Proceedings of the Institution of Mechanical Engineers, Part A:
Journal of Power and Energy, 2003, Vol. 217, No A6, pages 605-614

Optimisation of Combustor Wall Heat Transfer and Pollutant Emissions for Preliminary Design Using Evolutionary Techniques

J.M. Rogero P. A. Rubini*

Department of Aerospace Sciences,

School of Engineering

Cranfield University, MK43 0AL, Bedfordshire, UK

Abstract

This paper presents the concept and application of a design optimisation toolbox, based upon evolutionary techniques, for the preliminary design of a gas turbine combustor. The toolbox has been designed to interface with existing analysis packages and to perform optimisation in parallel over a heterogeneous network of workstations. The optimisation capabilities of the toolbox are demonstrated for gas turbine combustor design by automatically attaining twenty-two performance targets in a combustor design whilst performing minimisation of wall cooling flow and NOx emissions.

1 Introduction

Over recent decades the use of computational methods in engineering design has dramatically increased, indeed there is some similarity with the advent of mechanisation in the manufacturing industry over two centuries ago. However, designers are now frequently presented with an increasing number of complex decisions to be made during

*Corresponding author

the early stages of the design process especially as the desired performance targets become ever more difficult to achieve.

In order to optimise the design process, many of these decisions can be automated by integrating numerical simulation and performance analysis with a software optimisation tool. The objective being to reduce lead times and costs whilst increasing the perceived and measured performance of the design [**1**].

1.1 Status of design optimisation in the industry

The procedure in a conventional manual design process typically consists of taking an existing design and further developing, by appropriate scaling and slight modifications, to meet the new requirements. Optimisation is achieved through past experience (rule of thumb) and trial and error, optionally employing user driven computational analysis programs iteratively until a suitable solution is found. This is particularly the case for the preliminary design of gas turbine combustors, where new designs are mostly based upon previous proven designs, the experience of the engineer and a costly trial process.

The majority of complex engineering products incorporate a significant number of strongly correlated design variables, each with conflicting performance targets. This presents a challenging problem if an optimal solution is to be sought. The application of the traditional iterative process can be inefficient, resulting either in a lengthy and costly design phase or a compromised optimal result if time and resources are constrained. Optimisation algorithms offer the potential to improve the overall efficiency of the design process by reducing the time spent manually iterating towards a suitable optimised design. In this spirit a number of research papers have recently discussed the use of optimisation techniques for engineering design, for example [**2-4**].

The Microprocessor industry has already started the transition towards automation of the design process [**5**], however many others industries continue to lag behind in this field. A recent survey of British industries [**6**] highlighted the fact that optimisation algorithms are not yet widely used in the engineering design process. Several inhibiting factors were identified as being responsible for this limited usage:

- Lack of integration of existing optimisation tools
- Limited optimisation skill among design engineers
- The computational cost of simulation
- The designer wants control over the optimisation
- The complexity of real life optimisation problems

These inhibitors need to be overcome for optimisation to be accepted as a routine tool in practical engineering applications.

1.2 Gas turbine combustor preliminary design

1.2.1 Background

The preliminary design of a gas turbine combustor largely concentrates upon determining the core features of the combustor such that the capacity and performance requirements of the engine are achieved. The required steps for the preliminary design of a combustor can be simplified as shown in Figure 1.

The performances targets for combustor design usually include: wall temperature, ignition and altitude relight limits, combustion efficiency, exhaust emissions, pressure drop, and exit temperature traverse [**7**]. The design process consists of selecting the type of combustor, sizing it and finally tuning the design parameters by iteratively performing simulations and modifications until all of the performance targets are

satisfactory. Such simulations are mostly based on semi-empirical mathematical models and correlations allowing rapid simulation during this first stage of the design process. More detailed CFD analysis, small scale rig tests and larger scale sector tests typically follow to validate and refine the final design before manufacture of a full size engine demonstrator.

1.2.2 The need for improvement

Preliminary design is a demanding and time consuming processes relying upon considerable past experience, empirical design rules, and trial and error. The process is becoming increasingly complex as pressure is exerted on the one hand by both regulations [**8**] and customers requiring increased performance, and on the other hand companies desiring a reduction in design costs and cycle time to remain competitive.

The use of optimisation algorithms has the potential to reduce the overall costs of the preliminary design process whilst achieving the desired optimal design [**9-11**].

The technique demonstrated in this paper is a new and unique approach for gas turbine combustors, representing the first step towards autonomous engineering design with user-specified characteristics and objectives. This novel concept is based on genetic algorithms (GA) supported by an appropriate set of analysis [**12**].

A graphically based engineering optimisation toolbox has been developed, to ease the preliminary design of gas turbine combustors, grouping in a modular framework the necessary set of analysis and optimisation tools.

2 Gas Turbine Combustor Preliminary Design Optimisation Toolbox

2.1 Problem approach

The original concept of the Toolbox was to provide a set of methods and tools suitable for general engineering optimisation with large numbers of both targets and constraints, whilst employing the original analysis software to evaluate the solutions. To be useful the tool was required to be as efficient as possible and achieve results in the minimum time necessary. It was also required to be sufficiently flexible to permit improvements of existing methods and the addition of new tools. Finally to be used in real life the Toolbox was required to be user friendly, allowing the designer to interact with the optimisation process without requiring detailed training in the optimisation domain.

In order to achieve these objectives a modular object oriented architecture was selected, that permitted additional tools to be easily incorporated in the future. In order to facilitate the interface between legacy software and the optimisation suite a generic interface module was identified as being critical to the overall utility of the Toolbox. Finally, in order to be applied to practical engineering problems the optimisation tool was required to be robust and efficient over a wide range of problems.

Genetic algorithms were selected in this project as the principle optimisation tool because of their robustness when optimising complex objective functions. However it was found that a number of enhancements to the so-called “Simple Genetic Algorithm” were necessary to achieve efficient and reliable optimisation of engineering problems described using floating point real numbers. In addition, to reduce the computational overhead of simulation it was found necessary to design the Toolbox to accommodate distributing the evaluation of the analysis code over a network of heterogeneous computers.

The final component of the toolbox is related to the program/user interaction. The addition of a user friendly graphical interface, where the designer can follow the evolution of all the problem parameters during the optimisation, was found to be essential to allow the user to get a 'feeling' of the optimisation process without requiring a detailed knowledge of optimisation process itself.

2.2 Modular design

A significant degree of modularity was required in order to meet the desired aims of versatility and future development of the toolbox. This was achieved by the use of Java as the main programming language, being object oriented it readily allowed development of the required modular architecture. In addition, the platform independence of Java is a significant advantage when working on a heterogeneous set of computers especially the advanced support for networking and graphics.

There are some disadvantages which were considered, Java suffers from being relatively slow compared to fully compile languages such as C/C++ and in some instances is noticeably less memory efficient. The Toolbox was required to be used for engineering design where simulation is performed by one or more external simulation codes. Since the simulation processes themselves take a very high proportion of the computational time the relative slowness of Java was not considered to be a significant handicap. The capacity of modern computer systems is such that for the simulation codes employed there was no significant constraint on memory usage.

The object-oriented design of the Toolbox was such that modules could be loaded at runtime depending on the specific optimisation requirements. The individual modules were organised to allow alternative optimisation algorithms or by providing functionality such as for example, interfacing with other software and distributing the

evaluation over a network. The optimisation modules were themselves composed of interchangeable sub-modules. These implemented the basic functions of the optimiser. Figure 2 shows an example of the Toolbox modular organisation.

2.3 Adaptation to engineering design

In order to overcome the inhibitors mentioned in section 1.1, the optimisation Toolbox was carefully designed to accommodate the practical engineering design process.

Robustness was considered to be of particular importance for the optimisation algorithm, hence the adoption of a method based on genetic algorithms [**13**] modified for engineering design. See section 2.3.1.

One of the key factors defining the usability of the Toolbox was the capability to integrate with existing legacy software. This was achieved through an interface module as described in section 2.3.2.

The overall time taken to carry out a full optimisation is another key factor particularly for engineering simulations which are often very costly in terms of CPU time. For this reason a distributed evaluation module was incorporated. See section 2.3.3.

