide whether larger or smaller neural network architecture was required to reach the training goal. Thus, utilization of the Early Stopping technique was not needed due to the remarkable results obtained using the automated regularization technique (see Section 3.3.5).

The inputs and the targets were supplied in batches to the neural networks during training. The use of batch learning was the only choice since the training algorithms (Quasi-Newton BFGS, LM and Bayesian Regularization) only operate in this mode [33,38].

Each neural network was trained independently using the described methodology.

C. Neural network system validation

Validation was performed in order to evaluate how well each neural network generalized [32]. A set of input samples not encountered in the set of training samples was entered into each neural network. The network outputs were compared to the targets or actual outputs. Two analyses were performed to assess the performance of each trained network. First, the maximum relative error (MRE: % difference between the network outputs and the actual targets) was calculated. Then, linear regression analysis between the network outputs and the corresponding targets was performed. Correlation coefficient (R-value) was calculated to quantify the similarity between the corresponding targets and outputs [33].

3.3.4. Implementation

The neural network system was designed to model different emission levels generated by natural gas SISIC engines as a result of the variation of some crucial control parameters. Gas emissions selected* to show feasibility of the present study were NO_x, CO, HC, and Co2, which generally present the greatest environmental concern. Therefore, four neural networks were designed, trained and validated. The following steps were performed to design and implement these neural networks.

A. **Defining the input and output vectors, learning process and training algorithm**

Initially, the input and output vectors used to train each neural network were defined. The input vectors corresponded to the key operational parameters related to the emission generation. Five operational parameters were selected as elements of each input vector. As a result, four input vectors related to the NO_x, HC, CO, and CO2 estimation containing six elements (air/fuel ratio, speed, load, spark timing, average right exhaust temperature, and average left exhaust temperature) were established for each neural network. The number of input samples for training each network was determined based on a wide range of engine operational conditions. Subsequently, manual variation of individual control parameters (air/fuel ratio, speed, load and spark timing) was performed, recording the impact of these variations on the remaining monitoring parameters (in this particular case only the average exhaust temperatures, which are used as estimates of the combustion temperatures for each engine cylinder).

The output vectors corresponded to the gas emissions (NO_x, CO, HC and Co_2) generated as a result of the different operational conditions obtained by manual variation

' Selection was limited to test the neural network approach. Other gases might be included.

of each control parameter. Four output vectors were defined, each containing the dynamics of a single element representing a specific gas emission. The number of emission samples (outputs) was made equal to the number of individual variations of each control parameter needed to cover the predefined engine operational range. These output samples were recorded using a sophisticated gas analyzer.

Based on similar previous applications [6,9,39], "Backpropagation" was selected for implementing the learning process [31]. However, as mentioned before, the application of traditional backpropagation training techniques was avoided, and two high performance algorithms, the BFGS and the LM techniques mentioned earlier, were adopted. Additionally, Bayesian Regularization was applied during training and results are presented in Section 3.3.5.

A.1. Engine data collection

Engine operational and emission data were collected at the North Caroline Gas plant in September 2000. The machine used was a 16 cylinder SISIC engine made by Waukesha, Model P9390GSIU, Rated Power 1800HP located at a BP Energy gas plant. A sophisticated gas analyzer (GA-40T plus, MADUR, Vienna, Austria) was utilized to record the different emission levels emanating from this unit. Data is presented in Table 3.2.

Sample	*X*	Speed	Load	SparkT	RExhT	LexhT	NO_x	CO	co_2
		RPM	HP	°bTDC	°C	°C	ppm	ppm	
1	1.23	1000	1441	24	521.75	579.25	5366	940	9.75
2	1.28	1000	1456	24	500.75	575.37	4552	930	9.22
3	1.33	1000	1536	24	500	570.87	3342	710	8.63
4	1.28	1000	1546	24	507.87	577.87	4507	820	9.07
5	1.23	1000	1559	24	520.37	588	5485	950	9.56
6	1.28	1000	1575	24	508.75	575.12	3599	810	8.69
7	1.4	1000	1593	24	505.25	573.25	3359	770	8.65
8	1.35	1000	1595	24	504.5	576.12	4057	790	8.84
9	1.35	1025	1638	24	515.12	582.12	4240	770	8.93
10	1.33	1025	1643	24	514.25	579.87	3604	790	8.76
11	1.28	1025	1650	24	513.12	581.12	3960	800	8.97
12	1.23	1025	1647	24	526.75	591.12	4763	950	9.99
13	1.4	1000	1509	24	490.37	559.5	2051	530	8.35
14	1.45	1000	1506	24	496.5	556.62	1920	510	8.17
15	1.5	1000	1504	24	494.12	557.37	1568	520	8.15
16	1.5	1050	1565	24	496.87	565.62	1573	550	8.15
17	1.45	1050	1566	24	500.25	563.62	1881	550	8.19
18	1.4	1050	1562	24	502	562.75	2053	550	8.39
19	1.4	950	1421	24	481.75	535.62	2248	560	8.1
20	1.35	1000	1577	24	502.75	569.5	2947	590	8.65
21	1.3	1000	1568	24	504.87	573.37	4025	650	9.14
22	1.3	1000	1594	27	499.87	564.62	4416	650	8.93
23	1.3	1000	1596	21.9	520	577.12	3072	640	8.78
24	1.3	1000	1585	19.9	519.12	582.37	2953	590	9.01
25	1.3	1000	1591	18.8	524.12	587.62	2792	570	9.05
26	1.27	1000	1589	24	524.37	575.87	4383	640	9.24
27	1.23	1000	1586	24	515.75	582.5	4991	690	9.84
28	1.18	1000	1573	24	521.75	590.12	5433	680	9.99
29	1.18	950	1467	24	514.75	573.25	5563	730	10.01
30	1.23	950	1472	24	501.37	565.12	5320	710	9.48
31	1.27	950	1476	24	494.25	559.25	4706	670	9.1
32	1.3	950	1478	24	490.37	555	3828	610	9
JJ	1.35	950	1476	24	484.37	547.37	2615	550	8.57
34	1.4	950	1462	24	479.25	546	2498	550	8.53
35	1.4	90D	1370	24	471.62	532.75	2633	550	8.51

Table 3.2. Engine operational and emission data

A detailed section describing the overall experimental procedure for engine data collection is provided (See Appendix B). In addition, uncertainty of experiments is discussed.

B. **Neural network system architecture**

In order to obtain an optimal neural network structure, an algorithm was designed to determine the number of layers, the number of neurons in each layer, and the neural network layout, considering two main goals: (1) minimal MSE between the neural network outputs and the actual targets after training; and (2) good generalization.

Initially, the algorithm created four random-sized neural networks, which were relatively small multilayer networks. These initial networks were too small for accomplishing the given task. Therefore, a new neuron, or a new layer of hidden neurons, was added only when the training process was unable to meet the design specification. This technique is known as the Network Growing technique [31].

Before training the networks, the inputs and the targets had to be normalized so that they were always within a specified range. In this particular case, both inputs and targets fell in the range [-1,1]. Additionally, principal component analysis [33] was applied to avoid high correlation between the components of the input vector. Subsequently, training was performed by supplying a set of training samples into the networks in batch mode, i.e., adjustment of weights and biases started after all inputs and targets had been already entered into the networks.

Adjustment of weights and biases during training was performed and partially stopped when a maximum mean square error (MSE) between the network outputs and the target of 2.5x10"^{5} was obtained. After achieving this initial performance goal, each neural network was validated using a set of different input samples not encountered in the training set to assess the generalization ability of the network. Validation was evaluated by calculating the maximal percent difference between the network outputs and the targets. The entire training process was stopped when a Maximum Relative Error (MRE) of 5% was obtained. Otherwise, the training process continued iteratively.

3.3.5. Results

After several trials, four medium size networks were obtained. Architecture of one of these networks is presented in detail in Figures 3.8.a, 3.8.b, 3.8.c and 3.8.d, and a structure summary is given in Table 3.2. In addition, the adequate transfer functions to incorporate the necessary non-linearity between all layers (with the exception of the output layer) were determined, as mentioned in Section 3.3.3 (B)

Layer	#Neurons	Transfer function
Input	22-30	Nonlinear: Hyperbolic tangent
Hidden	8-12	Nonlinear: Hyperbolic tangent
Output	1	Linear: f(x)=mx+b

Table 3.2. Internal structure of the neural networks used for emission estimation

Four three-layer networks were obtained. Each network contained a different number of neurons in the input and hidden layer, ranging between 22-30 and 8-12 neurons respectively.

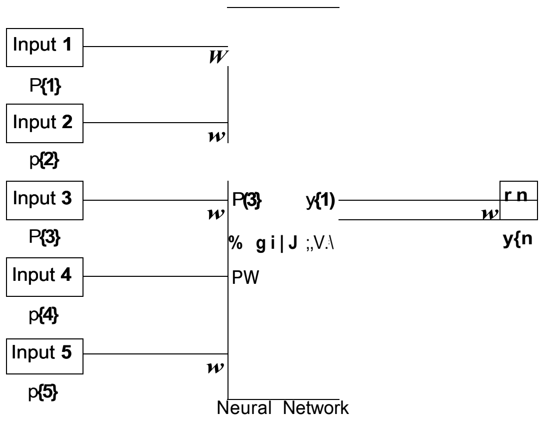


Figure 3.8.a. General neural network structure. Five inputs were supplied to a black box containing a number of layers and neurons determined by the developed algorithm.

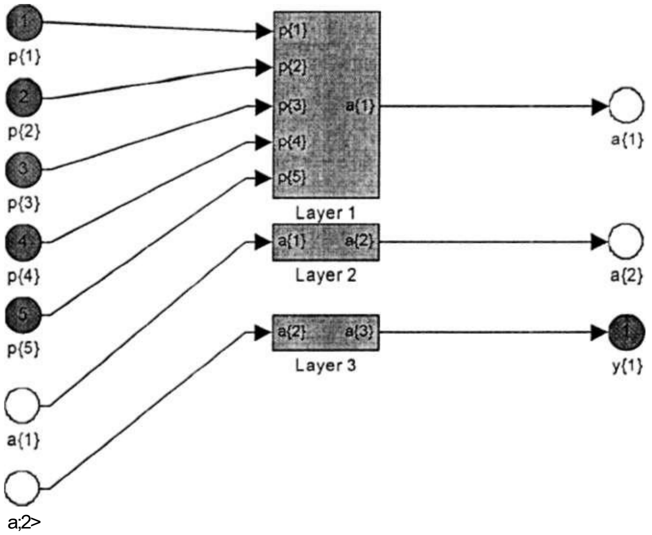


Figure 3.8.b. Detailed layout of connections between the layers encompassing one of the neural networks of the present system.

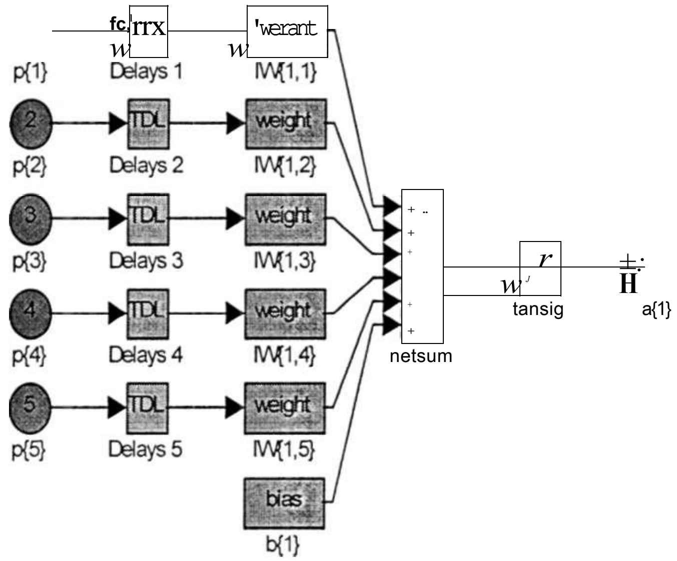


*Figure 3.8.**C.** Detailed layout ofweights and biases connectionsfor thefirst layer ofa neural network in thepresent system. In thisparticular example, the layer contained a 25x5 weight matrix, and a 1x5 bias. Note that the weight indices indicate the input-weight connections and not the dimension ofthe weights. Each input sample was delayed in order to synchronize the network operations.*

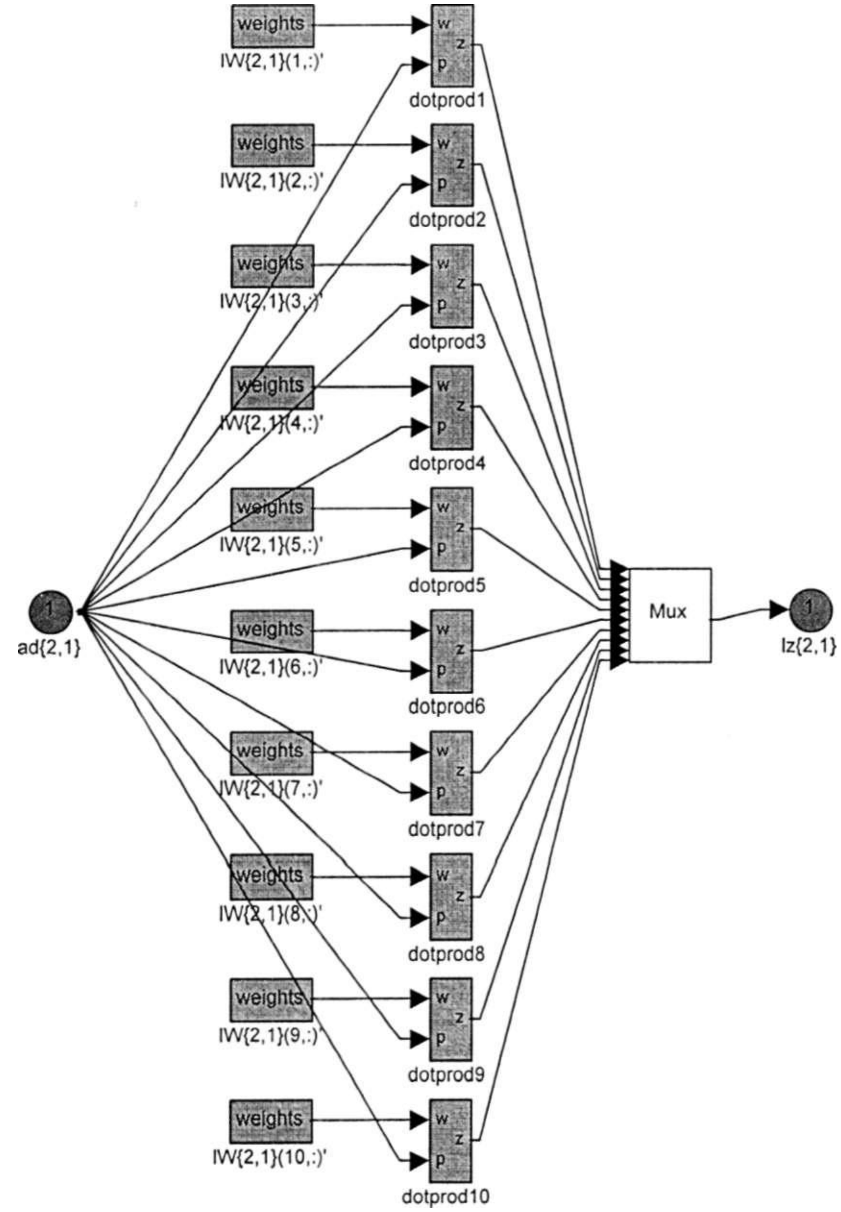


Figure 3.8.d. Detailed picture of the second layer of one of the neural networks in the system. Ten 1x25 weights were multiplied (dot product) with the results obtained from the first network layer (ad(2,l)).

