



**HAL**  
open science

## Realization of a Virtual Lambda Sensor on a Fixed Precision System

Pierre Amato, N. Cesario, M. Di Meglio, F. Pirozzi

► **To cite this version:**

Pierre Amato, N. Cesario, M. Di Meglio, F. Pirozzi. Realization of a Virtual Lambda Sensor on a Fixed Precision System. DATE'05, Mar 2005, Munich, Germany. pp.192-197. hal-00181847

**HAL Id: hal-00181847**

**<https://hal.science/hal-00181847v1>**

Submitted on 24 Oct 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Realization of a Virtual Lambda Sensor on a Fixed Precision System

P. Amato, N. Cesario, M. Di Meglio, F. Pirozzi  
STMicroelectronics  
SST - Methodology for Complexity  
Via Remo de Feo, 1 Arzano (Napoli)

## Abstract

*The aim of this work is to study the implementation feasibility of a VLS (Virtual Lambda Sensor) by a TSK (Takagi, Sugeno, Kang) singleton FIS (Fuzzy Inference System). Such a sensor could be used in a model based EMS (Engine Management System) for trade gasoline engines. FIS design target is to obtain a system with a fixed data representation (i.e. 10 bit) and a limited number of inputs, outputs, rules and membership.*

## 1. Introduction

In this work, we study the implementation feasibility of a Virtual Lambda Sensor (VLS) model by a TSK (Takagi, Sugeno, Kang) singleton FIS (Fuzzy Inference System). The VLS is a model able to forecast engine air-fuel ratio processing the cylinder pressure signal of the gasoline engine. In this work we study the implementation feasibility of an emulator of the trade *on-off* Lambda sensor and of an emulator of the trade linear Lambda sensor. The attribute *on-off* suggests us that the Lambda sensor determines only whether Lambda ratio is up or down the stoichiometric condition.

The target of our analysis is to develop the best solution, that is the best model of VLS, according to restrictive operating constraints. All data and FIS parameters must be transformed according to constraints on data representation (i.e. 10 bit parameter resolution). To validate the model we used both Matlab simulations with hardware real constraints and real-time experiments on a trade 125cc gasoline engine.

### 1.1. Virtual Lambda Sensor

In the last years, the engine manufacturers guideline is to achieve further reductions in the quantities of polluting gases emitted by the engine and a decrease in the engines fuel consumptions without vehicle characteristics, that are desirable to the driver, must be compromised. For these rea-

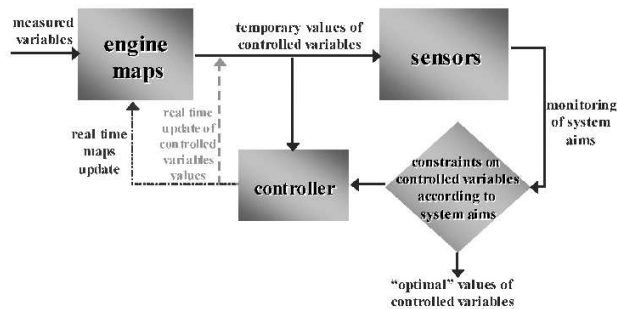
sons, an efficient engine control and a comprehensive monitoring of the engine working parameters are required.

To maintain a strict control of the engines working parameters Engine Management Systems (EMS) are used. The EMS implements control strategies which achieve the optimum trade-off between several contradictory objectives. At the same time, in a spark-ignition engine, the EMS must bring the engine in an operating range in which the three-way catalytic converter can further reduce the harmful content of the exhaust gases. The EMS controls the amount of fuel injected in the engine combustion chamber (fuel pulse width), the point in the engine cycle the mixture air fuel is ignited (ignition timing) and other parameters in advanced engine designs, for example, the valve timings. The EMS determines values for these parameters from measured quantities such as speed, load torque, air mass flow rate, inlet-manifold pressure, temperatures at various points and throttle angle.

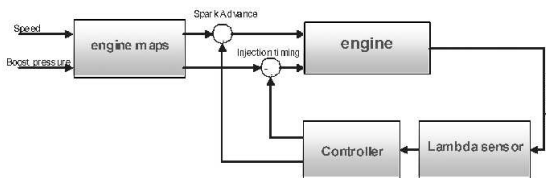
EMS is essentially composed by three components: engine maps, controller and sensors, see figure 1.