The final factor is the ability of the engineer to control the optimisation without in depth knowledge of optimisation techniques themselves, whilst describing the complexity of the objective function and the target parameters defining the engineering design. This was approached by developing a system able accommodate a large number of performance parameters, for which the user could define targets, validity range, and the requirement to minimise or maximise any particular target parameter [**14**].

2.3.1 The optimisation algorithm

This subsection describes the implementation of the Genetic Algorithm (GA). Genetic Algorithms, first developed by Holland [**15**], provide a robust global optimisation method while being able to handle high-dimensional problems. For these reasons GA's posses significant potential for design [**11**] and have been successfully applied to many engineering problems [**16-18**].

The GA optimiser module was based on SGAJ from Hartely [**19**], a Java implementation of the "simple GA" (SGA) from Goldberg [**13**]. The original model was extensively modified to both adapt it to engineering design problems and maximise performance.

Traditional implementations of SGA use binary encoding to describe the optimisation parameters. However the work of Janikow & Michalewicz [**20**] suggests that coding the parameters using real numbers can improve the optimisation performance, especially for high dimensional problems. The chromosome modules were therefore modified to support real numbered parameter encoding.

The performance of real coded GA can be further improved by using the linear crossover operator introduced by Davis [**21**]. A module implementing this linear crossover was implemented.

The adaptation of the mutation operator to real numbers was performed by implementing the decayed creep mutation operator. This operator is based on the real number creep mutation from Davis [**21**]. As implemented, the operator adds a decay rate. The decay reduces the mutation range as the GA population ages resulting in a broad capability to explore during the initial stages of the optimisation and fine local searches in the later stages.

For engineering optimisation the evaluation of the fitness function requires calls to one or more analysis codes, which can be significantly expensive. In order to reduce the numbers of calls to the evaluation functions Davis [**21**] suggested preventing the creation of duplicate genes. Such a duplicate prevention method was implemented in the selection module. In an effort to further reduce the number of evaluations Steady State replacement without duplicates was implemented as well. The module was extended to prevent the re-evaluation of genes by using a database of all the genes generated during the optimisation process.

A random search phase was introduced at the beginning of the optimisation process to perform a broad exploration of the problem domain. This was found to improve the quality of the initial population and, for design problems limited by constraints within the simulation code, such as valid ranges of empirical data, ensured that only feasible designs were selected for the initial population after the random search.

2.3.2 Interfacing with simulation software

One of the main requirements of the Toolbox was the capacity to interface with a variety of existing legacy software, in order to enable the Toolbox to be applied to a range of different problems. Another requirement was ease of use, requiring the user to program a specific interface to their analysis code was deemed to be unacceptable.

A common feature of many legacy codes is the use of text input / output files. It was decided to capitalise upon this feature and communicate through text files rather than direct socket or inter-process communication. Where the analysis code is CPU intensive the resultant I/O was found to be relatively insignificant. The most robust way to locate the relevant data within an I/O text file was found to be the use of regular expressions [**22**] combined with tokenisation of the file. This was implemented by searching for data

via a keyword regular expression within the file or from its position (line/column) or any combination of the two. Experience demonstrates this to be a very robust way to access data for both read and write.

The interface module only requires the user to input the data location in the problem configuration file. For example data retrieval could be of the type: in the "node" section of the result file search for node "456" which is on the first column and read the data on the second next column.

2.3.3 Distributed computing

A very efficient way to reduce the computational overhead of recurrent runs of CPU intensive simulation software is to perform the execution of the applications within a Heterogeneous Distributed Computing [**23**] environment. The principle of this approach is to distribute the execution of individual processes over a network of computers which might not all be of the same architecture.

In the case of the distribution of multiple evaluations of the fitness function through the execution of a sequential analysis code, communication only takes place between the GA and the analysis software. This communication is performed through text files rather than inter-process communication alleviating the need to use potentially machine dependent communication libraries.

2.3.4 Control of the performance parameters

One of the principal difficulties when optimising practical engineering designs is the large number of parameters that must be accommodated. This is especially true for combustor preliminary design, where between 20 and 50 performance parameters are typically involved. In addition it is also necessary to have a straightforward way for the designer to interact with those parameters.

A unique method was adopted whereby the designer can define for each parameter a target to be attained, a range this parameter should stay within, and the requirement to maximise or minimise the parameter. The quality of the design is then determined from achievement of the targets, the possible violation of ranges, and the optimisation of the selected parameters.

This approach enabled the designer to have total control over the optimisation procedure without having to know too much about the detailed algorithms and without having to devise a fitness function himself.

2.4 Modelling

The gas turbine combustor simulations were obtained using the flow simulation code Flownet. This code has been developed by Stuttaford and Rubini [**24**]. It is capable of modelling geometrical features of a gas turbine combustor and to return information regarding temperature, pressure, mass flow-splits, fuel air ratios, and flow velocities, in different regions of the combustor.

Flownet is a network solver based on a semi-empirical method. The combustor is divided into a series of one-dimensional sub-flows containing independent semi-empirical governing equations. These equations are selected depending on the geometrical feature modelled. An overall governing equation links each sub flow to obtain a complete solution. This enables solutions to be obtained very rapidly in comparison to CFD. A detailed explanation of the network algorithm and Flownet can be found in [**25**].

In order to give some information to the optimiser about the NOx emission of the combustor, a simple NO production model based on the Zeldovich [**26**] mechanism was incorporated. The model calculates the NO production rate from the equilibrium

concentration of O2 and N2 using mean zone temperatures and zone residence times. Total NO is obtained by summing all elements along the flow path. It is important to note that this is only a basic model, of a complexity somewhere between that of an empirical correlation and a stirred reactor [**27**]. It is not quantitatively accurate but should be suitable to demonstrate the trend of the NOx production depending upon modification of the combustor parameters. This model is described in detail in [**14**].

3 Application of the optimisation toolbox

In order to demonstrate the design automation and optimisation capabilities, the optimisation toolbox was applied to the preliminary design of a conventional 1970's combustor design.

3.1 Preliminary design of a single annular combustor

An existing single annular aero-engine combustor was selected as a test case. A model of this combustor was made for the Flownet simulation code, see Figure 3. In this combustor flame tube cooling is achieved through effusion ports and Z-rings and fuel/air mixing is achieved by two sets of dilution ports. The geometric characteristics or the features of the combustor were encoded into twenty-five real-numbered chromosomes. The alternative combustor networks generated by the optimiser were evaluated using the network simulation code.

3.1.1 The optimisation problem

The quality of a combustor design is defined by a number of performance parameters. Ideally these performance parameters should be: the overall pressure drop, the pollutant emission, the relight altitude, and the exit temperature profile with a constraint on the

maximum wall temperature. In order to accommodate these additional design parameters, extra constraints were added to provide an indirect control

Mixing, which affects pollutant emissions and the exit temperature traverse, was constrained by limiting the overall, and flame tube wall, pressure loss, as well as the port cross flow Mach number. The relight altitude was controlled by an empirical relight loading parameter.

Overall this resulted in 22 performance targets to be achieved. With these defined two alternative strategies can be adopted. Firstly to use the optimisation Toolbox as an automatic design tool to achieve predefined values of the performance parameters. This is demonstrated in section 3.12. Secondly, to search for an optimal solution, by allowing selected parameters to float within a defined range, whilst constraining other parameters to predefined targets. This is demonstrated in sections 3.1.3 and 3.1.4.

To perform these tasks the Toolbox was configured to modify the dimensions of the combustor cooling and dilution ports, which define the flow within the combustor. A series of genes were defined to encode and control the size of the ports, effusion patches, Z-rings, fuel injector, and the base plate cooling orifices. The profile of the combustor flametube was not allowed to be modified.

3.1.2 Designing for targets

The first task required an original manual design to be reproduced based upon identical performances targets. For all of the 22 performance parameters an allowable range was set to ensure the validity of the design. This range was ±5% for the critical parameters and ±10% for the less critical parameters. The effects of reducing these tolerances was not investigated, though it is reasonable to assume that if too tight a constraint is imposed, no optimal solution might be found.