Table 3.2 compares the results from the applied training algorithms (BFGS, LM and Bayesian regularization). The comparison was done based on the number of iterations required to successfully train the network and the time of convergence.

Algorithm	Number of iterations	Time of convergence
BFGS	70-100	5-10 minutes
LM	20-30	1/2-1 minute
Bayesian R	50-80	7-12 minutes

Table 3.2 Comparison between training algorithms based on the number of iterations and time of convergence to reach the training goals.

Figures 3.9.a, 3.9.b, and 3.9.c show the response of three different neural networks after training. Clearly, no differences are evident in these graphs due to the fact that the trained networks were capable of learning the training samples. One might think that any of these algorithms would efficiently work for this application. However, quantitative validation is the only method capable of proving this.

After the first validation phase (see Figures 3.10.a, 3.10.b. and 3.lO.c), these algorithms showed capability to generalize well. Small but consistent differences showed that the Bayesian Regularization technique generated the minimal relative error between the targets and the estimated outputs from the network. Furthermore, a second validation phase was performed to evaluate more precisely the capabilities of each neural network to generalize. Figures 3.1 La, 3.1 Lb., and 3.1 l.c reveal the unquestionable benefits of using the last but best training algorithm, the Bayesian regularization. However, the Quasi-Newton BFGS and LM algorithms also complied with the set goals (i.e. MRE

<5%). Therefore, the discriminating factors would be the speed of convergence and the efficiency.

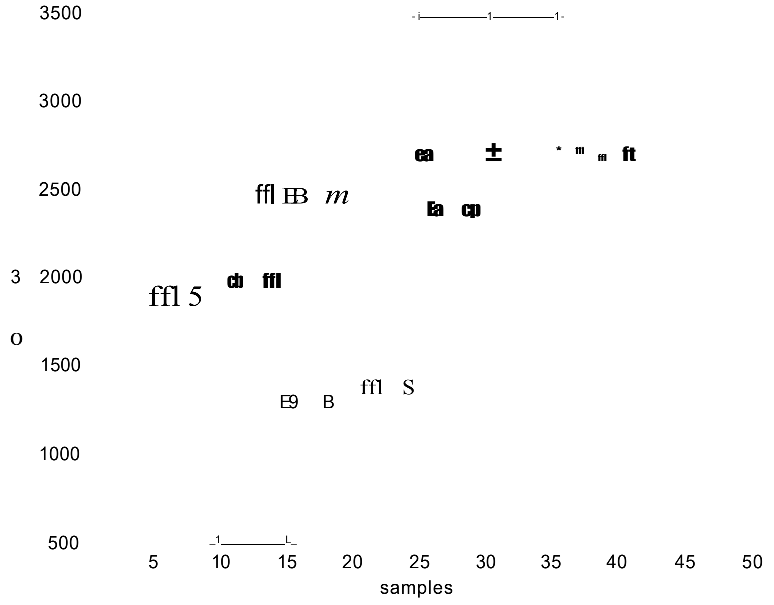


Figure 3.9. a. Neural network response after training using the Quasi-Newton BFGS algorithm. Square points represent desired outputs and cross points denote the estimated values by the neural network. Both outputs are normalized

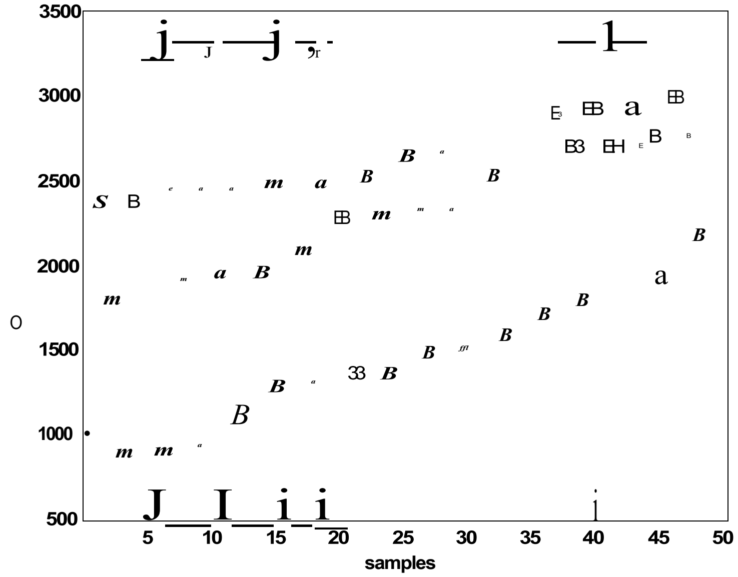


Figure 3.9.b. Neural network response after training using the LM algorithm. Square points represent desired outputs and cross points denote the estimated values by the neural network.

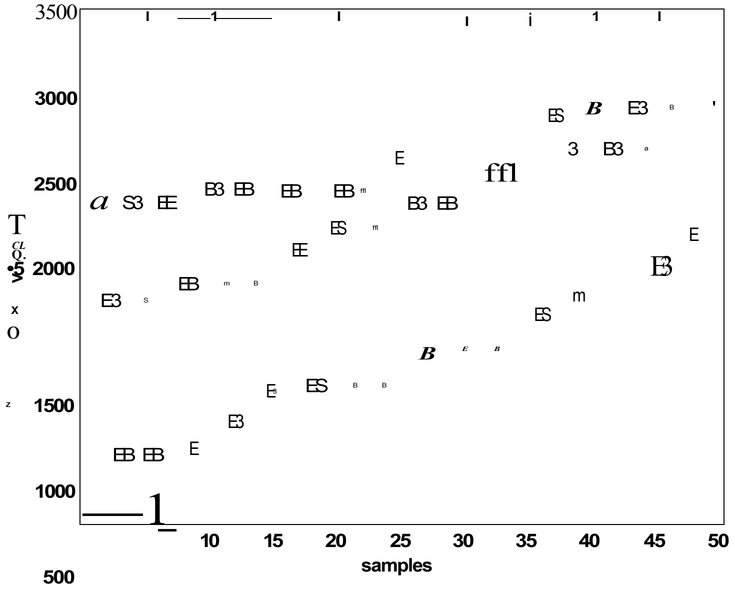


Figure3.9.c. Neural network response after training using the Bayesian regularization algorithm. Square points represent desired outputs and cross points denote the estimated values by the neural network. Both outputs are normalized

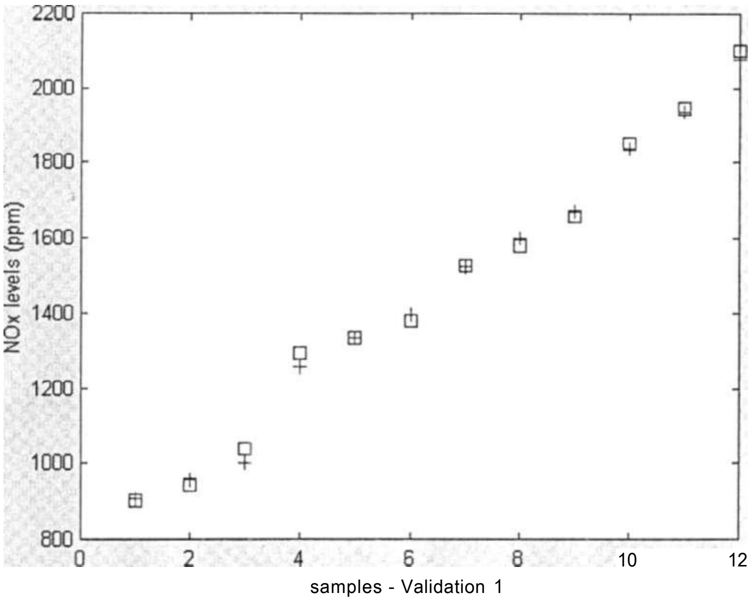


Figure 3.10.a. Initial validation of the Quasi-Newton BGFS training algorithm; MRE =4.6 %

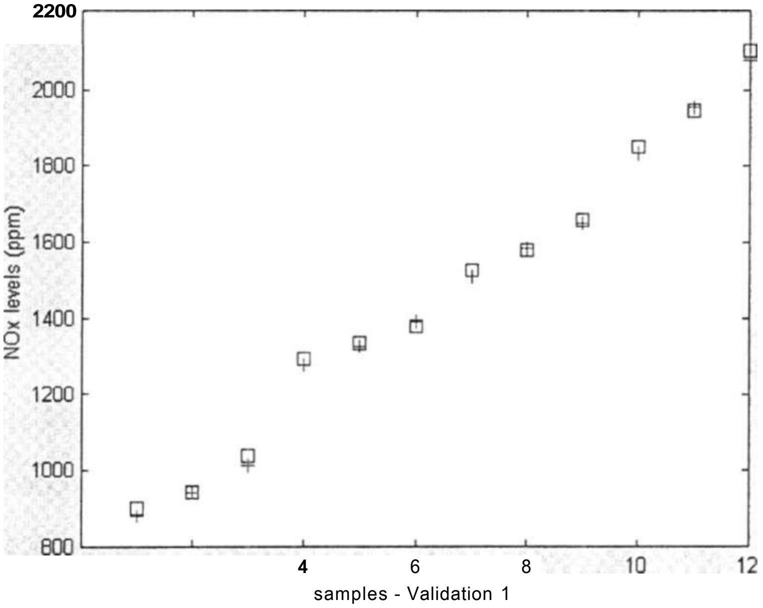


Figure 3.10.b Initial validation of the LM training algorithm; MRE -2.69%

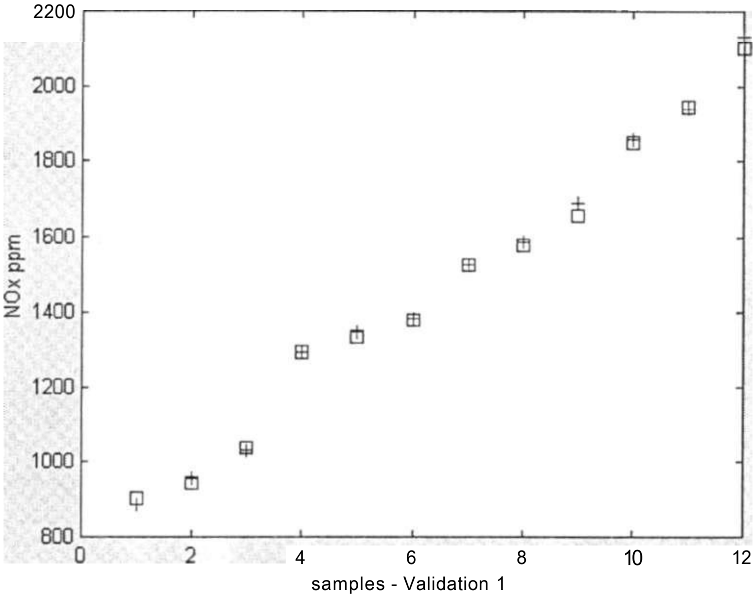


*Figure 3.10.***C.** *Validation J- Initial validation of the Bayesian regularization training algorithm; MRE =2.141%*

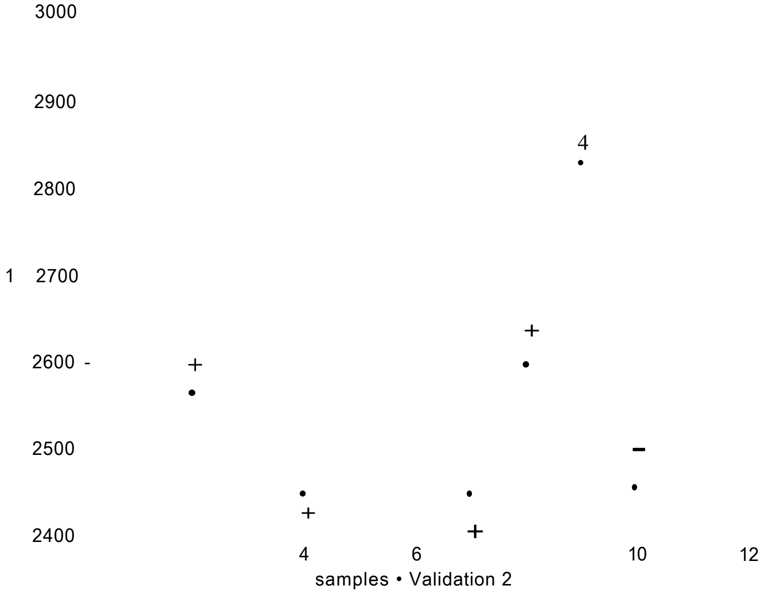


Figure 3.11.a. Additional validation of the Quasi-Newton BFGS training algorithm; MRE= 4.98%