In the figure 2 other sensors appear too. These additional devices are able to monitor if the engine is working according to the *Systems Aims*, such as fuel economy, high output power and low emissions. So, they have an active part in the real time update process of controlled variables and of engine maps. For instance, in a spark-ignition engine a sensor like this is the Lambda sensor. This device, mounted in the exhaust stream, determines whether the Lambda ratio (i.e.  $AFR/AFR_s$  where  $AFR$  is the air/fuel ratio and the subscript  $s$  is for stoichiometric condition) is above or below unity from the amount of oxygen present. The EMS uses this information to adjust the fuel pulse width and/or the ignition timing to keep the Lambda ratio near to unity. In this value range of Lambda ratio, the three way catalytic converter, used in the current engines to reduce emissions levels to within legislative limits, “works in optimal conditions” (i.e. have the maximum percentage of conversion of exhaust toxic gases into less toxic products).

The last target of engine manufacturers is to achieve an enhanced engine control without additional sen-



**Figure 1. The block framework of a common EMS.**



**Figure 2. The EMS block framework for spark ignition engines.**

sory devices, that is without other costs. In these conditions, *virtual-sensors* techniques are much desirable. *Virtual-sensors* allow to estimate quantities of interest without real sensors dedicated to the measurements. In this field, intelligent system models, such as neural networks, fuzzy systems and neuro-fuzzy systems, are attractive because of their capabilities in pattern recognition and signal analysis problems [1].

### 1.2. Description of our Target FIS Constraints

We will consider a TSK singleton Fuzzy System, with some limiting constraints, as the target system

to map our VLS. TSK is a fuzzy system proposed by T. Tagaki, M. Sugeno and G.T. Kang [2, 3]. The output of each rule is a linear combination of input variables plus a constant term, and the final output is the weighted average of each rule's output. In this paradigm the rule set is made up of  $r$  rules having the following form:

*Rule i:* If  $x_1(t)$  is  $M_{i1}$ ,  $x_2(t)$  is  $M_{i2}, \dots$ , and  $x_n(t)$  is  $M_{in}$ , then  $y = w_{i0} + w_{i1}x_1(t) + w_{i2}x_2(t) + \dots + w_{in}x_n(t)$ . where  $x_1, x_2, \dots, x_n$  are the antecedent variables and  $y$  is the consequent variable. Moreover,  $M_{i1}, M_{i2}, \dots, M_{in}$  are fuzzy sets defined over the definition domains of  $x_1, x_2, \dots, x_n$  while  $w_{i0}, w_{i1}, \dots, w_{in}$  are constant coefficients which describe the linear relationship defined by the  $i$ -th rule set,  $i = 1, \dots, r$ . A TSK singleton is a fuzzy system in which the consequents are simply constants (*crisp*),  $w_{i0}$  with  $i = 1, \dots, r$ . From a linguistic point of view, this means we have the following simplified *if-then* rules:

*Rule i:* If  $x_1(t)$  is  $M_{i1}$ ,  $x_2(t)$  is  $M_{i2}, \dots$ , and  $x_n(t)$  is  $M_{in}$ , then  $y = w_{i0}$ .

For a sake of simplicity, we described TSK fuzzy systems with only one output but the system description can be merely extended to systems with vectorial outputs.

Our constraints are to use up to 256 *if-then* rules with up to 64 different serial inputs, 12 different outputs and 10 bit parameter resolution. TSK membership functions can be both triangular and trapezoidal. These constraints were suggested by a neuro-fuzzy peripheral for MCU (micro controller unit) that STMicroelectronics has been developing.

## 2. Description of STMicroelectronics Laboratory Equipment

The experimental tests have had an important role in the Lambda prediction and in the definition of the Lambda control strategy. Particularly they helped us to find the parameters of the pressure cycle used by the soft computing Lambda prediction algorithms. Moreover they allowed us to set the soft computing algorithms and to test the whole Lambda control system (Lambda prediction + Lambda control). The experimental activity was performed on a 125cc engine equipped with an AVL pressure sensor mounted on the cylinder head and with a TAG linear oxygen sensor in the exhaust system.