The optimisation was performed in just over four hours using a single workstation for 10000 evaluations. All the targets were within the allowable range after 2700 evaluations.

The results of the optimisation are shown in Table 1. From this table it is possible to see that all the critical parameters were achieved with a high precision and that all the targets parameters were achieved with less than 3.5% error.

3.1.3 Designing for minimum cooling flow

The second task required the optimiser to achieve the same targets as in section 3.1.2 but with the added goal of reducing the mass flow of air used to cool the flametube walls.

The optimisation was again performed in just over 4 hours using a single workstation for 10000 evaluations. All of the targets were within their allowable range after 2500 evaluations. The cooling flow was reduced by 23.2% with respect to the baseline design.

The results of the optimisation are shown in Table 1. From this table it is possible to see that all of the parameters were achieved within their allowable range of error. Figure 4 shows the change in the cooling flow as the optimisation evolves. It can be seen that no optimisation actually occurs until all the targets are within their allowable range. Figure 5 contrasts the final mass flows through individual cooling devices for the baseline design and for the optimised design.

3.1.4 Designing for minimum NOx

The final task was to demonstrate the capability of the optimiser to optimise the design of the combustor for NOx emissions.

The optimisation was performed in approximately 30 minutes using 10 networked workstations for a total of 10000 evaluations. All of the targets were within their allowable ranges after 2000 evaluations. The optimiser managed to reduce the predicted emissions by 18.6%. The results of the optimisation are shown in Table 1. Again, all the targets were achieved within the allowed error band.

Figure 6 and 7 illustrate the change in predicted NOx emissions and empirical altitude relight factor with the evolution of the optimisation. It is interesting to note that the reduction of NOx ceases when the relight factor reaches the limit of its range. This demonstrates that altitude relight, which is dependent on the combustor mass flow, is a limiting design parameter.

Figure 8 presents the air/fuel ratio (AFR) along the flametube for the baseline design and the optimised design. The figure clearly illustrates how the optimiser has selected a leaner primary zone.

Figures 9, 10 and 11 illustrate the evolution of the individual air/fuel ratios (AFR) of respectively, the injector, intermediate and dilution zones of the combustor. It is again evident that the optimiser selects a lean air/fuel ratio to achieve low NOx emissions.

4 Conclusion

The automation and the optimisation of a gas turbine combustor preliminary design using the Toolbox has demonstrated the potential for this type of software tool to be applied to real life problems involving complex objective functions and large numbers of conflicting performance parameters.

The use of an optimisation algorithm based upon an evolutionary genetic algorithm successfully improved the predicted performance of a typical gas turbine combustor during the preliminary design phase. Furthermore the time taken to achieve these

results, in the order of a few hours, was significantly less than that for an equivalent manual design.

Acknowledgements

The authors wish to acknowledge the grateful support of the Defence Evaluation and Research Agency (DERA) and Rolls Royce plc.

References

1 Rogero, J.M. et al. Applications of evolutionary algorithms for solving real life design optimisation problems. Sixth conference on parallel problem solving from nature, workshop proceedings, September 2000, pp. 85-87.

2 P.M. Pardalos, P.M. et al. Recent developments and trends in global optimisation, Journal of computational and applied Mathematics, Vol. 124, (pp209-228), 2000.

3 Sobieszczanski-Sobieski, J., and Haftka, R.T. Multidisciplinary aerospace design optimization: survey of recent developments. Structural Optimization 14. 1997, pp. 1-23, Springer-Verlag.

4 Bullock, G.N., et al. Developments in the use of genetic algorithm in engineering design, Design studies Vol. 16, 1995, pp507-524.

5 Johnson, W. M. Superscalar Microprocessor Design, Prentice-Hall, 1991, Englewood Cliffs, NJ.

6 Roy, R., et al. Design Optimisation: A Survey of British industries. Flexo Report 5, SIMS, Cranfield University, 2000.

7 Lefebvre, A.H. Gas turbine combustion second ed., Taylor & Francis, 1999, ISBN 1-56032-673-5.

8 Le Dilosquer, M. The aero engine response to the protection of the global atmosphere, IMechE seminar on Gas Turbine Pollutant emissions Technology Advances and Updates, 1999.

9 Deb, K., and Goyal, M. Optimising engineering designs using a combined genetic search. Proceedings of the Seventh international conference on genetic algorithms, Morgan Kaufman, 1997, pp. 521-528.

10 Esping, B. Design Optimisation as an engineering tool. Structural Optimisation 10, Springer-Verlag, 1995, pp. 137-152.

11 Parmee, I.C. Exploring the design potential of evolutionary / adaptive search and other computational intelligence technologies. Adaptive Computing in Design and Manufacturing, Springer-Verlag, April 1998, pp. 27-44.

12 Rogero, J.M., and Rubini, P.A. A platform independent engineering optimisation tool based on genetic algorithms and distributed computing applied to gas turbine combustor preliminary design, computational engineering using metaphors from nature, B. Topping, Civil-Comp Press, 2000, ISBN 0-948749-66-0, pp143-149.

13 Goldberg, D.E. Genetic algorithms, in search of optimisation & machine learning, Addison-Wesley Publishing, 1989.

14 Rogero, J.M. Preliminary design of gas turbine combustor using genetic algorithms, PhD thesis, Cranfield University UK, to be submitted 2002.

15 Holland, J.H. Adaptation in natural and artificial systems, Ann Arbor, University of Michigan Press, 1975.

16 Makinen, R.A.E., et al. Multidisciplinary shape optimisation in aerodynamics and electromagnetic using genetic algorithms, International Journal For Numerical Methods In Fluids 30, 1999, pp149-159.

17 Trulove, A.M., and Whitaker, K.W. Rocket Stage Optimization Using a simple genetic algorithm. 29 Joint Propulsion Conference and Exhibit, Monterey, CA, June 1993, AIAA-93-1778.

18 Tappeta, R.V. et al. A multidisciplinary design optimisation approach for high temperature aircraft engine components. Structural optimisation, Vol. 18, 1999, pp 134-145.

19 Hartley, S.J. Concurrent programming the Java language. Oxford University press, 1998.

20 Janikow, G.C., and Michalewicz, Z. An experimental comparison of binary and floating point representations in genetic algorithms. Proceedings of the fourth international conference on genetic algorithms, Morgan Kaufman, 1991, pp. 31-36.

21 Davis, L. Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.

22 Knuth, D.E. The art of computer programming, fundamental Algorithms. 2nd ed., vol. 1, Addison-Wesley, 1973.

23 Sunderam, V.S., and Geist, G.A. Heterogeneous parallel and distributed computing. Parallel Computing 25, Elsevier, 1999, pp. 1699-1721.

24 Stuttaford, P.J., and Rubini,, P.A. Preliminary gas turbine combustor design using a network approach. ASME paper 96-GT-135, 1996.

25 Stuttaford, P.J. Preliminary gas turbine combustor design using a network approach, PhD Thesis School Of Mechanical Engineering Cranfield University, UK, 1997.

26 Zeldovich, J. Acta Physiochimica, Vol.21, 1946, pp 577.

27 Moss, J.B. Predictive methods for gas turbine combustor emissions. IMechE seminar on Gas Turbine Pollutant emissions, 1999.