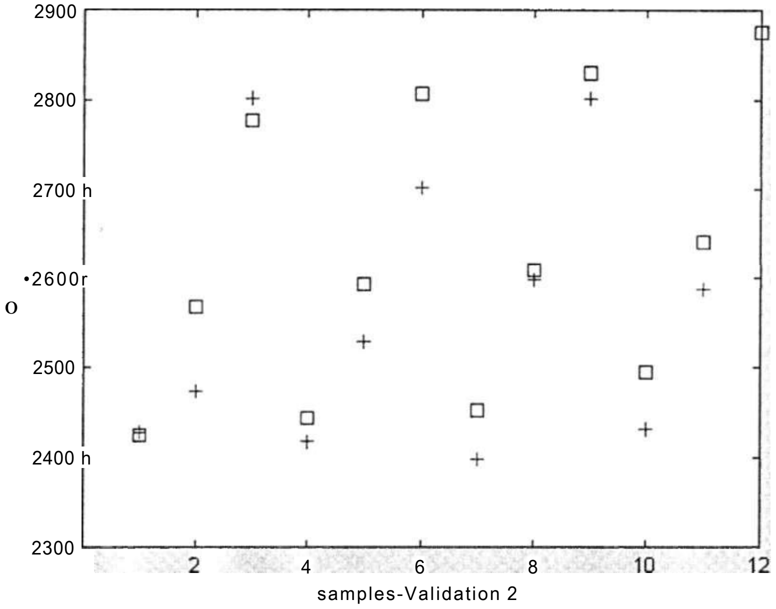


Figure 3.1 Lb. Additional validation of the LM training algorithm; MRE =3.75%

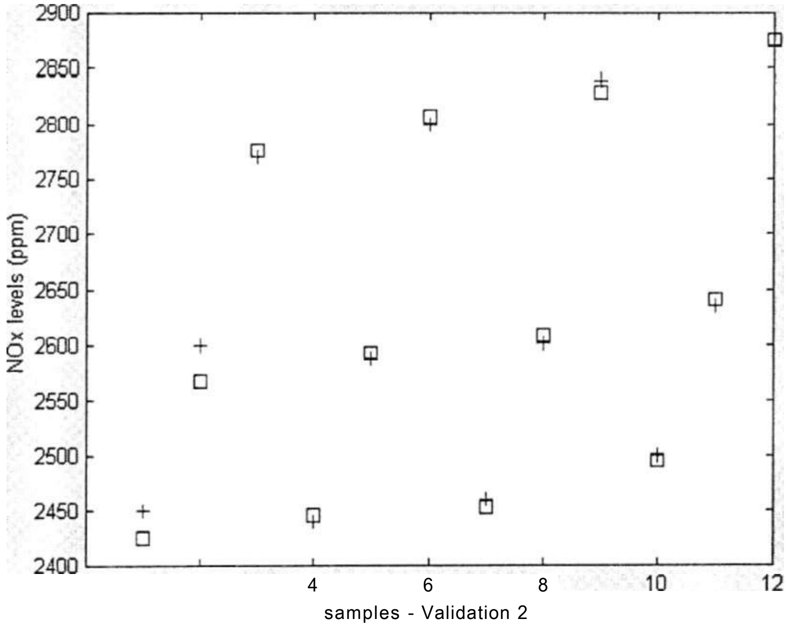


Figure 3.1 I.e. Additional validation of the Bayesian regularization-training algorithm;
MRE = 1.9%

Linear regression between the network outputs and the corresponding targets for the overall data was carried out for all training algorithms. High correlation between the outputs and the targets was obtained (R-value between 0.995 and 1). Figure 3.12 illustrates the post-training analysis for a neural network trained using the LM, the BFGS and the Bayesian regularization algorithms.

The neural network implementation was developed based on the Neural Network and Simulink toolboxes provided by the CAD system MATLAB 5.3.1 (The Mathworks Inc., Natick, MA). A graphic user interface (GUI) was developed to represent the system. The description of this GUI is given in Chapter 4, section 4.2.5.

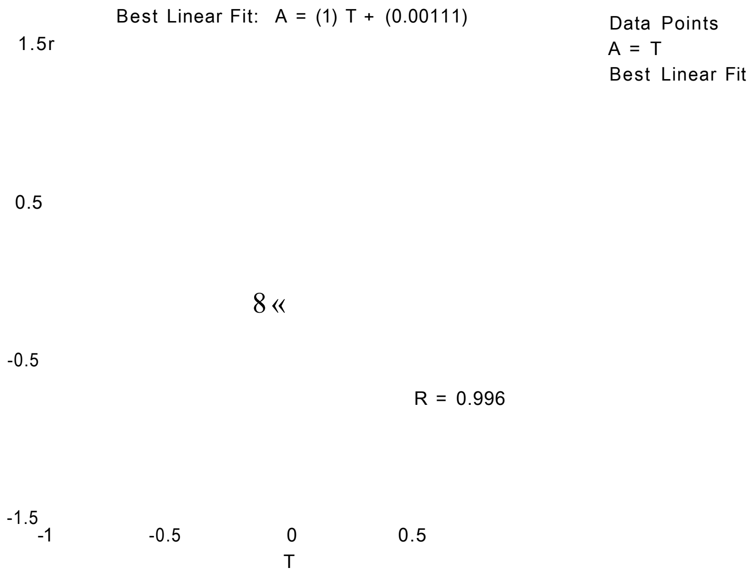


Figure 3.12.a. Linear regression analysis (R-value-0.996), using the BFGS algorithm to train the networks.

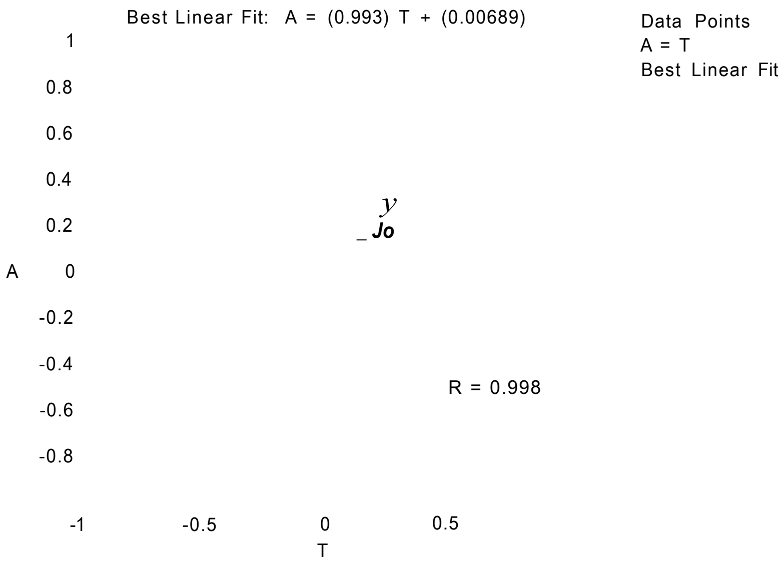


Figure 3.12.b. Linear regression analysis (R-value=0.998), using the LM algorithm to train the networks.

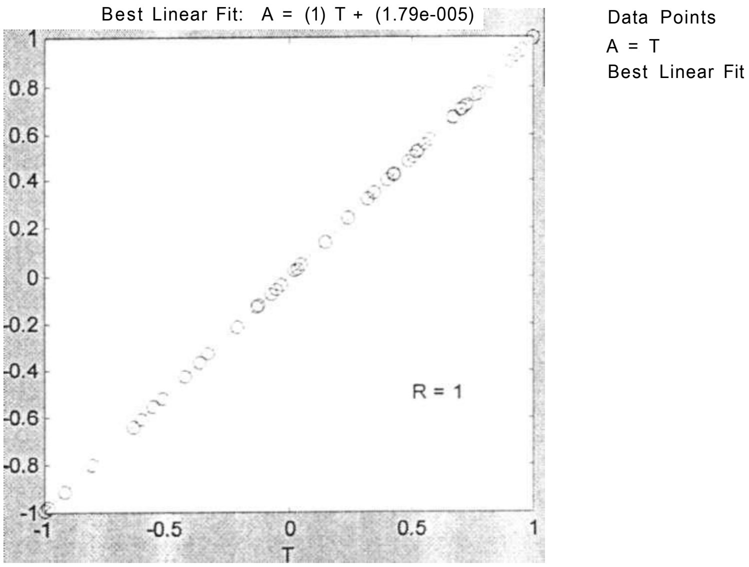


*Figure 3.12.***C.** *Linear regression analysis (R-value=l), using the Bayesian regularization algorithm when training the networks.*

3.4. Discussion

The benefits of using artificial neural networks for emission estimation for SISIC engines can be extended to any combustion unit such as boilers and stationary turbines where there exists a relationship between monitoring and control parameters with the generation of the same or different pollutants. Thus, a parametric emission monitoring system based on neural networks can replace the existing CEMS, without any problems that could jeopardize the standards of performance of CEMS for stationary combustion units.

CHAPTER IV: EMISSION CONTROL SYSTEM FOR SISIC ENGINES

4.1. Overview

Nowadays, more restrictions have been placed on the performance of different engine installations, therefore, innovative treatment technologies originated, and frequently new materials capable of reducing those emissions have been applied [3,40].

One of the most common techniques to reduce the different emissions emanating from SISIC engines is using a 3-way catalytic converter, described before (See Chapter I). The use of 3-way catalytic converters introduces high installation and maintenance costs which owners of engine facilities see as prohibitive to undertake.

Therefore, alternative techniques to control these emissions have been in the focus of different environmental organizations, industries and research institutes [2,11]. These techniques involve an appropriate manner of finding optimal operational conditions that might force the engines to generate the legally required emission levels or reduce them so that the related environmental effect is less, and permission to operate engine is obtained.

This chapter aims to present a novel and universal approach for emission control, subsequent to the design of a neural network system for emission estimation. This control technique is also based on neural networks so that a complete emission estimation and control system can be integrated.

The system can be envisioned as a set of inverse models of the emission estimation system previously developed (see Chapter III). These inverse models were based on a new set of neural networks, designed to represent an open loop system capable of estimating the proper tuning of the control parameters related to emission generation.

The main idea was to provide an effective tool that could facilitate a SISIC engine operator in finding the optimal values for certain control parameters and, by tuning them, to comply with the required emission levels. Therefore, using this technique, the system operator would be asked to manually enter a desired emission level, and the model would estimate the proper changes to the selected control parameters that might be implemented to obtain the desired engine behavior for generating the emission targets.

In general, the concept of applying NN for emission control could be applied in a closed loop fashion. However, there were few, but important reasons that limited thL implementation, e.g. possible unsafe engine operation, and limited owner's control over the engine operation.

4.2. Open-loop emission control concept

Based on the neural network-based parametric emission estimation model, manipulation of the selected control parameters led to the variation of the different emission levels. Therefore, control parameters that required proper tuning in an open-loop control setup were the lambda factor, the speed, the load and the spark timing.

In a control system, these parameters were the targets (or the outputs) of a new set of neural networks used in the system design. Initially, the problem of controlling the emissions was virtually solved by designing four new neural networks corresponding to each emission-related operational parameter. The inputs for these neural networks were the engine operational parameters used to estimate the emissions in the previous PEM model (see Chapter III), excluding the parameter to be tuned, and an additional input corresponding to the emission level required. The required emission level was entered manually by the user and immediate calculation and description of the optimal engine operation was generated by the corresponding neural network.

After obtaining the optimal control parameters, tuning could be achieved by manipulating these control parameters through the existing engine management system.

4.2.1. Description

Figure 4.1 presents a diagram of the initial proposed emission control system. The figure represents the open loop control of one of the parameters related to emission generation.

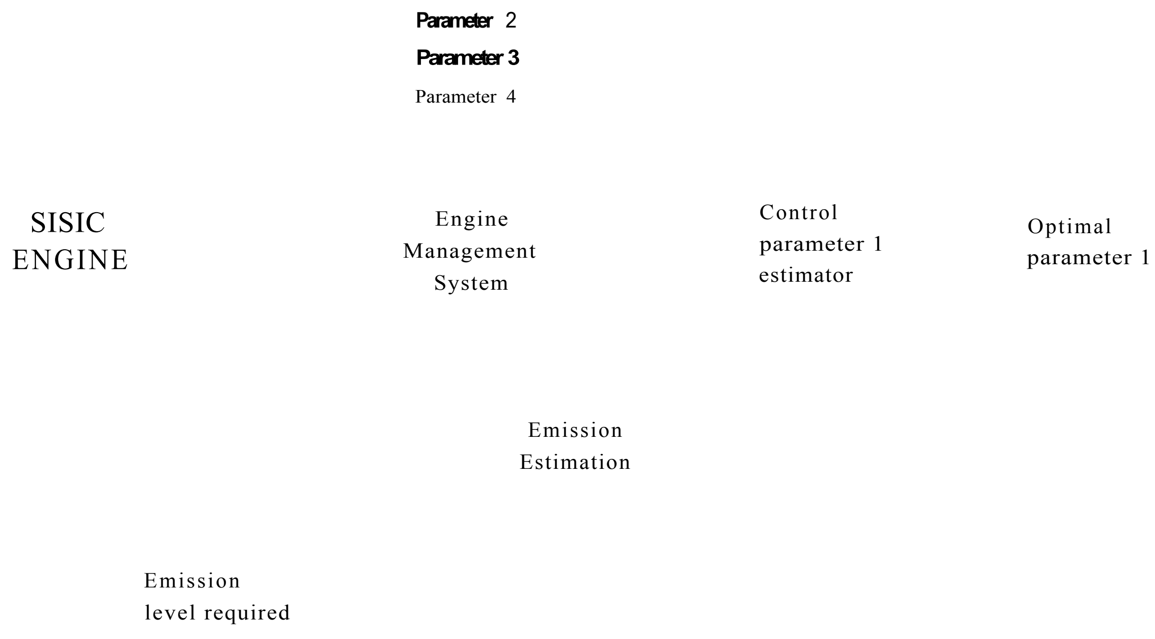


Figure 4.1. Open-loop concept for emission control

The selected control parameters were supplied to the already designed emission estimation system. The system estimated the selected emission level, which was compared to the emission level required. The resulting error was made part of the input vector supplied to the new module, the control parameter estimator. Control parameters were made part of the input vector, excluding the parameter to be tuned (Parameter 1 in the Figure 4.1). This module could estimate the optimal value of the selected control parameter needed to reach the emission level desired.