The STMicroelectronic test bench is also equipped with:

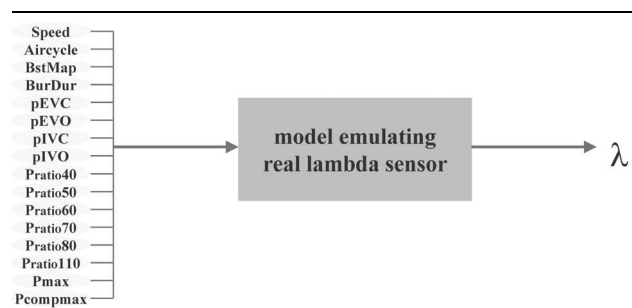
- Asynchronous Dynamometer AVL APA 102 (Power 120 kW)
- Test Bed and Instrumentation Automation System AVL PUMA 5.6.2

- Fast Prototyping Control System dSPACE
- AVL Fuel Balance with Fuel Conditioner
- Indicating System AVL Indiset Advanced 631
- Air Flow Meter
- Exhaust Gas Analyzer AVL DIGas 4000

The injection and the ignition systems were controlled by the Fast Prototyping Control System dSPACE . The dSPACE system was set to work either in “map mode” (the engine is controlled using the engine maps) that in “manual mode” (for every engine condition it is possible to change in real time all the control parameters). The dSPACE system was also used to implement and to test in real time the new Lambda control strategies.

### 3. Data Analysis

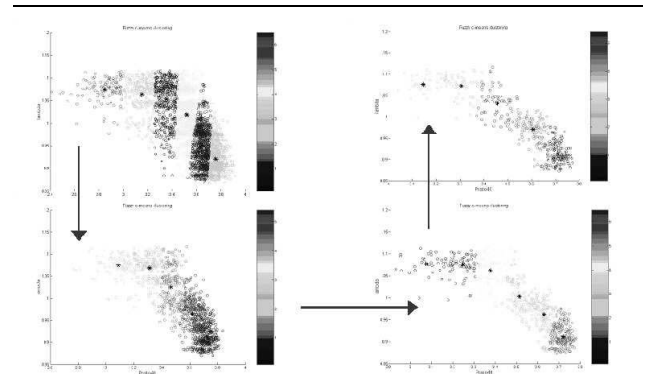
Our VLS models were tested on a data set measured by our laboratory on a real 125cc trade scooter engine. The



**Figure 3. List of the possible inputs for the VLS model.**

choosing of the TSK model inputs from our data set (see figure 3) was based on the the pressure cycle signal processing according to several literature algorithms in which *Pratio* are used as pressure signal features in order to forecast Lambda values (i.e. *Pratio* is the pressure ratio defined by  $P(\text{angle})/P(-\text{angle})$  where angle is a fixed crank angle).

To make a pre-processing of the data set we used clustering analysis. We analyzed the clustering process between  $Pratio_{40}$  (a possible input of the VLS model) and  $\lambda$  versus the number of the pressure cycle instantaneous values which we averaged on. Growing up this number, the correlation between  $Pratio_{40}$  and  $\lambda$  highly grows. In the figure 4 we can see how this correlation corresponds to cluster spherical shape. In this case, clusters cover only a bounded



**Figure 4. Clustering between  $\lambda$  and  $Pratio_{40}$ .**

range of variable domain such a way as a small input variation corresponds to a small output variation. The last condition is necessary in order to the model realization (i.e. building a mapping function) be a *well-posed* problem ([4], [5] and [6]).

A more detailed description of used clustering analysis was in [7]. Using this approach we found that the optimal number of instantaneous input values, to average on, is 16. We decided that our model must have three inputs chosen by a clustering algorithm. With this algorithm we analyzed the correlation between groups of three input variables, in our measure set, and the Lambda values to be forecast. The inputs chosen with this technique were respectively the maximum pressure, the pressure ratio at 40 and at 50 degree.