	Design For Targets	Minimise cooling	Minimise NOx
Pressure drop comb	Range ±5% 0.27%	Range ±5% -0.49%	Range ±5% -0.39%
Pressure drop wall 1	Range ±5% 0.43%	Range ±5% -0.80%	Range ±5% -0.63%
Pressure drop wall 2	Range ±5% 0.41%	Range ±5% -0.75%	Range ±5% -0.60%
AFR injector	Range ±5% -2.82%	Range ±5% 3.69%	Range NA 12.80%
AFR 1	Range ±5% 0.56%	Range ±5% 0.67%	Range NA 4.82%
AFR 2	Range ±5% 0.90%	Range ±5% 2.00%	Range NA 4.16%
Flametube cooling	Range ±5% -1.90%	Range NA -23.28%	Range ±10% -1.67%
Base Plate Cooling	Range ±5% 1.42%	Range ±5% -4.97	Range ±5% 4.69%
Max Temp Combustor	Range +0% -0.02%	Range +0% -0.70%	Range +0% -0.10%
Max Temp Zone 1	Range +0% -3.00%	Range +0% -0.89%	Range +0% -1.95%
Max Temp Zone 2	Range +0% -0.02%	Range +0% -0.70%	Range +0% -0.10%
Avg Temp Combustor	Range +5% 0.15%	Range +5% 1.22%	Range +5% 1.08%
Avg Temp Zone 1	Range +5% -0.54%	Range +5% -0.73%	Range +5% -1.21%
Avg Temp Zone 2	Range +5% 0.75%	Range +5% 2.34%	Range +5% 2.39%
Recitculating flow	Range ±5% 0.31%	Range NA 1.84%	Range NA 4.97%
SI Loading	Range ±5% 0.03%	Range ±5% -0.05%	Range ±5% -0.04%
Relight loading factor	Range ±5% 0.29%	Range ±5% 1.82%	Range ±5% 4.99%
NOx	Range +0% -1.21%	Range +0% -5.97%	Range +0% -18.59%
Mach ratio Outer Ports 1	Range ±10% -1.89%	Range ±10% -5.36%	Range ±10% -1.06%
Mach ratio Inner Ports 1	Range ±10% 0.73%	Range ±10% -0.17%	Range ±10% 8.10%
Mach ratio Outer Ports 2	Range ±10% -0.93%	Range ±10% 1.17%	Range ±10% 3.91%
Mach ratio Inner Ports 2	Range ±10% 3.33%	Range ±10% -2.02%	Range ±10% 7.21%

Table 1: Allowable range and target achievement of the three design optimisation cases

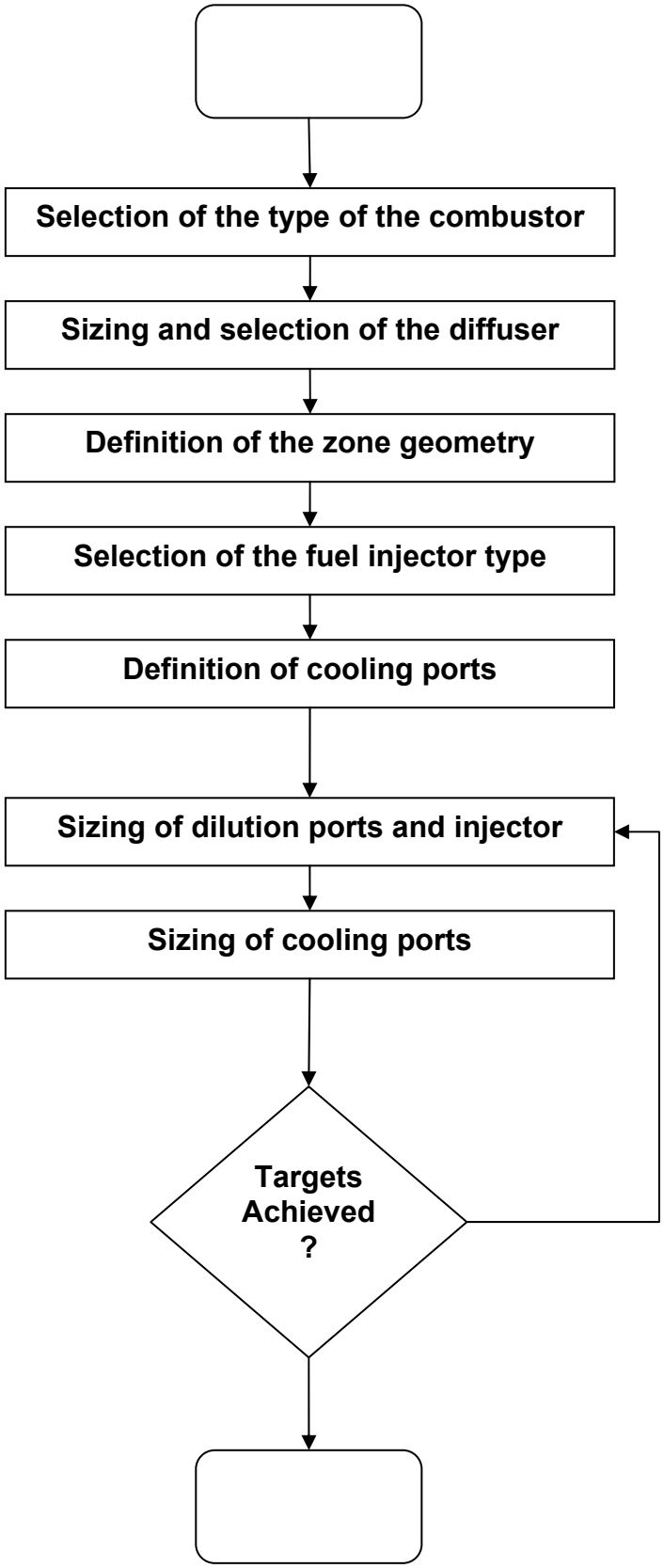

Figure 1: Combustor design flow chart.

Figure 2: Optimisation ToolBox modular design.

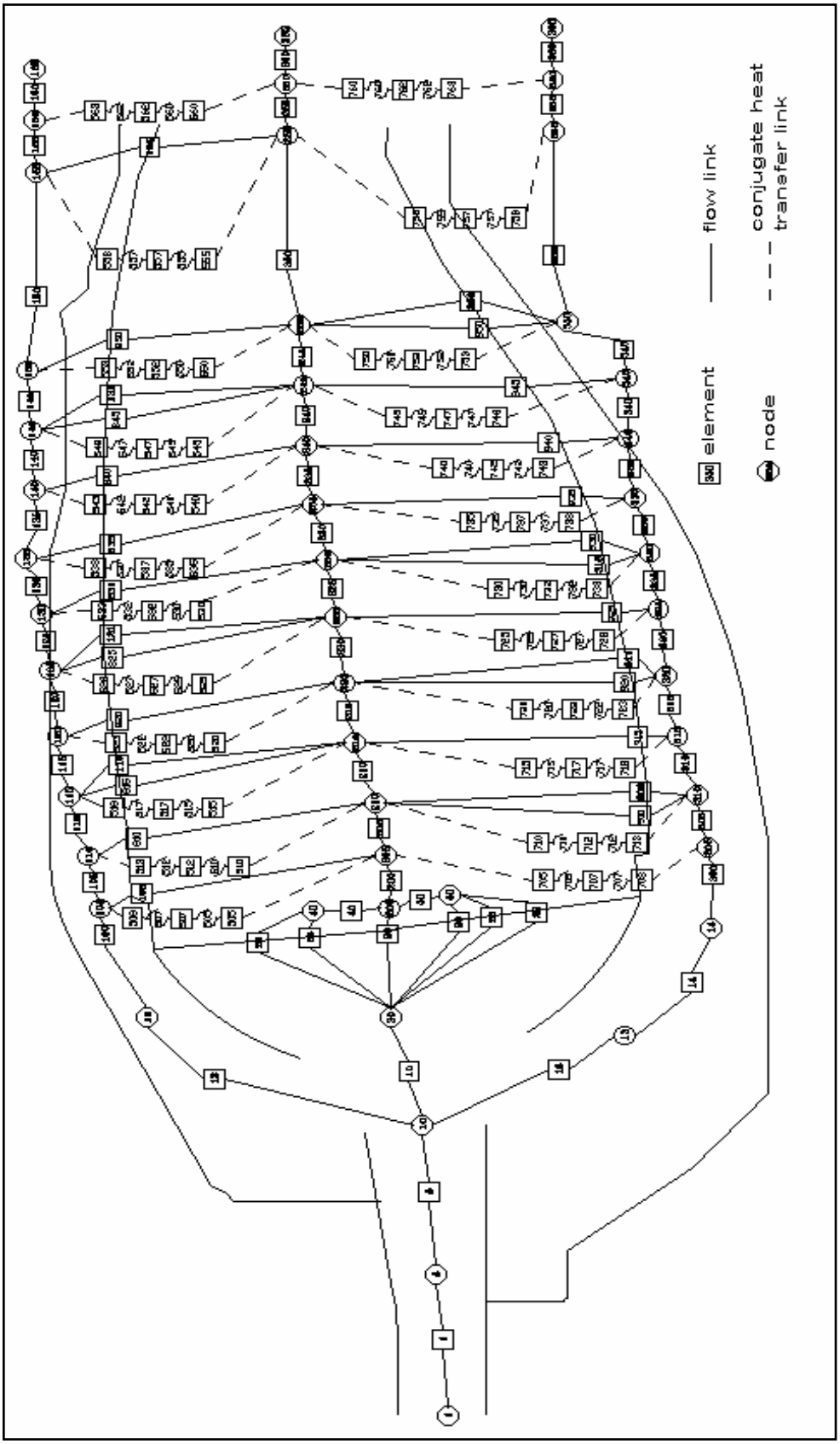

Figure 3: Combustor Network Model.