The overall control system encompasses 17 modules: 1 module for emission estimation (the neural network PEM system), and 16 additional modules, each one capable of estimating/tuning a selected control parameter (lambda factor, speed, load, and spark timing) for a selected emission (NO_x, HC, CO and $C0_2$).

This modular structure allows expansion of the system at any time, i.e., the system can be configured so that any new operational parameter or emission species can be included for subsequent emission estimation or control.

4.2.2. Open-loop emission control system design using artificial neural networks

An algorithm to create 16 neural networks was developed using the same methodology employed to build the parametric emission monitoring system described in Chapter 3. The process involved training and validation of 16 neural networks. In the following sections, details are provided to explain the different steps followed to develop the present control system.

Integration of the parametric emission monitoring system with the proposed open loop control system is explained, and discussion about the application and importance of the overall system is provided.

A. Defining input and output vectors, learning process and training algorithm of the control parameter estimator.

Eight input-output vectors were defined. They corresponded to the different lambda factor, speed, load and spark-timing ranges required to cover all possible engine operational conditions, and their effect on the variation of the NO_x, HC, CO and $C0_2$ emission levels. The vectors were defined as follows:

CONTROL 1: $X_t<X<K$

CONTROL 2 : Speed, < Speed < Speed,,

CONTROL 3 : Load, < Load< Load,,

CONTROL 4: Spark-timing, < Spark-timing < Spark-timing,,

EMISSION 1: NO_x, < NOx < NO_x,,

EMISSION 2: HQ < HC < HC,,

EMISSION 3: CO; < CO < CO,,

EMISSION 4: $C0_2$, < $C0_2$ < $C0_2$,,

Each module had different input and output vectors based on the operational parameter to control in order to reach a specific emission target. Table 4.1 describes the input and output vectors for each of the 16 modules or neural networks encompassing the control system.

MODULE (Neural network)	**INPUT VECTOR**	**OUTPUT VECTOR**	**EMISSION**
1	Control (2,3,4) Emission 1	Control 1	NO_x
2	Control (2,3,4) Emission 2	Control 1	HC
3	Control (2,3,4) Emission 3	Control 1	CO
4	Control (2,3,4) Emission 4	Control 1	$C0_2$
5	Control (1,3,4) Emission 1	Control 2	NO_x
6	Control (1,3,4) Emission 2	Control 2	HC
7	Control (1,3,4) Emission 3	Control 2	CO
8	Control (1,3,4) Emission 4	Control 2	$C0_2$
9	Control (1,2,4) Emission 1	Control 3	$N0_x$
10	Control (1,2,4) Emission 2	Control 3	HC
11	Control (1,2,4) Emission 3	Control 3	CO
12	Control (1,2,4) Emission 4	Control 3	$C0_2$
13	Control (1,2,3) Emission 1	Control 4	NO_x
14	Control (1,2,3) Emission 2	Control 4	HC
15	Control (1,2,3) Emission 3	Control 4	CO
16	Control (1,2,3) Emission 4	Control 4	$C0_2$

Table 4.1. Input and output vector structure for the 16 modules of the present control system

Similar to the neural network PEM system design process, the number of input samples for training each network was determined based on the range in which the engine is operated. Previous data recorded from the manual variation of individual control parameters (air/fuel ratio, speed, load and spark timing) and the corresponding emission levels were utilized to train and validate each network.

Based on the methodology followed to design and implement the neural network PEM system and the remarkable results obtained (see Section 3.3.5), a backpropagation algorithm was selected for implementing the Bayesian Regularization learning technique.

B. **Neural network system architecture**

An algorithm was developed to determine the number of layers, the number of neurons in each layer, and the neural network layout of each module encompassing the overall parameter control estimator. The performance of this algorithm was constrained by two main goals: (1) minimal MSE between the neural network outputs and the actual targets after training; and (2) good generalization.

Good generalization was the most important goal to achieve due to the fact that frequently it was difficult to collect engine data at all desired operational conditions. Commonly, the recorded data was restricted to a narrow operational range due to production schedules. For example, if the operator needed to restrict the engine operation between 80% and 100% of the allowed emission range, data was collected taking into account this restriction. However, if, at a later date, the owner or operator elected to operate at a different condition within this range or outside of the restricted range, the system must be able to provide acceptable new control parameter values capable of making the engine generate the desired emission levels.

The control module can be described using Figure 4.2. It consisted of two subsystems, Subsystems A and B. Subsystem A utilized the engine control parameters to estimate the expected exhaust temperature. The exhaust temperature, a crucial monitoring parameter utilized to represent the combustion temperature, was estimated to simulate the response of the engine to the variation of the control parameter to be tuned. Therefore, all control modules had associated with it an exhaust temperature estimation module, which was also built using neural networks.

In this particular example, Subsystem B used (1) the estimated exhaust temperature from Subsystem A; (2) the error generated from comparison between the estimated emission level provided by the neural network PEM system, and the level entered by the user, and (3) the control parameters except for the lambda factor which was to be tuned.

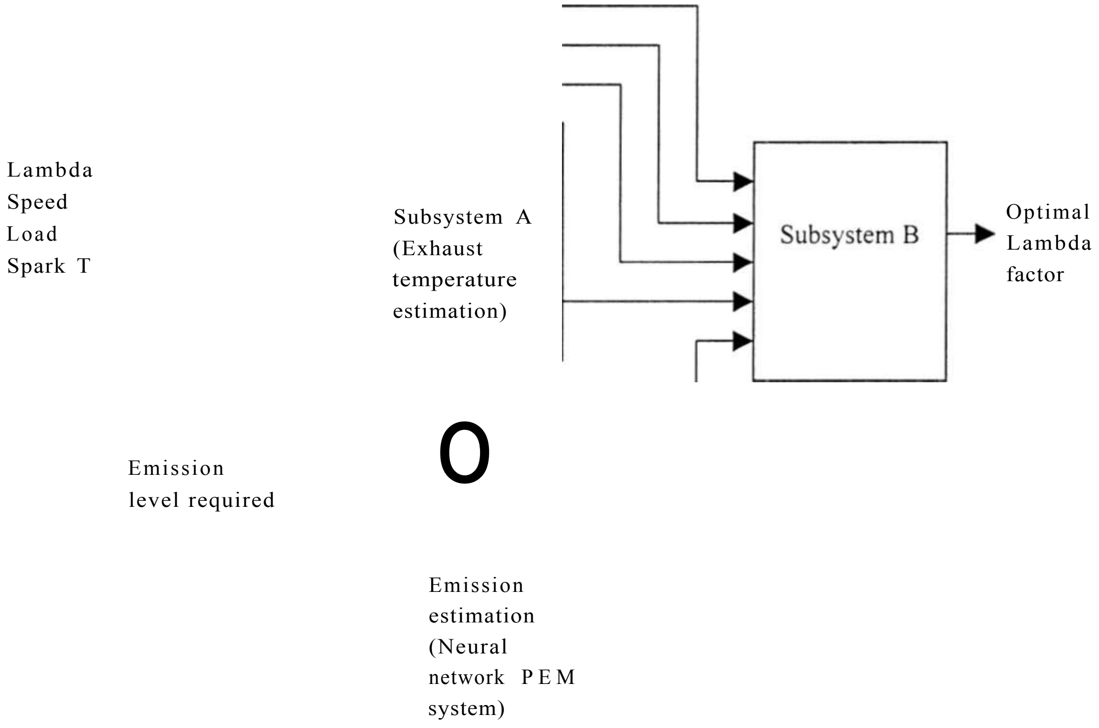


Figure 4.2. Detailed open-loop concept for emission control

The same architecture was applied for each control module designed for tuning the different control parameters (See Table 3.1). Each Subsystem B had associated a neural network properly trained, and validated.

C. Neural network system training

All neural networks were trained using the most effective backpropagation technique observed when developing the neural network PEM system, the Bayesian Regularization [33,38].

Before training the required neural networks, same steps to train the different neural networks of the PEM system were followed: (1) Normalization of inputs and targets, (2) Principal Component Analysis, (3) Selection of hyperbolic tangent as the type of transfer function to use, (4) Initialization of weights, and (5) Adjustment of weights and biases.

The inputs and the targets were supplied in batches to the neural networks during training, and each neural network was trained independently.

The developed algorithm used the Network Growing Technique [31] in order to generate the different networks that could comply with the performance goals of minimal MSE and good generalization.

D. Neural network system validation

The overall system was validated using the methodology explained in section 3.3.3 (C). The main difference in the present validation process was the quantity of modules (16 modules corresponding to 16 neural networks) to be validated, and the different input vectors and targets for each control module (see Table 3.1).

The goal of this process was to test how well the control system tuned a specific control parameter in order to achieve the emission level entered by the user.

4.2.3. Implementation

Open loop emission control modeling was pursued to demonstrate the functionality of this approach. After training each neural network, validation of each control network proceeded.

Validation process was divided in 2 sub-validation processes. The first validation (Valid1) involved the validation of the exhaust temperature estimation networks (Subsystem A), and the second corresponded (Valid2) to validation of the control parameter estimation networks (Subsystem B). Thus, each module was validated using different input samples not encountered during the training process. A maximum relative error of 5% was achieved, based on the stipulated generalization goals.

Integration of the overall neural network PEM system and the new control system was pursued (Figure 4.1). Testing the overall system was carried following four steps:

(1) An initial engine operational condition was set, based on the lambda factor, the speed, the load and the spark timing. This condition was entered into the neural network PEM system to obtain the corresponding emission levels.

(2) An emission target was entered regardless of the emission requirements.

(3) An engine control parameter module was activated so that estimation of only that control parameter according to the emission target was calculated. This step was repeated for all control parameters and emission species. However, if no control module was selected, the system was designed to tune the various engine control parameters with a logical sequence pre-established by the user. Additionally, control

parameter limits were set according to the real and the possible operational conditions supported by the existing engine management system.

(4) Comparison between the value of the estimated control parameter and the actual one was made, and the maximal relative error to verify the generalization level was calculated.

A graphical user interface (GUI) was developed to visualize the necessary tuning process to reach the emission target. The overall system development and integration was built using the Neural Network toolbox and graphical capabilities of the CAD system MATLAB 5.3.1.

4.2.4. Engine data collection

In order to train the overall system, engine operational conditions and emission data were recorded from a 16 cylinder SISIC engine made by Waukesha, Model P9390GSIU. The engine was located in BP Energy Canada, North Caroline gas plant (see Appendix B).

Table 3.2 shows the data collected in November 2000. HC emissions are not included due to the fact that 0 % concentrations were observed and recorded using the possible engine operational conditions that could be set during engine mapping.

Sample	X	Speed	Load	Spark T	RexhT °C	LexhT °C	NOx	CO	C02
		RPM	HP	"bTDC			ppm	PPm	%
1	1.45	750	682	24	470.3	470.25	244	380	6.56
2	1.4	750	682	24	473	473.125	426	440	6.76
3	1.35	750	682	24	479	478.875	917	490	7.17
4	1.3	750	682	24	483.87	483.125	1948	530	7.71
5	1.25	750	682	24	491.75	490.75	3331	580	8.27
6	1.2	750	682	24	504.5	497	4165	660	8.65
7	1.3	750	682	27.1	487.6	484	1929	530	7.71
8	1.3	750	682	21.9	487	484.5	2160	540	7.81
9	1.3	750	682	19.9	485.75	483.625	1867	520	7.67
10	1.3	750	682	18.8	487.25	483	2034	570	7.73
11	1.25	850	782.55	24	521.37	515.875	3188	620	8.25
12	1.3	850	782.55	24	513.75	510	2442	590	7.97
13	1.35	850	782.55	24	506.37	503.375	1387	550	7.5
14	1.4	850	782.55	24	500.87	498.625	838	520	7.15
15	1.45	850	782.55	24	498.62	496.75	442	500	6.82
16	1.3	850	782.55	27.1	514	510	2749	610	8.09
17	1.3	850	782.55	19.9	515.62	511.125	2511	590	7.99
18	1.25	950	886.22	24	556.12	552.75	3735	630	8.73
19	1.3	950	886.22	24	548	543.75	2303	570	8.09
20	1.35	950	886.22	24	541	537.5	1514	530	7.7
21	1.4	950	886.22	24	539	533.625	994	520	7.41
22	1.45	950	886.22	24	536.12	531.125	638	490	7.12
23	1.3	950	886.22	27.1	552,5	538.875	2633	570	8.17
24	____1,3	950	886.22	19.9	549.87	541.875	2416	550	8.13
25	1.25	1050	993	24	588.62	579	3446	950	8.61
26	1.3	1050	993	24	584.12	575.625	3404	640	8.66
27	1.35	1050	993	24	576.5	569	2256	560	8.06
28	1.4	1050	993	24	570.75	564	1608	530	7.74
29	1.45	1050	993	24	568	560.75	1053	510	7.41
30	1.3	1050	993	27.1	583.87	572.375	3461	610	8.51
31	1.3	1050	993	19.9	581.12	571.5	2942	590	8.36

Table 4.2. Engine operational and emission data

Twenty two samples were used to train each neural network, and the remaining samples were applied to the networks in order to verify the generalization.

4.2.5. Results

Following the four steps outlined in section 4.2.3, the engine was set at a stable condition (condition #25, Table 4.2). At this condition the NO_x emission level was approximately 3446 ppm.

Next, a NO_x emission target of 495 ppm was entered to the neural network PEM and control system. The tuning process sequence was established based on the following priorities: (1) 1.2< Lambda<1.45; (2) 700 RPM<Speed<l 100 RPM; (3) 650 HP<Load<1000HP; (4) 21°bTC<Spark Timing<27 °bTC. The system immediately estimated the optimal engine operational condition to reach the NO_x target, and generated the following results:

Lambda: 1.428; Speed=889.5 RPM; Load=841.3 HP; Spark timing 24°bTC. Using the existing engine management system, the engine was set at Lambda factor=1.43; Speed=900 RPM; Load=834 HP; Spark Timing 24°bTC, resulting in a NO_x concentration of 510 ppm. Clearly, an error of 3.9% was obtained. Several other targets were demanded and Table 4.3.a shows the results along with a comparison between the actual NO_x emission levels generated when setting the engine at very close operational conditions indicated by the control system. Table 4.3.b and Table 4.3.C present the same comparison for CO and CO_2 control.