## 4. VLS Development

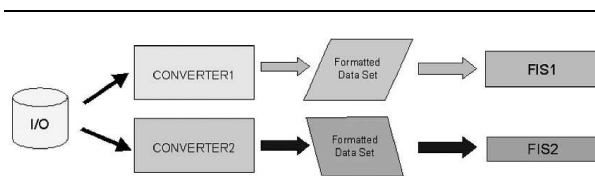
After the analysis of experimental data and the choice of the input variables, we developed our VLS models according to the constraints on data representation. Indeed, we normalized data values in the range  $[0, 1023]$ , that is, we used a data precision of 10 bits.

### 4.1. Workflow for VLS system development

The FIS system was made using pre-normalized integer data in  $[0, 1023]$ , this binds the other FIS parameters to stay in the same range. In this way, FIS performance was evaluated directly on several input data combinations. The formula used to calculate the normalized data values in the range  $[0, 1023]$  is the usual one ( $x_{\text{norm}} = (x - x_{\text{min}})/(x_{\text{max}} - x_{\text{min}})$ ).

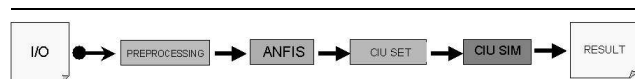
We previously fixed  $x_{\text{max}}$  and  $x_{\text{min}}$ , in this way successive data sets can have the same pre-processing. Data values less than  $x_{\text{min}}$  are in underflow, similarly values greater than  $x_{\text{max}}$  are in overflow. In this cases, they are respectively set to 0 and 1023. Considering the above remarks, it

is needful formatting data in a suitable way according to the problem to be resolved.



**Figure 5. The conceptual scheme of a solving process for two different problems.**

In figure 5 we can observe two different procedures to develop a FIS, modelling a system and fitting the constraints of a given device. We followed some approaches, which are complied to the above scheme, to solve both the problem of simulation of a linear Lambda sensor and the problem of simulation of an *on-off* Lambda sensor (i.e. classification problem). The implemented architecture, see scheme in the figure 6, allows us a rapid modelling process to create FIS and evaluating output in the current problem. We di-



**Figure 6. The conceptual scheme of data flow.**

vided the set of experimental data in the following sets:

- $\lambda < 0.95$  *very rich*
- $\lambda \geq 0.95$  and  $\lambda \leq 1$  *rich*
- $\lambda > 1$  and  $\lambda \leq 1.05$  *lean*
- $\lambda > 1.05$  *very lean*

Finally, we composed data set for FIS training, check and test at the same way. So we guaranteed that the FIS is trained with the same number of data for each region of Lambda values. Moreover, we can monitor model performance in each of the four regions of Lambda values.

#### 4.2. Model for a Virtual *linear* Lambda Sensor

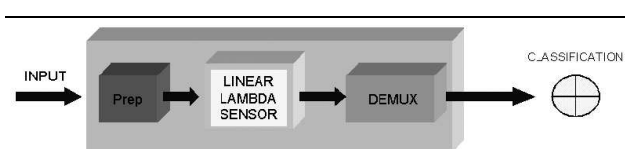
In order to develop a FIS model for a virtual linear Lambda sensor we used a technique proposed in [8], Matlab ANFIS tool. It is very important to highlight that the AN-

FIS procedure is not completely compliant to our specifications, in fact the ANFIS tool, working with a data precision of 32 bits, does not allow us to fix the data resolution to 10 bits. In order to work with data normalized in range  $[0, 1023]$ , we changed some parts of the ANFIS interface. Furthermore, the produced FIS was processed by an *ad hoc* “parser” developed by us, because it has a rule coding not compliant to our system simulator and consequent values not in range  $[0, 1023]$ .

The input data set was divided in three different sets. The 1st is for the FIS training, the 2nd for the validation and the 3rd for the testing. The percentage error was obtained scaling the absolute error by the maximum experimental data. Final results are: mean error equal to 0.0288 and maximum error equal to 0.1271.

#### 4.3. Model for a Virtual *On-Off* Lambda Sensor

From a mathematics viewpoint, the virtual *on-off* Lambda sensor design can be described as a classification problem. This problem, apparently simpler, can be approached in different ways. In this case a particular care must be taken in the choosing of a classifying way. The way we used to do this is to consider a linear sensor with a decoder circuit in the output, see figure 7.