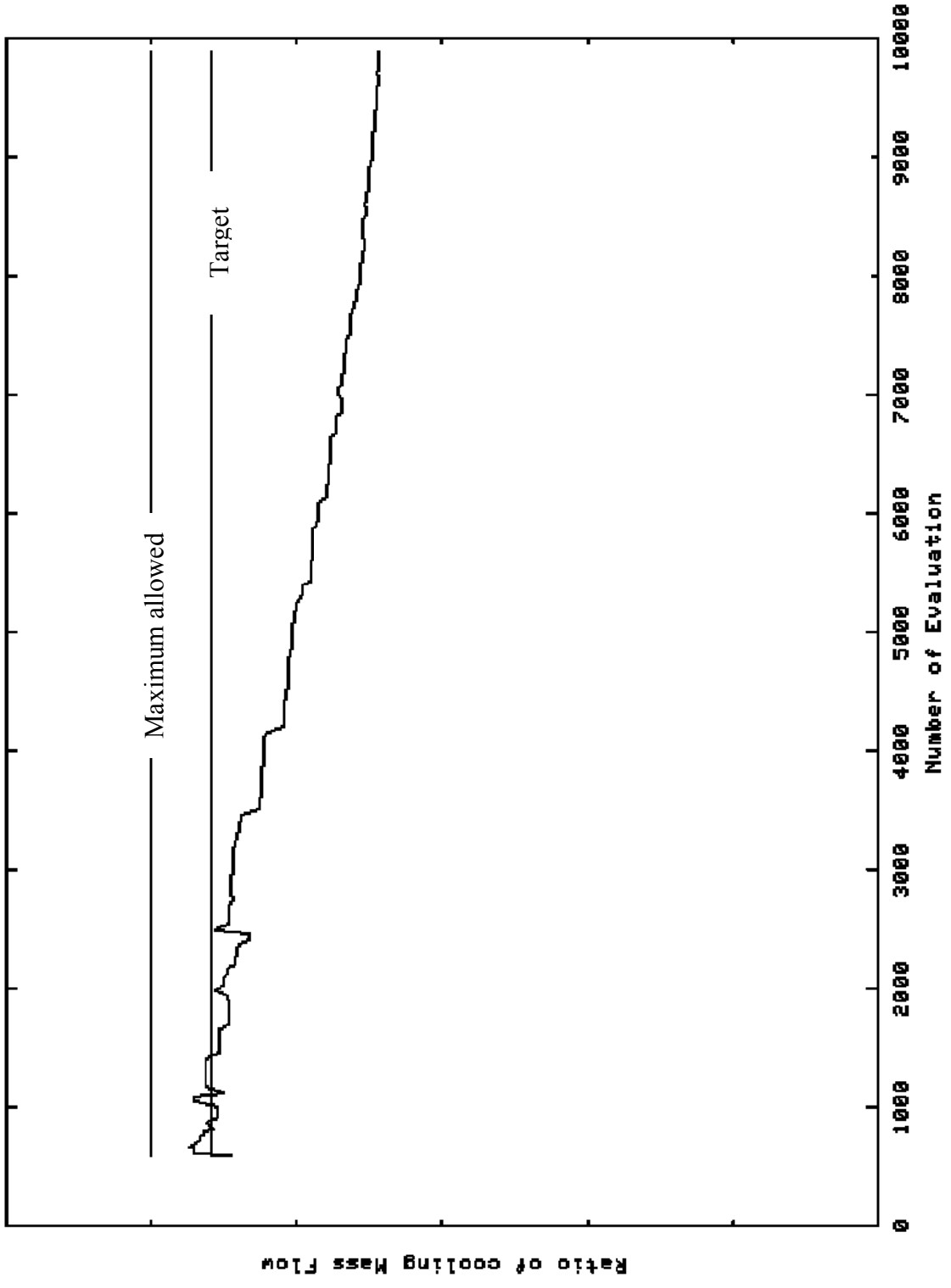

Figure 4: Evolution of the cooling flow ratio.

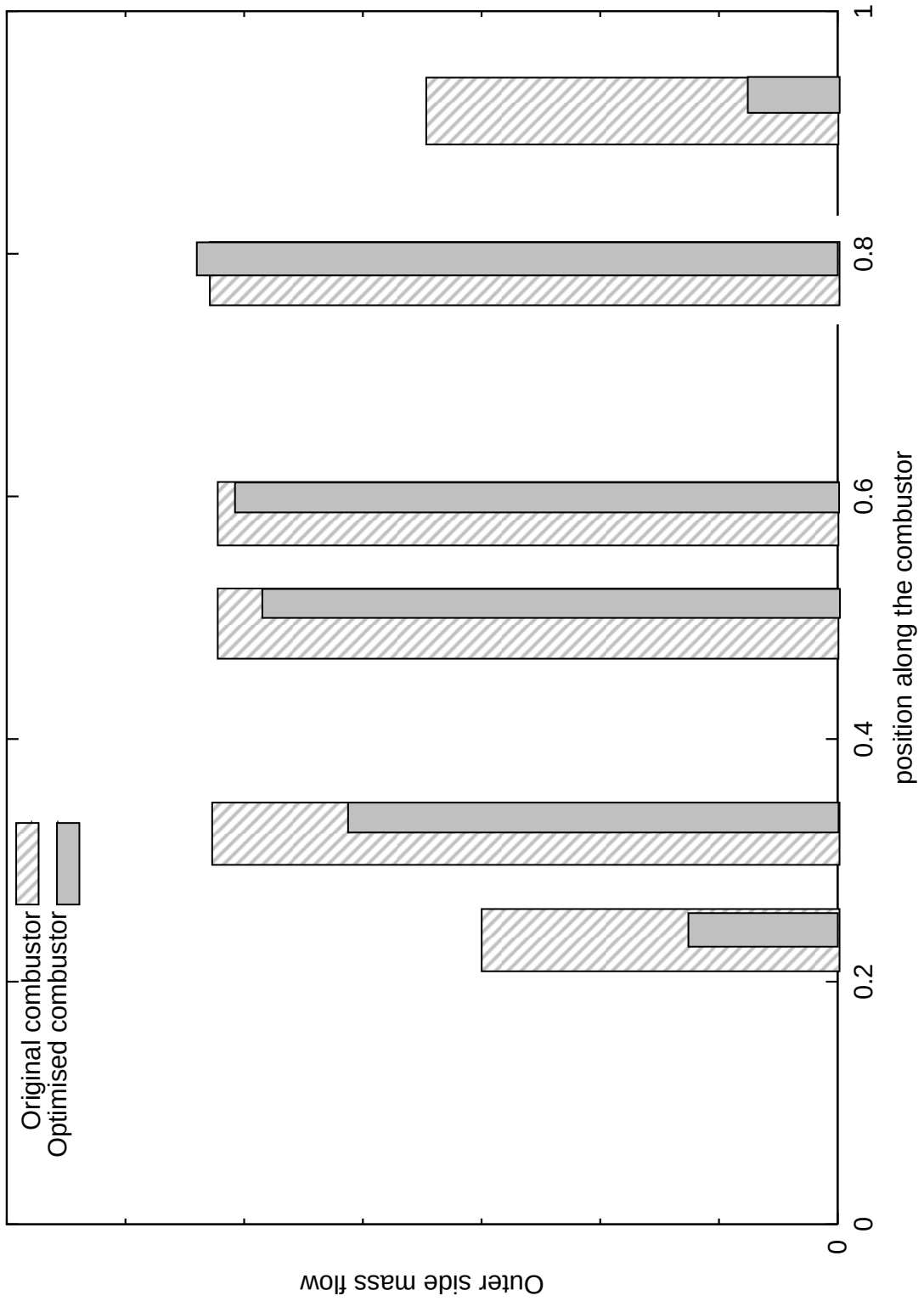

Figure 5: Comparison of wall cooling flow.