It should be clarified that even though engine/compressor load is a control parameter, it proportionally varies with the engine speed. Experiments were limited to engine conditions where the load changed due to speed variation and not due to manual manipulation. However, the neural network system simulated this effect and it corresponded to the operational behavior of the real engine.

Xa	Speeda	Loada	SparkTa	Ae	Speede	Loade	SparkTe	AEmi (ppm)	EEmi
1.38	800	731.88	24	1.37	780	725 _j	24	970	990
1.32	800	731.88	24	1.32	785	730	24	1769	1750
1.27	800	731.88	24	1.26	795	730	24	2806	2890
1.43	900	834	24	1.44	895	830	24	802	840
1.38	900	834	24	1.39	895	825	24-	1038	990
1.32	900	834	24	1.33	910	830	24	1728	1715
1.27	900	834	24	1.27	900	835	24	2977	2963

Table 4.3.a. Comparison between actual engine operational conditons (Xa,Speeda, Loada.SparkTa) and NO_x emission level (AEmi) to estimated engine operational conditions (Xa, Speeda, Loadz,SparkTa) and NO_x emission level (EEmi).

Xa	Speeda	Loada	SparkTa	Xe	Speede	Loade	SparkTe	AEmi (ppm)	EEmi
1.43	800	731.88	24	1.42	795	735	24	520	510
1.38	800	731.88	24	1.38	810	730	24	520	528
1.32	800	731.88	24	1.3	799	725	24	560	590
1.27	800	731.88	24	1.25	800	735	24	600	595
1.43	900	834	24	1.42	905	830	24	510	525
1.38	900	834	24	1.38	890	825	24	550	540
1.32	900	834	24	1.3	900	840	24	650	665
1.27	900	834	24	1.28	890	840	24	1370	1405

Table 4.3.b. Comparison between actual engine operational conditons (Xa, Speeda, Loada, SparkTa) and CO emission levels (AEmi) to estimated engine operational conditions (Xa,Speeda, Loadz,SparkTa) and CO emission levels (EEmi).

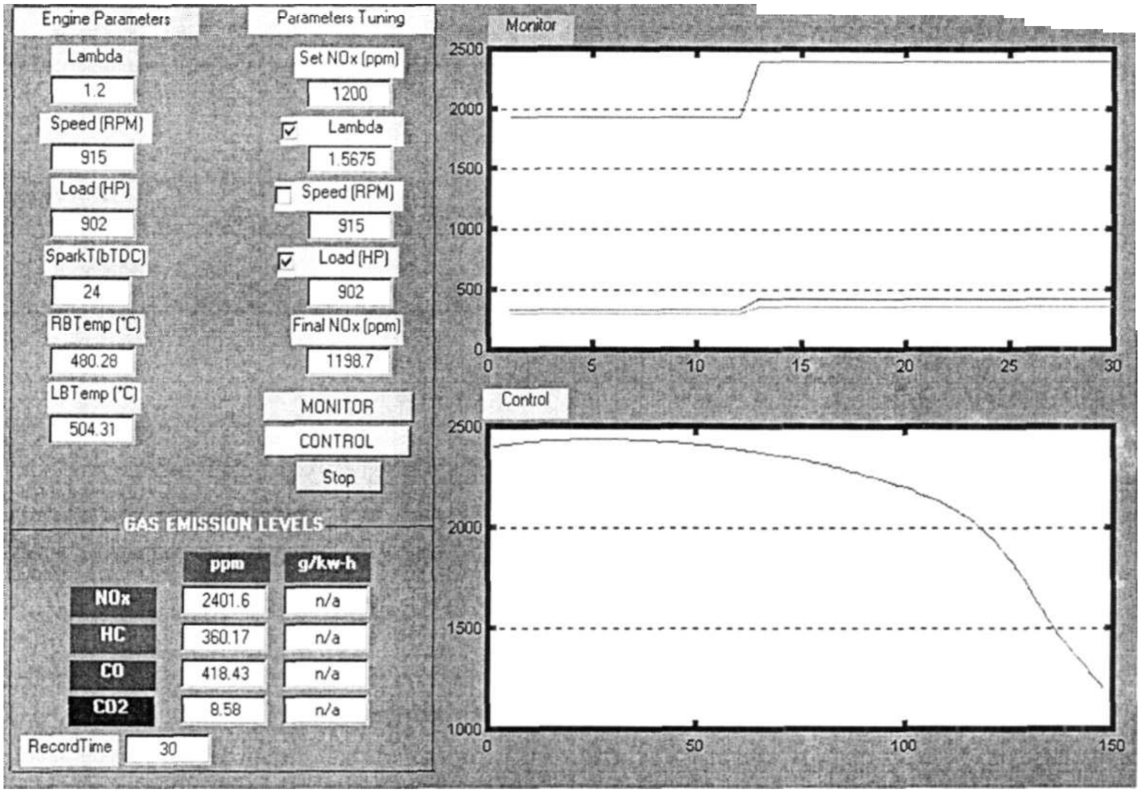
Xa	Speeda	Loada	SparkTa	Xe	Speede	Loade	SparkTe	AEmi (%)	EEmi
1.43	800	731.88	24	1.44	800	730	24	7.08	7.11
1.38	800	731.88	24	1.38	810	735	24	7.22	7.15
1.32	800	731.88	24	1.3	800	735	24	7.67	7.9
1.27	800	731.88	24	1.28	790	735	24	8.17	8.23
1.43	900	834	24	1.43	895	825	24	7.1	7.15
1.38	900	834	24	1.35	805	840	24	7.32	7.35
1.32	900	834	24	1.31	800	845	24	7.82	7.75
1.27	900	834	24	1.27	900	830	24	8.37	8.25

*Table 4.3.**C.** Comparison between actual engine operational conditions (Xa, Speeda, Loada,SparkTa) and CO2 emission level (AEmi) to estimated engine operational conditions (Xa,Speeda,Loadz,SparkTa) and CO2 emission level (EEmi).*

Maximal relative errors of 4.87%, 5.02% and 3.96% were obtained after comparing EEmissions (estimated emission levels) and AEmissions (actual emission levels) in Tables 4.3.a, 4.3.b., and 4.3.c, respectively. As a result, the estimation process complied with the demanded goals, thus showing excellent generalization.

Limited engine operational conditions were present during engine mapping. Therefore, the engine load could not be varied in a wide range due to production schedules. Additionally, spark timing is generally set and kept at 24 "bTC for all engine operational conditions.

Figure 4.3 shows the developed graphic user interface for emission monitoring and control. Engine operating parameters and the resulting emission levels were visualized, numerically and graphically. Subscreen A (top) shows the emission levels per sample, and subscreen B (below) shows the process of achieving an emission target. The user can enter emission type and desired level, and the system automatically starts searching the appropriate engine condition according to the pre-selected engine control parameters until the target level is reached. In cases where emission targets cannot be met, the system generates the closest operational condition to reach the closest emission level demanded by the user. Additionally, GUI includes datasheets where the user can manually input the engine and emission data.



Engine Parameters

Lambda: 1.2

Speed (RPM): 915

Load (HP): 902

SparkT(bTDC): 24

RBTemp (°C): 480.28

LBTemp (°C): 504.31

Parameters Tuning

Set NOx (ppm): 1200

☑ Lambda: 1.5675

□ Speed (RPM): 915

☑ Load (HP): 902

Final NOx (ppm): 1198.7

GAS EMISSION LEVELS

	ppm	g/kw-h
NOx	2401.6	n/a
HC	360.17	n/a
CO	418.43	n/a
CO2	8.58	n/a

RecordTime: 30

Figure 4.3. Graphic user interface (GUI) for emission monitoring and control

CHAPTER V: SYSTEM INTEGRATION

5. Integration of the neural network system for estimation and control of gas emissions with an existing engine management

The final goal of this project was to investigate and propose a methodology to integrate the developed emission estimation and control system with the existing engine management system, REMVue, provided by REM Technology Inc.

Integration of the system is considered a very important step, because it will allow REMVue operators to not only efficiently monitor and control the engine performance, but also to perform on-line emission monitoring and open-loop control without the use of any external sampling equipment. Furthermore, description of the REMVue system is provided so that a clearer concept for the integration phase can be outlined.

On the other hand, complete system integration with the REMVue system is not pursued, since the present system could be interfaced with any existing engine management system such as Programmable Logic Controllers (PLC), Distributed Control Systems (DCS) and some standard or customized data acquisition systems, so that a stand-alone application can originate from the proposed system. Therefore, description of building a stand-alone application is presented in order to explain the feasibility of integration.

5.1.1. Engine management system

The engine management system REMVue (REM Technology, Port Coquitlam, BC) can be defined as a system capable of effectively monitoring and controlling key engine and compressor parameters using a unique air-fuel ratio control and optimization, on-line diagnostics and safety shutdown control system in a single package [42].

Its improved operating performance results in remarkable benefits in terms of reduction of fuel consumption with optimized horsepower, reduction of engine emissions, and minimal maintenance costs. In addition, the system can be configured to interoperate in conjunction with other hardware or software systems.

A. REMVue software

The REMVue system consists of a software, which operates on 586 IBM 133-MHz processor, and computer hardware which contains main computer board, and additional 32 Mbytes RAM. The main computer board has an Ethernet port, 2 serial ports and connectors for monitor, parallel port, and keyboard. Additional cards provide extra serial ports, modem, non-volatile memory (NVM), analog to digital (A/D) converters for the process and diagnostics input signals, and one or more digital to analog (D/A) output cards. An external watchdog verifies the operation of the computer.

NVM retains data without electrical power, whereas data stored on RAM is lost when the power is turned off. In the REMVue system the NVM is in the form of flash memory PCMCIA card (often referred to as flash disk), which is removable.

The flash disk contains ROM BIOS, the QNX operating system, the REMVue embedded software, configuration data, and data stored by the REMVue software during its normal operation. Figure 6.1 presents a block diagram of the REMVue software structure.

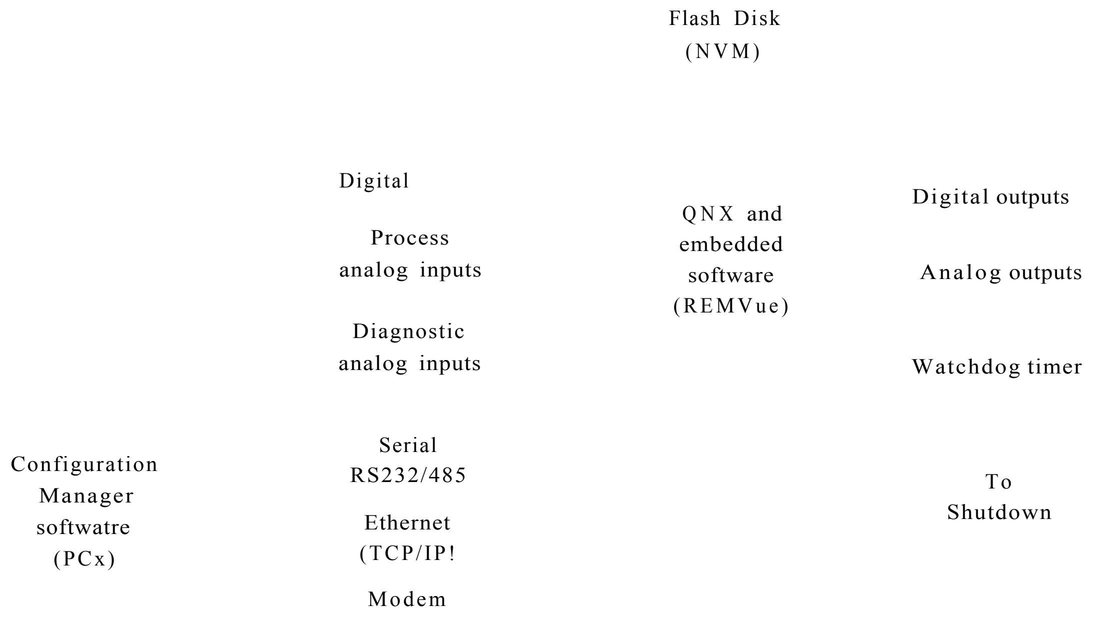


Figure 5.1. Block diagram of the REMVue software system

The configuration data is usually generated on a separate computer (PCx) and then is transferred to the REMVue using one of the communication links. The embedded software uses Configuration Manager software, which is responsible for obtaining the desired input data and performing the action specified in the configuration data.

When the REMVue computer is turned on, it uses the ROM BIOS information to start (boot-up). This loads the operating system and the embedded software from the flash disk and automatically starts the REMVue embedded software.

B. Operating system

The operating system is the interface between the REMVue software and the computer hardware. The REMVue uses the QNX real-time software [42], which is responsible for scheduling various software tasks (e.g., send data to a specific port). Each software task within the REMVue embedded software has a priority rank, so that tasks with high priority can be performed before tasks with a low priority. In addition, tasks can be interrupted, while another task is processed. This feature ensures that the computer does not halt other tasks when any particular task awaits a response.

The QNX operating system has been widely used on time-critical systems for over 30 years [42], and has proven to be a reliable platform for the REMVue software.

C. REMVue Embedded Software

The REMVue embedded software performs the following main tasks: (1) Collection of analog process data from the A/D converter card; (2) Collection of analog diagnostic data from the A/D converter card; (3) Collection of inputs (e.g. contact openings/closures) from the digital input card; (4) Placing the input data on the assigned registers; (5) Calculation of results using the diagnostic data blocks; (6) Performing ladder logic sequences; (7) Sending outputs to the D/A card; (8) Sending outputs to the digital outputs (energize or de-energize relays); (9) Storing data on the flash disk; (10) Recalling data from the flash disk; (11) Sending data to serial ports; (12) Receiving data from serial ports; (13) Operating the Ethernet port; (14) Operating the modem, if installed; (15) Performing Proportional-Integral-Derivative (PID) calculations; (16) Updating the watchdog timer, and (17) various other tasks.