**Figure 7. How to use a virtual linear sensor to classify Lambda values.**

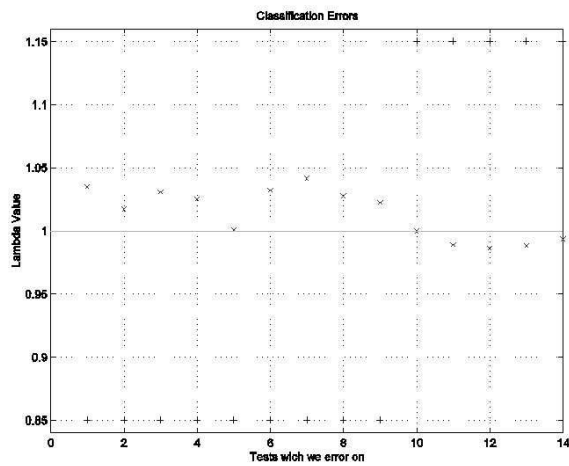
In figure 8, we can see that there are 14 errors on 200 testing samples.

### 5. Experimental Results

In this section we will show our results in testing the VLS, with a common fuzzy controller, on the real 125cc engine.

#### 5.1. Testing of the Virtual Lambda Sensor: Results

The Virtual Lambda Sensor was tested by Simulink on experimental data acquired in laboratory and by dSPACE in real-time application on the 125cc mono-cylinder gasoline engine.



**Figure 8. The classification error obtained using the classifier model.**

Mean Error in linear forecast	Max Error in linear forecast	Mean Error in ON-OFF forecast	Max settling time at 4600 rpm
1.18 %	4.5%	5%	250 ms

**Table 1. Experimental test results for the VLS on real engine**

The obtained results are collected in the tables 1 and 2:

Our *on-off* Virtual Lambda Sensor is able to forecast without error the *lean* mixture condition for Lambda value greater than 1.03 and the *rich* mixture condition for Lambda value less than 0.97. The virtual sensor shows a classification error of 7% for  $0.97 \leq \lambda \leq 0.99$  and of 9% for  $1.01 \leq \lambda \leq 1.03$  (see next table for a summary of these results). This means that all errors are committed at the *stoichiometric* condition ( $\lambda = 1$ ) with a bit polarization in the *lean* mixture condition zone.

Class. Error for $\lambda \leq 0.97$	Class. Error for $0.97 \leq \lambda \leq 0.99$	Class. Error for $1.01 \leq \lambda \leq 1.03$	Class. Error for $\lambda > 1.03$
0%	7%	9%	0%

**Table 2. More detailed analysis of error distribution for the *on-off* VLS**

	rich mixture	lean mixture
mean	45.74	37.41
variance	3.43	4.42

**Table 3. Statistic analysis of pressure signals used as inputs for the VLS**

The above described phenomenon is due to two kind of factors:

- non uniformity of measured data
- greater variability of pressure cycle in the *lean* mixture condition

By our analysis we can state that the mean value of  $P_{max}$  in *lean* condition is lesser than *rich* condition while the variance is greater. This last consideration means that in *lean* condition the signal is more instable. The table 3 shows the numerical results related to this analysis.

## 5.2. Testing of the Complete Control System: Results

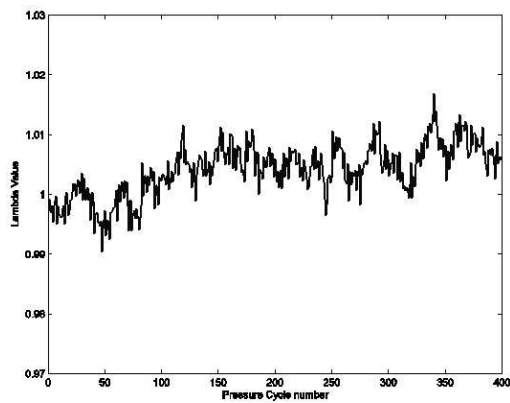
The Lambda Control System is a closed loop control system based on soft computing models. It is composed by a Virtual Lambda Sensor and by a Lambda Controller.