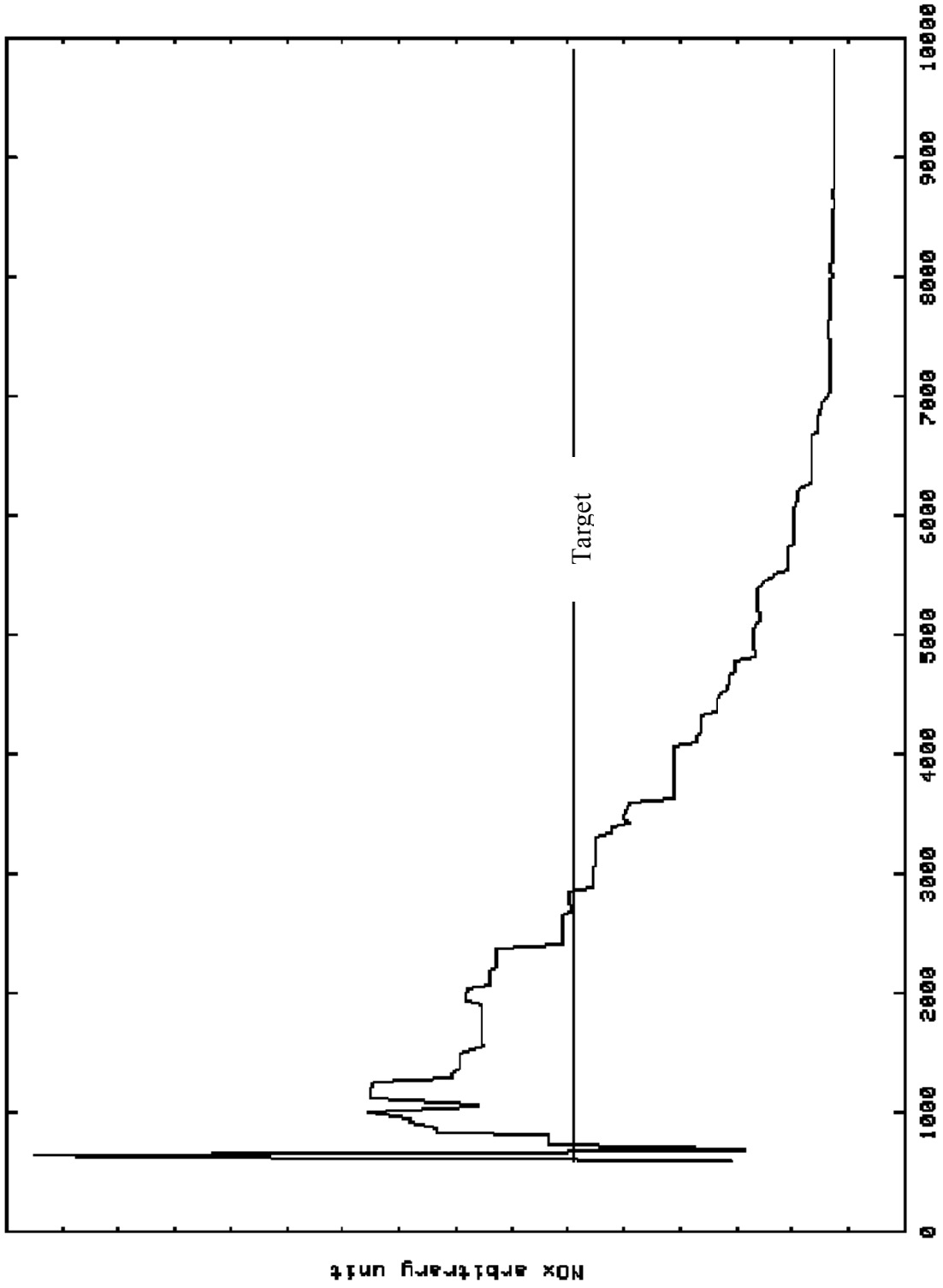

Figure 6: Evolution of the predicted NOx.

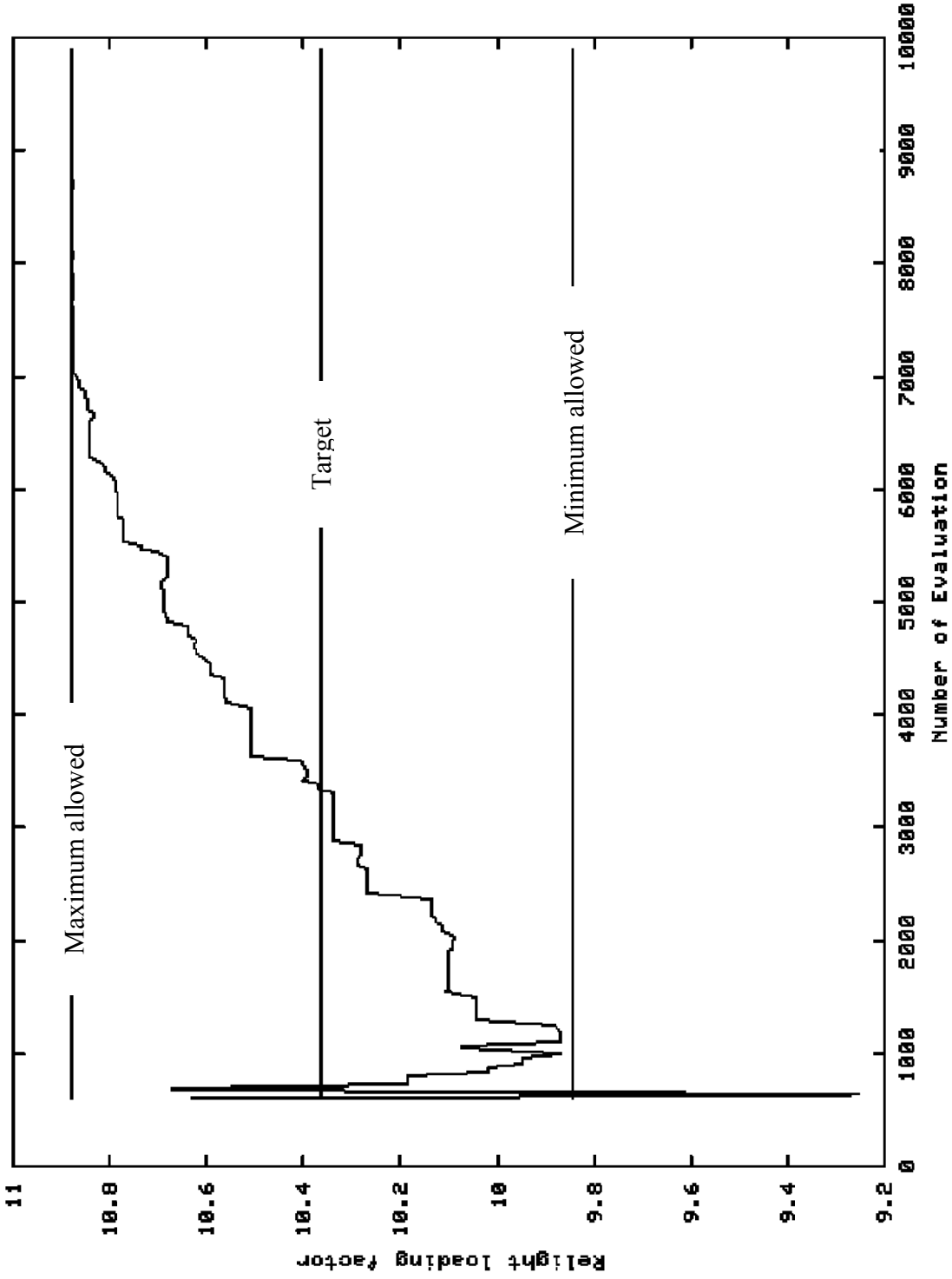

Figure 7: Evolution of the relight loading factor.