Analog process data (e.g., intake manifold temperature, atmospheric pressure) is collected by one A/D card, while the diagnostic data (e.g., dynamic cylinder pressures) is collected by separate A/D card.

Analog process data is collected in a channel by channel format. When the REMVue software requests a process analog input, the input channel is selected (a multiplexer is switched to the desired channel), one or more analog-to-digital conversions occur, and the results are averaged. Following a scaling to engineering units, the result is stored in an assigned register. The collection time required is normally less than 1 ms.

Analog diagnostic data must be collected in a continuous manner (no interruptions). The REMVue uses a proprietary method to ensure the diagnostic data is collected without affecting the operation of the other tasks being scheduled by the operating system. Typical diagnostic data includes cylinder pressure or vibration signal for a minimum of one full revolution. This data is acquired at 25,000 samples per second [42].

The REMVue embedded software incorporates several function blocks (process and diagnostic data blocks) which contain constants and equations that are used to calculate some parameters using analog input data located in the assigned registers. These blocks are created on a personal computer using C code using Configuration Manager software [42], and are stored in a project file database to be compiled and then transferred to the REMVue embedded software (See section 6.6.1(E)).

D. Registers and Coils

The REMVue software uses registers and coils extensively. Registers and coils refer to software locations where numeric or digital data may be stored [42].

Data contained in registers is used in ladder logic functions, for receiving analog input data, for providing analog data for outputs, and for interfacing with external devices.

A coil is a name for a bit. A coil can be "on" (bit=l) or "off (bit=0). Coils contain information used in the ladder logic, provide a storage location for digital inputs, location for digital output data, and for interfacing with external devices.

The Configuration Manager software allows the user to attach a description to registers and coils, thus aiding in the interfacing of other devices and in constructing the ladder logic.

E. Configuration data

The embedded REMVue software needs configuration data to be able to perform the desired tasks. Without the configuration data, the software will not perform any useful monitoring, control, communication or diagnostic tasks.

The configuration data includes: (1) Input and output card existence and number of channels; (2) Other hardware information such as serial channels, NVM size, and modem presence; (3) Description of the I/O channels; (4) Description of coils and registers; (5) Engine name and identification; (7) Engine specifications used in calculations; (8) Specification of calculations to be performed; (9) Creation of the sequence logic; (10) Set-up of the PID loops; (11) Specification of data to be stored on the NVM and frequency of storage; (12) Specification of alarm levels, and (13) Set-up of Modbus (serial protocol) communications to other devices.

As mentioned before, the configuration is created on a personal computer (PCx), and the data is stored in a Project File database. When the user has completed a configuration, a compilation step is required on the user computer where the information in the Project File database is converted to a format that can be used by the embedded software. The formatted information consists of several files known as system files. To install a new configuration on the REMVue, these system files must be downloaded to the REMVue via one of the communication ports (serial, modem or Ethernet). Ethernet is generally recommended because of the higher transfer speed. This Project File, which is stored in NVM, can be up-loaded at a later time from the REMVue to the Configuration Manager or to other software that needs the database information.

F. ASCII Files

ASCII files can be edited using a normal text editor. They can be read directly into MATLAB using the *load* function which reads the content of an ASCII file into a variable with the same name as the file (without the extension). If the ASCII data file has *m* lines with *n* numerical values on each line, the result is an *m-by-n* numerical array.

In addition, alternate value delimiters can be specified using the *dlmread* function. A delimiter is any character that separates the values within the ASCII file (e.g., semicolon). [Using MATLAB, Version 5, The Mathworks Inc., Natik, MA, U.S.A., 1998],

5.1.2. Building stand-alone external applications using MATLAB

A. MEX Files

The MATLAB Engine Library is a set of functions that allows the users to call MATLAB from their own programs, thereby employing MATLAB as a computational engine. MATLAB Engine programs are C or Fortran programs, which communicate with a separate MATLAB process via pipes (in UNIX), Dynamic Data Exchange (in MS-Windows and MATLAB 4.x), or ActiveX (in MS-Windows and MATLAB 5.x). There is a library of functions provided with MATLAB, which allows the users to start and end the MATLAB process, send data to and from MATLAB, and send commands to be processed in MATLAB [Matlab Compiler, Version 1.2, The Mathworks Inc., Natik, MA, U.S.A., 1998].

MATLAB MEX-fdes are dynamically linked subroutines that MATLAB interpreter can automatically load and execute. MEX-fdes are C or Fortran subroutines which, when compiled, can be called from MATLAB just like M-fdes and built-in functions. The advantages ofMEX-functions are running speed, the ability to incorporate already written C and Fortran code into MATLAB, and the functionality to access hardware.

B. MATLAB Compiler

MATLAB provides a toolbox, which allows for the generation of stand-alone external applications. Stand-alone external applications run without the help of the MATLAB interpreter. In fact, they can run even if MATLAB is not installed on the system that users wish to use (e.g., UNIX system).

MATLAB Compiler can generate the following source codes: (1) C source code for building MEX fdes; (2) C or C++ code for combining with other modules to form stand-alone external applications. Even though stand-alone external applications do not need MATLAB at runtime, MATLAB Compiler does require the C/C++ Math libraries to create applications that rely on the core math and data analysis capabilities ofMATLAB.

MATLAB compiler can be installed on UNIX, Macintosh, Windows 95 or Windows NT systems. System requirements, installation and configuration procedures for the MATLAB Compiler on these operating systems are well described.

Figure 6.2 illustrates the various ways a user can utilize MATLAB Compiler. The shaded blocks represent user-generated code; the unshaded blocks represent Compiler-generated code, and the remaining blocks (drop shadow) represent Mathworks or other vendor's tools.

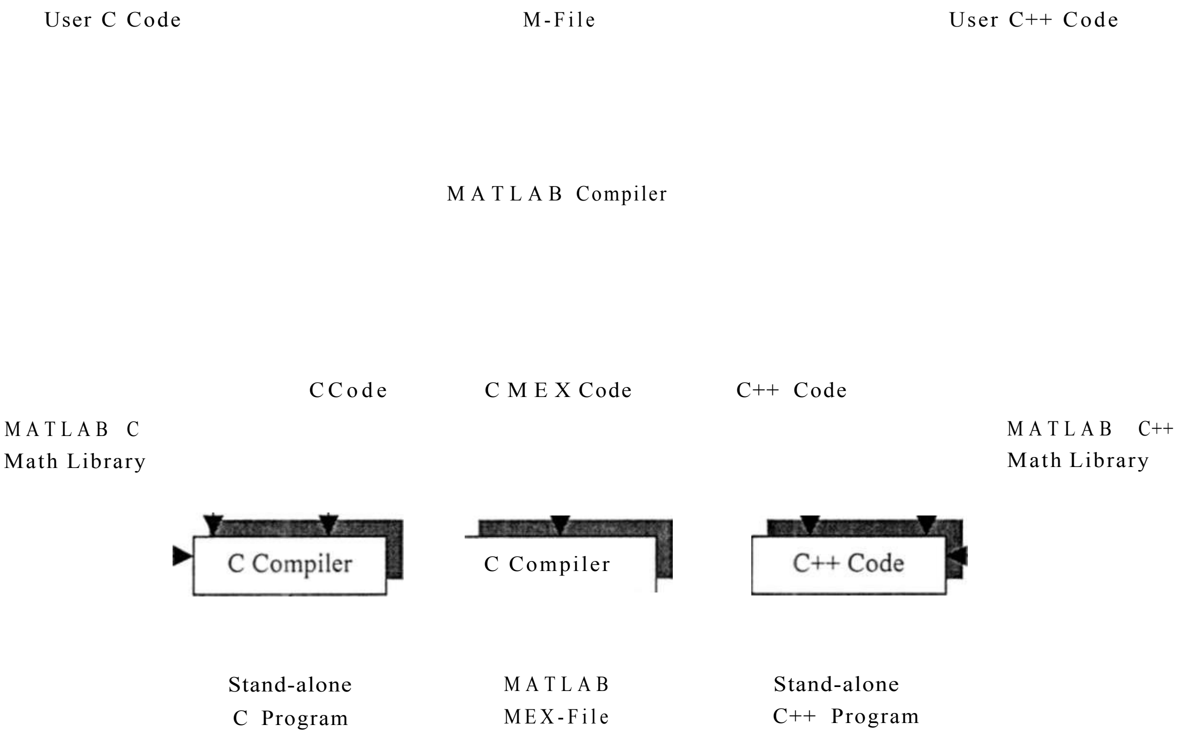


Figure 5.2. MATLAB Compiler applications

MATLAB M-files are ASCII text files that anyone can view and modify. MEX-files are binary files. Therefore, shipping MEX-files or stand-alone applications instead of M-files hides proprietary algorithms and prevents modification of the corresponding M-files. In addition, compiled C or C++ code speeds up M-file functions that contain loops, variables that the MATLAB compiler views as integers or real scalars, and operates on real data only [Matlab C++ Math Library, Version 1.2, The Mathworks Inc., Natik, MA, U.S.A., 1998].

C. Full Integration by Converting Neural Network M-files into C Program

As described in previous chapters, the neural network system for emission monitoring and control was developed using the MATLAB Neural Network toolbox. MATLAB compiler supports M-files generated using this toolbox, and therefore it is completely feasible to generate C source code from the developed M-files as required for the Project Profile management system.

In general, two blocks must be compiled to integrate them into the REMVue embedded software. The first block contains the designed neural network system for emission estimation. The second block contains the neural networks that represent emission control modules. Each block contains hundreds of weights and biases, plus the corresponding transfer functions for each neural network associated with each module. Sequence of calculations can be performed by specifying them in the Configuration Manager software, and can be stored in the Project File database.

As a result, the user can be capable of easily integrating different emission estimation and control modules into the existing REMVue system. Furthermore, due to the modularity of the neural network PEM system, new modules can be added to the existing system at any time without major computational effort. In other words, if an additional emission (e.g., CO2) is to be added to the system, the neural network PEM system would allow the user to add it, generate new neural networks related to the estimation and control of this new emission, and finally integrate the new modules to the REMVue system.

Figure 6.3 summarizes the overall process to build and integrate the neural network PEM system into the REMVue system, so that online emission estimation and control could be incorporated as new features offered by the REMVue system.

(1) Engine mapping: parameters
(control and monitoring), and emission
levels recording

4

(2) Enter samples into the NN PEM
system for training and validation

4

(3)Generation of the neural network modules encompassing the overall system for emission
estimation and control

(4) Compilation: Generation of C code

(5) Optimization/customization of C code

4

(6) Generated C code (compiled neural network modules) transferred into the
Project File (location: PCx)

4

(7) Define internal tasks: REMVue
Configuration Data (location: PCx)

!

(8) Generated C code + Configuration data from PCx, transferred to the
REMVue system (flash disk) via Ethernet (TCP/IP protocol)

Figure 5.3. General steps for the integraion of the neural network-based system with the REMVue

Configuration data can define what input parameters (lambda factor, speed, load or power, average exhaust temperatures, etc.) the new modules require in order to estimate the emission levels, so that information stored in specific registers can be accessed and mapped into these new modules. The output results (NO, CO, HC, CO2) can be stored in pre-defined registers, and then transferred into the RAM or NVM of the system. Due to the fact that emission regulations do not specify the sampling frequency, it is recommended to average and store emission levels at least once per second. This sampling frequency can be specified in the REMVue configuration data module.

The decision to define the on-line estimation and control process as a low priority REMVue task was made, so that the engine-compressor user would have the option to visualize the emission levels. However, information regarding daily emission level generation should be continuously recorded, stored and transferred periodically to a personal computer so that complete emission records can be analyzed to describe the machine performance, and to comply with the existing emission regulations related to continuous emission monitoring.

Alarm levels can be defined warning the engine operator when the engine is generating emission levels outside the pre-defined allowable limits.

On the other hand, the open-loop parameter control could be activated as a very helpful feature at the moment of changing engine operational conditions. Thus, the user can notice immediately how the new variations might affect the emission performance, and how the engine could effectively be tuned in order to keep the engine operating under the desired emission boundaries.

D. Partial Integration between REMVue and external PC

An alternate approach for system integration can be based on transferring ASCII data files from the REMVue system to an external PC. The idea is to configure the REMVue system so that ASCII data files containing data from various engine control and monitoring parameters could be transferred to an external PC running the neural network system using the MATLAB workspace. As mentioned before, ASCII data file can be loaded into the MATLAB workspace using a simple command (See Section 5.1.1 (F)). The REMVue system has the capability of generating ASCII data files and sending them to a specific hardware using TCP/IP protocol. Thus, complete independence can be obtained, and the neural network system can be implemented as a stand-alone application, which can be utilized not only by the REMVue system, but by other engine management systems as well.

Feasibility of integrating the REMVue system with an external PC that could run the emission estimation and control system in real-time overcomes the needs for joint of the developed neural network estimation and control system and the existing engine management system (See Section 5.1.2 (C))

CONCLUSION

A Parametric Emission Monitoring System (PEMS) can be a potential candidate to replace existing and traditional Continuous Emission Monitoring Systems (CEMS) for Spark Ignition Stationary Internal Combustion (SISIC) engines. After an extensive literature and patent survey, the decision to build a computer PEM model was pursued using two approaches: multidimensional arrays and artificial neural networks. Limitations were encountered when evaluating real-time implementation requirements for the first approach.

The PEMS based on multidimensional arrays could be implemented as an off-line solution for representing a specific engine behavior and its relationship with emission generation. Thus, this technique provided a good solution to obtain the so-called engine emission signature, facilitating engine operators to understand how emission levels vary at different engine operational conditions. Engine mapping provided the necessary data to build multidimensional arrays that virtually contained all possible engine operational conditions and corresponding emission levels in a pre-established operational range. Real-time implementation was evaluated and considered impractical due to the amount of data that these arrays needed to contain to achieve the required system resolution.