We tested real-time the system on the 125cc engine. At first, we tested the Lambda control system in the same engine conditions used to train our soft computing models (4600 rpm, WOT (Wide Open Throttle) condition, 28 Celsius degree of the intake fold). The figure 9 describes the trend of Lambda values, measured by the laboratory real Lambda sensor on the real engine, when our control system was activated. As we can see, engine is maintained close to the stoichiometric condition with an error within 1%. Finally, we tested our control system in several transient conditions. After few seconds, the control system is able to bring the engine in stoichiometric conditions with an error within 1%. The figure 10 shows the result of one of these tests on the real engine in transient conditions.

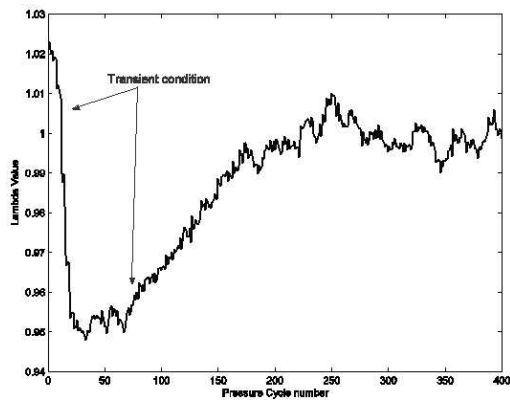
## 6. Conclusions and Forecast Objective

In this work, we showed how is possible to use a system with precision constraints to realize either linear or *on-off* Virtual Lambda Sensor. The results are encouraging even if some optimizations are possible in different directions.

The first future improvement could be to find an experimental data statistical analysis technique in order to identify the optimal data distribution to obtain a FIS with a maximum forecasting capability. Another future improvement could be to develop a classifier with more classes in order to



**Figure 9. Trend of real Lambda values after that we activated the control system; steady state with 4600 rpm, WOT condition.**



**Figure 10. Trend of real Lambda values after that we activated the SST control system; transient condition at 4600 rpm and WOT condition.**

obtain a more detailed information on Lambda values forecasting. A last step regards the implementation of several FIS, with a minimum overlap, in order to optimize the forecast capability of our system. This technique, used in software engineering and for safety critical systems [9], could allow us to yield systems more robust and with low sensibility to noisy input data.

## 7. Acknowledgements

We want to thanks our colleagues Ferdinando Tagliatalata, Francesco Carpentieri, Pasquale Cassese, for the help given us in laboratory activity on the engine. We, also, want to thanks the referee whom proposed

important corrections to our work. We apologize for the absence of more detailed block diagrams of our Virtual Lambda Sensor and the entire control system, but these systems are under patent approval.

## References

- [1] T.W. Long C.M. Atkinson and E.L. Hanzevack. Virtual sensing: a neural-network-based intelligent performance and emissions prediction system for on-board diagnostics and engine control. In *Proceedings of the 1998 SAE International Congress & Exposition, vol. 1357, 208*, pages 39–51, 1998.
- [2] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132, 1985.
- [3] M. Sugeno and G.T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28:15–33, 1988.
- [4] A. N. Tikhonov. On solving incorrecly posed problems and method of regularization. *Doklady Akademii Nauk USSR*, 151:501–504, 1963.
- [5] A. N. Tikhonov. On regularization of ill-posed problems. *Doklady Akademii Nauk USSR*, 153:49–52, 1973.
- [6] A. N. Tikhonov and V.Y. Arsenin. *Solution of Ill-posed Problems*. W.H. Winston, Washington, DC, 1977.
- [7] M. Di Meglio and N. Cesario. Realization and performance analysis of m-dimensional data clustering algorithms. *IASTED 2005 Reviewing*.
- [8] Jyh-Shing Roger Jang. Anfis: Adaptive-network-based-fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685, May/june 1993.
- [9] Amanda J. C. Sharkey, Noel E. Sharkey, Uwe Gerecke, and G. O. Chandroth. The “test and select” approach to ensemble combination. *Lecture Notes in Computer Science*, 1857, 2000.