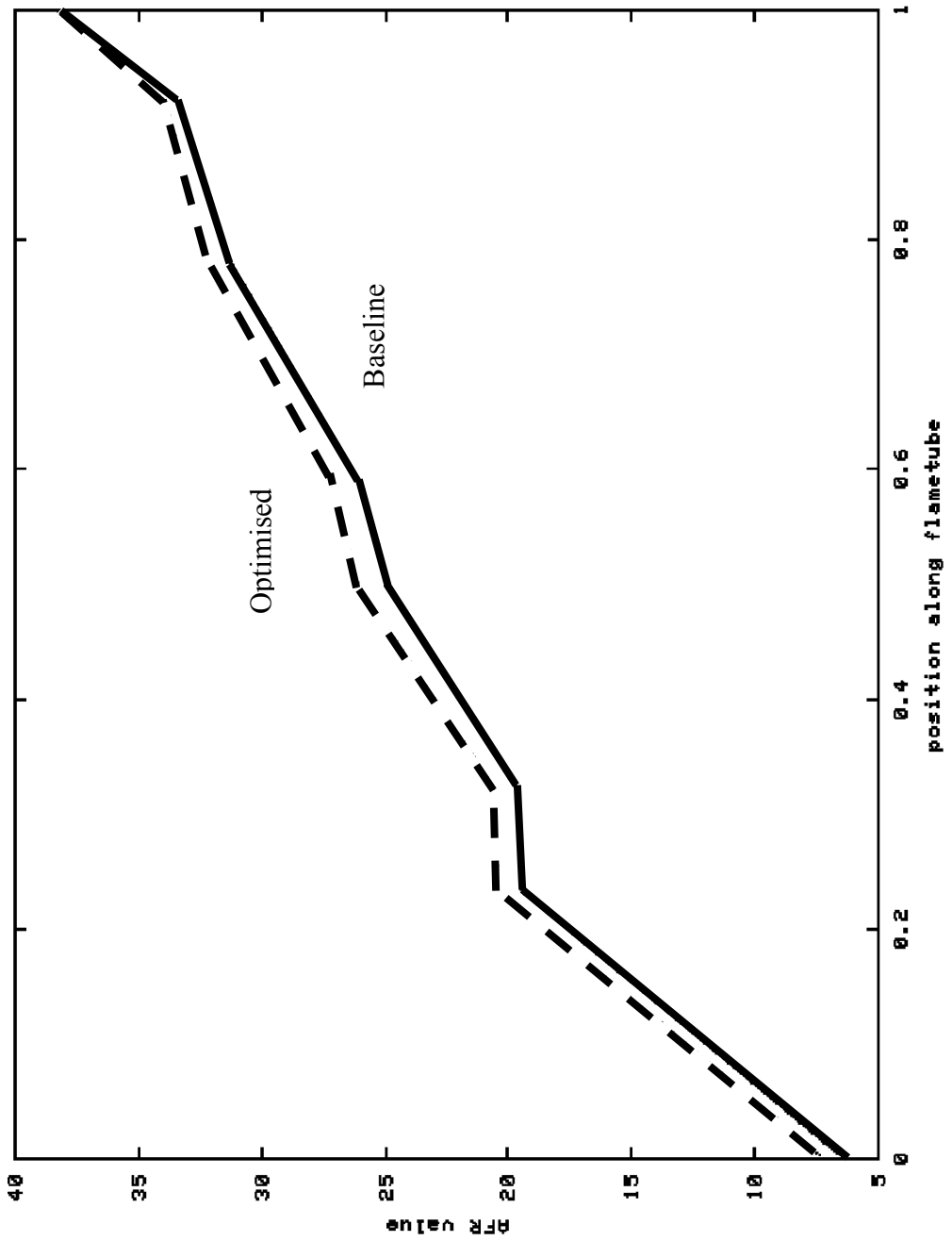

Figure 8: Comparison of air/fuel ratio along the axis of the Combustor

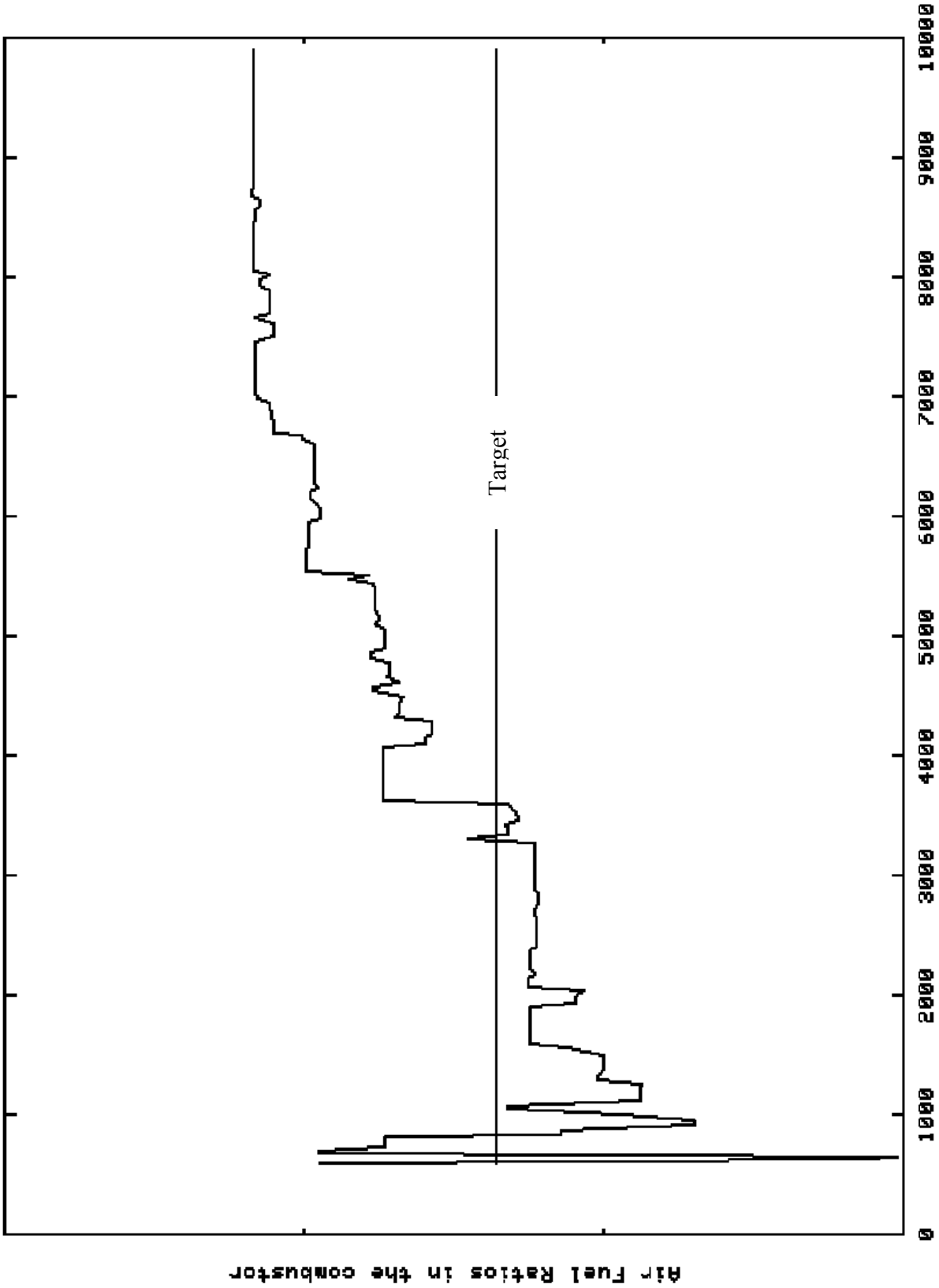

Figure 9: Evolution of the injector air/fuel ratio