A second PEM model was developed using artificial neural networks. The model proved that artificial intelligence could be applied to solve this particular problem. Three learning algorithms were used to train the different neural networks encompassing the overall system, and demonstrated effective training. Comparison between the different learning algorithms was pursued and showed that the Bayesian Regularization method presented the best performance. A Growing Network technique was implemented in order to find the best size for the different feedforward neural networks in charge of estimating the emissions of greatest concern generated from SISIC engines. In general, all neural networks resulted in mid-size networks containing from two to three layers, each layer presenting no more than thirty neurons. In the present study, the estimation of NO_x, CO, HC and $CO2$ was achieved by constructing, training, and validating four different neural networks, which finally were interconnected to work in parallel for monitoring purposes. The emission estimation depended on six engine parameters, four control engine parameters (Lambda factor, Speed, Load, and Spark Timing) and two monitoring parameters (Right Bank Temperature and Left Bank Temperature).

An Emission Control System was developed using a similar methodology to the one followed to build the neural network-based PEMS. The system aimed at estimating optimal engine conditions according to a desired emission level. Engine parameters tuned by the system were Lambda factor, Speed, Load and Spark Timing.

A preliminary study showed that the neural network PEM and control system can be implemented in real-time, or integrated with the existing engine management system REMVue (REM Technology, Port Coquitlam, Canada).

On the other hand, some facts must be addressed before closing this section:

- During the overall study, emission levels were recorded using the same gas analyzer so that results could be consistent at all times. However, the PEMS was limited at the measurements generated by the same analyzer, believing that its conditions satisfied environmental regulations. Third party personnel pursued calibration of the device each time a data collection was scheduled, but frequently problems arose due to erroneous readings, probably caused by internal toxic sensor/cells faults.

For future tests or engine data collection, a detailed protocol for analyzer calibration must be developed and conducted by the same person in charge of collecting the data needed to build the multidimensional arrays or train the neural networks.

- The developed PEMS could only model emission levels from SISIC engines in a specific engine operational range. Due to production schedules and additional restrictions, engine data collection was limited, but fortunately sufficient to demonstrate the feasibility of using both multidimensional arrays and artificial neural networks for emission estimation and control.
- The developed PEM systems depended on selected control and monitoring parameters provided by the engine management system (REMVue). In the event of a corrupted value of a certain parameter provided by the REMVue system, the PEMS will continuously estimate the emissions based on a corrupted signal, therefore, generating false emission levels. Likewise, this condition will directly affect the right control of the engine operation to comply with a desired emission level. Future work is necessary to implement a sensor validation module capable of detecting corrupted signals to eliminate the possibility of emission monitoring corruption.
- Due to limited engine data collection, additional parameters that might affect emission levels were assumed constant. These parameters are atmospheric temperature, pressure and humidity [1].

- Additional research is suggested to analyze how the developed PEM systems could be extended to units with similar mechanical characteristics (same model) but located at different sites. In other words, it would be interesting to verify whether a unique PEM system could be develop for a specific engine model. Thus, the economical benefits of a PEMS can be remarkable extended over traditional CEMS.

REFERENCES

[I] Sorge, G.: " A White Paper on Gas Engine Emissions Technology", Waukesha Engine Division, Dresser Industries, Inc., Wisconsin, U.S.A. (1995)

[2] Malm, H.: " A White Paper on Air-Fuel Control and Emissions for Gas Engines", REM Technology Inc. (1999)

[3] Ladommatos, N., Abdelhalim, S., and Zhao, H.: " Control of oxides of nitrogen from diesel engines using diluents while minimizing the impact on particulate pollutants", Journal of Applied Thermal Engineering, V. 16, N. 3, PP: 963-980 (1998)

[4] Hisayo, D., Katsukiko, K., and Shigenori, I.: " Air-Fuel Ratio Control System for Internal Combustion Engines," U.S. Patent No. 5,784,879 (1998)

[5] Axel Otto zur Loye: " Cylinder Pressure based Air-Fuel Ratio and Engine Control, " U.S. Patent No. 5,765,532 (1998)

[6] Frith A.M., Gent, C.R., and Beaumont, A.J.: " Adaptive Control of Gasoline Engine Air-Fuel Ratio using Artificial Neural Networks," In: Proceedings of IEE Conference No. 409 (1995)

[7] Buchop, T., D'Alleva, R., Darnel), R.K., and Pedersen, C.T.: "Parametric Emission Monitoring System having Operating Deviation Feedback," U.S. Patent No. 5,703,777 (1997)

[8] Heater, W., Home, T., and Gillete, A.: " Development of a Parameter-Based Method for Calculating Exhaust Emissions from Reciprocating Natural Gas Transmission Engines," Journal of Natural Gas and Alternative Fuels for Engines, ASME-ICE V. 21 (1994)

[9] Keller, J., Havener, J., Godbole, D., and Fegurson, R.: " Virtual Emission Monitoring System with Sensor Validation", U.S. Patent No. 5, 386, 373 (1995)

[10] Hunt, F., Kaneyasu, M., and Saikalis, G.: "Engine and Emission Monitoring and Control System utilizing Gas Sensors," U.S. Patent No. 5,426,934 (1995)

[II] Weisbrod, J.: " A Practical Approach to PEMS Development and Application," In: Proceedings of the Gas Machinery Conference, Colorado, U.S.A. (1998)

[12] Goto, S., and Fukuda, T.: "Study of NO_x Formation and Reduction of Lean-Burn Spark Ignition Gas Engine," Journal of the Society of American Engineers, V. 19, N.14, PP: 351-354 (1998).

[13] Heywood, J.: *Internal combustion Engine Fundamentals,* McGraw Hill Inc. (1998)

[14] Pulkrabek, W.: Engineering Fundamentals of the Internal Combustion Engine, Prentice Hall Inc. (1997)

[15] U.S. EPA: " Enhanced Monitoring Program Proposal," 40CFR Parts 51, 52, 60, 61, and 64 (1993).

[16] Clean Air Act Amendments, http://www.epa.gov/oar/oaq caa.htmk U.S.A (1990)

[17] Environment Canada and U.S. Environmental Protection Agency, *Air Quality Agreement - Progress Report,* Sections I, II, III, and IV. (1998)

[18] Environmental Protection Agency, www.epa.gov., California, U.S.A

[19] Pendlebury et al., "Sensor Technology Update", Automotive Engineering, September 1996.

[20] York, K. and Lewallen, S., " New Technology Provides Expanded Role for Portable Emissions Analyzers", Diesel Progress Engines & Drives, February 1995.

[21] TechMasters, http://home.att.net/~TechMasters/Sensors.html#electro, U.S.A

[22] Drager, http://www.draeger.com/english/st/gdt/gds/principles/ec.htm, Germany.

[23] Crowcon Detection Instruments Ltd., http://www.crowcon.com, U.S.A

[24] NASA Lewis Research Center, http://www.lerc.nasa.gov/WWW/chemsensors/ Ohio, U.S.A.

[25] Environmental Protection Agency, CFR-Promulgated test methods, http://www.epa.gov/ttn/emc/promgate.html, U.S.A

[26] Snyder, B., Neulicht, R.," Continuous Parametric Emission Monitoring System for Gas Turbines", Gas Research Institute, Report Number: GRI-96/0277, Illinois, U.S.A (1996)

[27] Chul Yoo, B., Sik Yoon, K, and Hae Lee, B., " Study on the Prediction of Performance and Emission in a 4-cylinder 4-stroke Cycle Turbocharged Gasoline Engine,"In: Proceedings of the 6th International Pacific Conference on Automotive, Seoul, South, SAE 912471 (1991)

[28] Ouenou-Gammo S., Ouladsine, M., and Rachid A., " Measurement and Prediction of Diesel Engine Exhaust Emissions", ISA Transactions 37, 135-140 (1998)

[29] [Bassam, EL: "Systems Approach for Modeling mechatronics systems," CIRP Annals-Manufacturing Technology, 47(1), p.p.: 431-436 (1998).

[30] Grun, P., Balasa, F., and Datt, N.: " Memory Size estimation for multimedia Applications," In: Proceedings of the 1998 6th IEEE International Workshop on Hardware/Software Coding, Seattle, WA, U.S.A (1998).

[31] Haykin, S.: *Neural Networks-A Comprehensive Foundation,* Prentice Hall, New Jersey (1999).

[32] Hagan, M.T., Demuth, H.B., and Beale, M., *Neural Network Design,* PWS Publishing Co. (1996)

[33] Demuth, H., and Beale, M.: *Neural Network Toolbox for use with MATLAB,* The Mathworks Inc., Natik, MA, U.S.A (1998).

[34] Orr, G.B., and Miiller, K-R.: *Neural Networks: Tricks of the Trade,* Springer-Verlag (1998)

[35] Marek, D.S., and Durachova, D.: " Investigation of the use of Back-Propagation Algorithm for CNN learning for Edge Extraction," In: Neural Networks World Proceedings, and SCANN'98, vol.8, No.5, Kosice, Slovakia, Czech Republic, pp.: 531-538 (1998)

[36] Dehan. L., and Chen, W.: " Active Back-Propagation Algorithm based on Adjusting Error for Multilayer Feed-Forward," Journal of Aeronautics and Astronauts, Beijing University, Beijing, China, pp.: 350-353 (1998)

[37] Hagan, M.T., and Menhaj, M.: "Training Fee-Forward Networks with the Maquardt Algorithm," IEEE Transactions on Neural Networks, vol. 5, No. 6, pp. 989-993 (1994)

[38] Dan Forsee, F., and Hagan, M.: " Gauss-Newton Approximation to Bayesian Learning," In: Proceedings of the IEEE International Conference on Neural Networks (1997)

[39] Grimaldi, C.N., and Mariani F.: " On Line Working Neural Estimator for SI Engines Operational Parameters," In: Proceedings of SAE, No. 2000-01-1247, Detroit, U.S.A (2000).

[40] Barker, T.: " Catalytic System Introduced for Continuous Emissions Monitoring," Turbomachnery International, vol.18, No. 5 (1999)

[41] Canada Environmental Protection Agency, http://vvww.ec.gc.ca , Canada

[42] REM Technology, Inc., http://wwvv7.remtechnology.com/products/overview.htm, Port Coquitlam, B.C., Canada

[43] QNX tutorial, http://www.cpc.wmin.ac.uk/~drotos/qnx/whatis/, Centre for Parallel Computing, University of Westminster, London, U.K.

APPENDIX A

BACKPROPAGATION LEARNING ALGORITHMS
USING NEURAL NETWORK TOOLBOX
MATLAB 5.3.1

The neural network-based emission monitoring and control system was designed, modelled, and tested using three backpropagation learning algorithms (Quasi-Newton, Levenberg Maquardt, and Bayesian Regularization), which were implemented using the Neural Network toolbox ofMATLAB 5.3.0.

A detailed description of the implementation of these learning algorithms is provided.

A.1. TRAINBFG BFGS Quasi-Newton backpropagation.

Syntax

```
    [net,tr]=trainbfg(net,Pd,Tl,Ai,Q,TS,VV)
    info = trainbfg(code)
```

Description

TRAINBFG is a network training function that updates weight and bias values according to the BFGS quasi-Newton method.

TRAINBFG(NET,Pd,Tl,Ai,Q,TS,VV,TV) takes these inputs,

NET - Neural network.

Pd - Delayed input vectors.

Tl - Layer target vectors.

Ai - Initial input delay conditions.

Q - Batch size.

TS - Time steps.

VV - Either empty matrix [] or structure of validation vectors.

TV - Either empty matrix [] or structure of test vectors,

and returns,

NET - Trained network.

TR - Training record of various values over each epoch:

TR.epoch - Epoch number.

TR.perf - Training performance.

TR.vperf - Validation performance.

TR.tperf - Test performance.

Training occurs according to the TRAINBFG's training parameters, shown here with their default values:

net.trainParam.epochs	100	Maximum number of epochs to train
net.trainParam.show	25	Epochs between showing progress
net.trainParam.goal	0	Performance goal
net.trainParam.time	inf	Maximum time to train in seconds
net. trainParam. min_grad	le-6	Minimum performance gradient
net.trainParam.max_fail	5	Maximum validation failures
net.trainParam.searchFcn	'srchch '	Name of line search routine to use.

Parameters related to line search methods (not all used for all methods):

net.trainParam.scal_tol	20	Divide into delta to determine tolerance for linear search,
net.trainParam.alpha	0.001	Scale factor which determines sufficient reduction in perf.
net.trainParam.beta	0.1	Scale factor which determines sufficiently large step size.
net.trainParam.delta	0.01	Initial step size in interval location step.
net.trainParam.gama	0.1	Parameter to avoid small reductions in performance
net.trainParam.low_lim	0.1	Lower limit on change in step size.
net.trainParam.up_lim	0.5	Upper limit on change in step size.
net.trainParam.maxstep	100	Maximum step length.
net.trainParam.minstep	1.0e-6	Minimum step length.

net.trainParam.bmax 26 Maximum step size.

Dimensions for these variables are:

Pd - No x Ni xTS cell array, each element P{i,j,ts} is a Dij x Q matrix.

Tl - Nl x TS cell array, each element P{i,ts} is a Vi x Q matrix.

Ai - N l x LD cell array, each element Ai{i,k} is an Si x Q matrix.

Where

Ni = net.numlnputs

Nl = net.numLayers

LD = net.numLayerDelays

Ri = net.inputs{i}.size

Si = net.layers{i}.size

Vi = net.targets {i}.size

Dij = Ri * length(net. input Weights {i,j}. delays)

If VV is not [], it must be a structure of validation vectors,

VV.PD - Validation delayed inputs.

VV.T1 - Validation layer targets.

VV.Ai - Validation initial input conditions.

VV.Q - Validation batch size.

VV.TS - Validation time steps,

which is used to stop training early if the network performance on the validation vectors fails to improve or remains the same for MAX_FAIL epochs in a row.

If TV is not [], it must be a structure of validation vectors,

TV.PD - Validation delayed inputs.

TV.T1 - Validation layer targets.

TV.Ai - Validation initial input conditions.

TV.Q - Validation batch size.

TV.TS - Validation time steps,

which is used to test the generalization capability of the trained network.

TRAINBFG(CODE) returns useful information for each CODE string:

'pnames' - Names of training parameters,

'pdefaults' - Default training parameters.