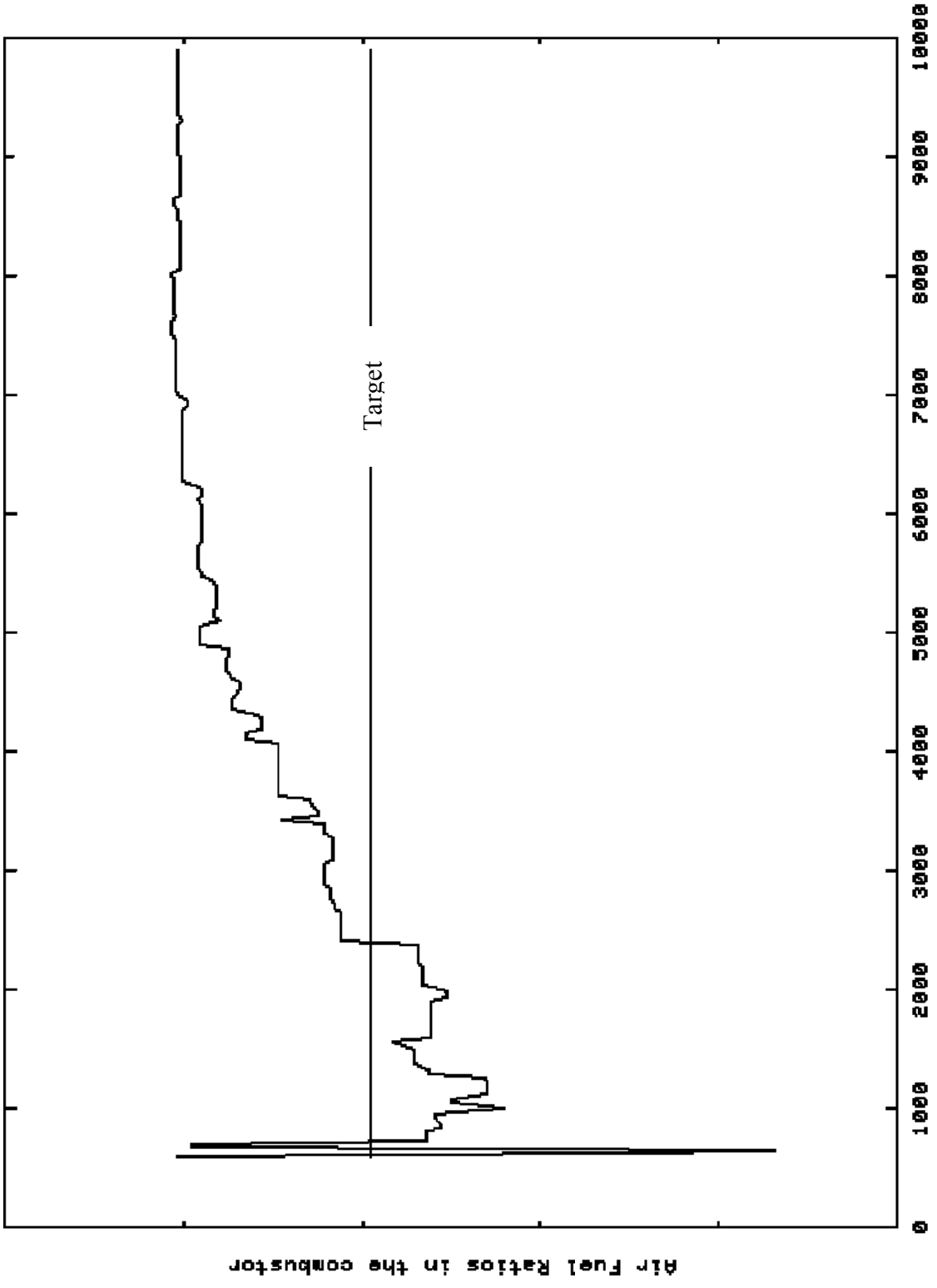

Figure 10: Evolution of the intermediate zone air/fuel ratio

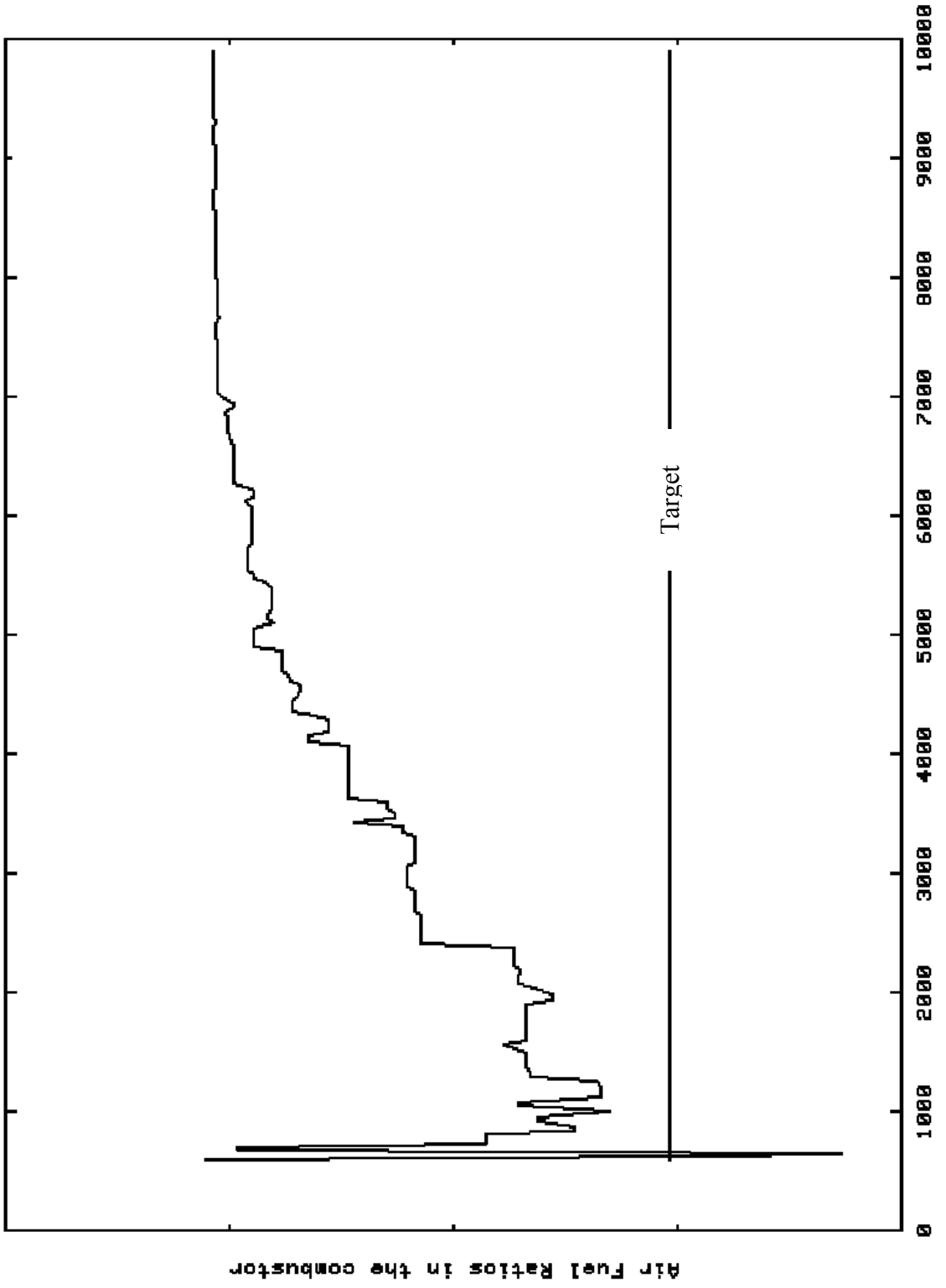

Figure 11: Evolution of the dilution zone AFR.