Network Use

You can create a standard network that uses TRAINBFG with NEWFF, NEWCF, or NEWELM.

To prepare a custom network to be trained with TRAINBFG:

1) Set NET.trainFcn to 'trainbfg'.

This will set NET.trainParam to TRAINBFG's default parameters.

2) Set NET.trainParam properties to desired valuesA

In either case, calling TRAIN with the resulting network will train the network with TRAINBFG.

Algorithm

TRAINBFG can train any network as long as its weight, net input, and transfer functions have derivative functions.

Backpropagation is used to calculate derivatives of performance PERF with respect to the weight and bias variables X. Each variable is adjusted according to the following:

X = X + a*dX;

where dX is the search *direction.* The parameter a is selected to minimize the performance along the search direction. The line search function searchFcn is used to locate the minimum point.

The first search direction is the negative of the gradient of performance.

In succeeding iterations the search direction is computed according to the following formula:

dX = -H\gX;

where gX is the gradient and H is an approximate Hessian matrix.

Training stops when any of these conditions occur:

1) The maximum number of EPOCHS (repetitions) is reached.

2) The maximum amount of TIME has been exceeded.

3) Performance has been minimized to the GOAL.

4) The performance gradient falls below MINGRAD.

5) Validation performance has increased more than MAX_FAIL times since the last time it decreased (when using validation).

A.2. TRAINLM Levenberg-Marquardt backpropagation.

Syntax

```
[net,tr] = trainlm(net,Pd,Tl,Ai,Q,TS,VV)
info = trainlm(code)
```

Description

TRAINLM is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization.

TRAINLM(NET,Pd,Tl,Ai,Q,TS,VV) takes these inputs,

NET - Neural network.

Pd - Delayed input vectors.

Tl - Layer target vectors.

Ai - Initial input delay conditions.

Q - Batch size.

TS - Time steps.

VV - Either empty matrix [] or structure of validation vectors,

and returns,

NET - Trained network.

TR - Training record ofvarious values over each epoch:

TR.epoch - Epoch number.

TR.perf - Training performance.

TR.vperf - Validation performance.

TR.tperf - Test performance.

TR.mu - Adaptive mu value.

Training occurs according to the TRAINLM's training parameters shown here with their default values:

net.trainParam.epochs	10	Maximum number of epochs to train
net.trainParam.goal	0	Performance goal
net.trainParam.lr	0.01	Learning rate
net.trainParam.max_fail	5	Maximum validation failures
net.trainParam.mem_reduc	1	Factor to use for memory/speed trade off.
net.trainParam.min_grad	le-10	Minimum performance gradient
net.trainParam.show	25	Epochs between showing progress
net.trainParam.time	inf	Maximum time to train in seconds

Dimensions for these variables are:

Pd - No x Ni x TS cell array, each element P{i,j,ts} is a Dij x Q matrix.

Tl - Nl x TS cell array, each element P{i,ts} is a Vi x Q matrix.

Ai - Nl x LD cell array, each element Ai{i,k} is an Si x Q matrix.

Where

Ni = net.numlnputs

Nl = net.numLayers

LD = net.numLayerDelays

Ri = net.inputs{i}.size

Si = net. layers {i}. size

Vi = net.targets{i}.size

Dij = Ri * length(net.inputWeights{i,j}.delays)

IfVV is not [], it must be a structure ofvalidation vectors, VV.PD - Validation delayed inputs.

VV.T1 - Validation layer targets.

VV.Ai - Validation initial input conditions.

VV.Q - Validation batch size.

VV.TS - Validation time steps,

which is used to stop training early if the network performance on the validation vectors fails to improve or remains the same for MAX_FAIL epochs in a row.

TRAINLM(CODE) return useful information for each CODE string:

'pnames' - Names of training parameters.

'pdefaults' - Default training parameters.

Network Use

You can create a standard network that uses TRAINLM with NEWFF, NEWCF, or NEWELM.

To prepare a custom network to be trained with TRAINLM:

1) Set NET.trainFcn to 'trainlm'.

This will set NET.trainParam to TRAINLM's default parameters.

2) Set NET.trainParam properties to desired values.

In either case, calling TRAIN with the resulting network will train the network with TRAINLM.

Algorithm

TRAINLM can train any network as long as its weight, net input, and transfer functions have derivative functions.

Backpropagation is used to calculate the Jacobian jX of performance PERF with respect to the weight and bias variables X. Each variable is adjusted according to Levenberg-Marquardt,

jj =jX * jX

je=jX*E

dX = -GJ+I*mu)\je

where E is all errors and I is the identity matrix.

The adaptive value MU is increased by MLMNC until the change above results in a reduced performance value. The change is then made to the network and mu is decreased by MU_DEC.

The parameter MEM_REDUC indicates how to use memory and speed to calculate the Jacobian jX. If MEM_REDUC is 1, then TRAINLM runs the fastest, but can require a lot of memory. Increasing MEM_REDUC to 2, cuts some of the memory required by a factor of two, but slows TRAINLM somewhat. Higher values continue to decrease theamount of memory needed and increase training times.

Training stops when any of these conditions occurs:

1) The maximum number of EPOCHS (repetitions) is reached.
2) The maximum amount of TIME has been exceeded.
3) Performance has been minimized to the GOAL.
4) The performance gradient falls below MINGRAD.
5) MU exceeds MU_MAX.
6) Validation performance has increased more than MAX_FAIL times since the last time it decreased (when using validation).

A.3. TRAINBR Bayesian Regulation backpropagation.

Syntax

```
[net,tr] = trainbr(net,Pd,Tl,Ai,Q,TS,VV)
info = trainbr(code)
```

Description

TRAINBR is a network training function that updates the weight and bias values according to Levenberg-Marquardt optimization. It minimizes a combination of squared errors and weights and, then determines the correct combination so as to produce a network which generalizes well. The process is called Bayesian regularization.

TRAINBR(NET,Pd,Tl,Ai,Q,TS,VV) takes these inputs,

NET - Neural network.

Pd - Delayed input vectors.

Tl - Layer target vectors.

Ai - Initial input delay conditions.

Q - Batch size.

TS - Time steps.

VV - Either empty matrix [] or structure of validation vectors,

and returns,

NET - Trained network.

TR - Training record of various values over each epoch:

TR.epoch - Epoch number.

TR.perf - Training performance.

TR.vperf - Validation performance.

TR.tperf - Test performance.

TR.mu - Adaptive mu value.

Training occurs according to the TRAINLM's training parameters, shown here with their default values:

net.trainParam.epochs	100	Maximum number of epochs to train
net.trainParam.goal	0	Performance goal
net.trainParam.mu	0.005	Marquardt adjustment parameter
net.trainParam.mu_dec	0.1	Decrease factor for mu
net.trainParam.mu_inc	10	Increase factor for mu
net.trainParam.mu_max	le-10	Maximum value for mu
net.trainParam.max_fail	5	Maximum validation failures
net.trainParam.mem_reduc	1	Factor to use for memory/speed trade off.
net.trainParam.min_grad	le-10	Minimum performance gradient
net.trainParam.show	25	Epochs between showing progress
net.trainParam.time	inf	Maximum time to train in seconds

Dimensions for these variables are:

Pd - No x Ni x TS cell array, each element P{i,j,ts} is a Dij x Q matrix.

Tl - Nl x TS cell array, each element P{i.ts} is a Vi x Q matrix.

Ai - Nl x LD cell array, each element Ai{i,k} is an Si x Q matrix.

Where

Ni = net.numlnputs

Nl = net.numLayers

LD = net.numLayerDelays

Ri = net.inputs{i}.size

Si = net.layers {i}.size

Vi = net. targets(i). size

Dij = Ri * length(net.input Weights {i,j}.delays)

If VV is not [], it must be a structure of validation vectors,

VV.PD - Validation delayed inputs.

VV.T1 - Validation layer targets.

VV.Ai - Validation initial input conditions.

VV.Q - Validation batch size.

VV.TS - Validation time steps,

which is normally used to stop training early if the network performance on the validation vectors fails to improve or remains the same for MAX_FAIL epochs in a row. This early stopping is not used for TRAINBR, but the validation performance is computed for analysis purposes if VV is not [].

TRALNBR(CODE) returns useful information for each CODE string:

'pnames' - Names of training parameters,

'pdefaults' - Default training parameters.

Network Use

You can create a standard network that uses TRAINBR with NEWFF. NEWCF, or NEWELM.

To prepare a custom network to be trained with TRAFNBR:

1) Set NET.trainFcn to 'trainlm'.
This will set NET.trainParam to TRAINBR's default parameters.

2) Set NET.trainParam properties to desired values.

In either case, calling TRAIN with the resulting network will train the network with TRAINBR.

Algorithm

TRAINBR can train any network as long as its weight, net input, and transfer functions have derivative functions.

Bayesian regularization minimizes a linear combination of squared errors and weights. It also modifies the linear combination so that at the end of training the resulting network has good generalization qualities.

This Bayesian regularization takes place within the Levenberg-Marquardt algorithm. Backpropagation is used to calculate the Jacobian jX of performance PERF with respect to the weight and bias variables X. Each variable is adjusted according to Levenberg-Marquardt,

jj=jx*jx
je=jX*E
dX = -Gj+I*mu) \je

where E is all errors and I is the identity matrix.

The adaptive value MU is increased by MU_LNC until the change shown above results in a reduced performance value. The change is then made to the network and mu is decreased by MU_DEC.

The parameter MEM_REDUC indicates how to use memory and speed to calculate the Jacobian jX. If MEM_REDUC is 1, then TRAINLM runs the fastest, but can require a lot of memory. Increasing MEM_REDUC to 2 cuts some of the memory required by a factor of two, but slows TRAINLM somewhat. Higher values continue to decrease the amount of memory needed and increase the training times.

Training stops when any of these conditions occur:

1) The maximum number of EPOCHS (repetitions) is reached.

2) The maximum amount of TIME has been exceeded.

3) Performance has been minimized to the GOAL.

4) The performance gradient falls below MINGRAD.

5) MU exceeds MU_MAX.

6) Validation performance has increase more than MAX_FAIL times since the last time it decreased (when using validation).

APPENDIX B

ENGINE MAPPING PROCEDURE AND EXPERIMENTAL UNCERTAINTY

This section addresses important aspects related to the experiments carried out to obtain the required engine data for implementing and testing the PEM models described in the present work.

B.1. Description of apparatus

Engine operational data was collected at BP Energy, North Caroline Gas plant in September and November, 2000.

The machine used was a 16-cylinder, natural gas SISIC Waukesha engine, Model P9390GSIU, with a rated power of 1800 HP.

Different emission concentrations were recorded using a sophisticated gas analyzer (GA-40T plus, MADUR, Vienna, Austria). The analyzer consisted of two main parts: 1) the actual unit or body, which enclosed different gas cells (e.g., NO, CO) analog/digital circuitry, LCD screen, etc., and 2) a heated probe of approximately 3 meters that was connected to the engine exhaust pipe to take gas samples continuously.

This analyzer was capable of providing NO, NO_2, CO, CO_2, and HC levels. NO, NO_2, and CO_2 concentrations were expressed in parts per million (ppm), while CO and HC concentrations were expressed in percentage % (1%= 10,000 ppm).

The gas analyzer was preliminary calibrated by a third party , who provided a certified document indicating that the analyzer presented a calibration error of approximately ± 20 ppm for each gas.

B. 2. Engine Control and Emission Recording

The data was cpllected under the supervision of a REMVue specialist; the engine operating conditions were changed with the consent of the BP Energy operator.

The engine was set at different operational conditions using the existing engine management system (REMVue system). The system had a Human Machine Interface (HMI), which facilitated the variation of the different control parameters (Lambda factor, Speed, Load, and Spark Timing) one at a time (e.g., Lambda factor could be set at different values while other control parameters remained constant).

The gas analyzer was connected to the engine exhaust pipe throughout the duration of the overall experiment.

When setting a new operational condition, the engine took approximately from 3 to 4 minutes to become stable at that specific condition. Therefore, 5 minutes were allowed to record: 1) monitoring parameters (such as exhaust temperatures), provided that the REMVue system measured and averaged these parameters, and 2) emission levels that corresponded to any new engine operational condition, monitored by using the gas analyzer mentioned before.

The engine was mapped according to the following operational ranges:

Lambda factor (1.2 < X <1.6)
Speed (750<Speed<l 100 RPM)
Load (650<Load<1500 HP)*
Spark timing (18bTC<St<27bTC)

* Technician working for Spartan Controls Ltd.

'Variation of the engine load was limited due to production schedules

B.3. Uncertainty of experiments

In the present work, a maximal relative error (MRE) between the emissions estimate from the different artificial neural networks (ANN) and the actual emission levels from the gas analyzer was calculated, assuming that readings obtained from the REMVue system and from the gas analyzer were not corrupted during the collection process. However, additional uncertainty could be present due to the fact that every measurement has uncertainty, and that some measurements have higher level of uncertainty than others.

Recorded measurements corresponding to the selected control and monitoring parameters used to train and validate the ANN encompassing the developed system also exhibit some uncertainty. Therefore, when comparing the emissions estimate from the ANN to the actual emission levels from the gas analyzer, a problem arises as to how much the estimate is erroneous and how much is the actual measurement erroneous.

Uncertainty can be evaluated by calculating two different errors:

1) Random error: When measuring and recording a physical quantity (e.g., gas pressure), the REMVue system reports this quantity with some variation that is typically expressed as a deviated percentage of its mean value.
2) Systematic error: A recorded measurement may be erroneous due to a calibration error or a similar error that is unknown, but systematic (e.g., a pressure diaphragm may respond to pressure in a nonlinear manner which the measurement device ignores). The magnitude of this systematic error may be determined by instrument recalibration.

Therefore, the problem of the uncertainty of the developed neural network system can be defined as the error in the ANN emissions estimate due to random errors and to errors in assigning the various weights and biases even if there were no errors in the measurements.

A detailed study of different errors that might affect the overall error in the ANN emissions estimate would be beneficial. Uncertain measurements that should be taken into account are the physical quantities related to the parameters used to train and validate the different ANN (e.g., Lambda factor, Speed, Load, Spark timing, and Exhaust temperatures).