PROGRAMMING AND COMPUTING TOOLS FOR JET ENGINE CONTROL DESIGN

Florian Vary[+], Luc Reberga[++]

[+] *Snecma Moteurs, Villaroche, France, florian.vary@snecma.fr*
[++] *CNRS-LAAS, Toulouse, France, lreberga@laas.fr*

Abstract : This paper briefly presents ATOL, which is a tool developed at Snecma Moteurs for jet engine controller design. ATOL could be considered as an interfacing tool with many identification and design algorithms. ATOL is able to make advanced synthesis of controllers, such as LPV H_∞ synthesis by solving LMIs. ATOL can also design controllers by using a traditional pole placement technique.

Keywords : Engine control, Computer-aided control system design, turbofan engine

1. INTRODUCTION

Today, most jet engines in operation all over the world are driven by single input-output (SISO) control systems. This mainly consists in regulating the fan speed or various pressure ratios by controlling a fuel gate-valve (Spang and Brown, 1999). Some SISO controllers can also be used to fulfil other goals : for instance, military turbofans have generally further needs such as nozzle control, which is considered to be a SISO control problematic in some cases.

On the oldest engines, the SISO control is provided by an hydro-mechanical follow-up system commonly referred to as an HMU (Hydro Mechanical Unit). In short, the latter is mainly made of bi- and three- dimensional cams, bellows, fly-weights, etc… (Scoles, 1986). As far as more recent engines are concerned, this control is provided by an electronic controller which increases the in-flight and maintenance functionalities, as well as the accuracy of the control in terms of thrust and engine protection. It led to a wider operational domain for a given engine, and at the same time it enabled the development of more recent engines with higher performances. Initially designed with analogical electronic systems, this technology has been supplanted twenty years ago by numerical electronic systems called FADEC (Full Authority Digital Electronic Control). Today, these are widely used to control aero-engines. With the FADEC control system and its embedded computing power, which grows years after years, control laws can be much more complex and accurate than in the past. What is more the FADEC easily covers the computing time requirements for SISO controllers.

Because of the rather low complexity of such SISO control laws, their designs are generally realised by using only the pole placement technique based on classical linear control theories developed during the ’50s and ‘60s. This traditional controller design process is broken into two steps : first, one designs local linear controllers based on the linearizations of the jet engine - a non-linear system - at several different operating points. Then, a global non-linear controller is obtained by interpolating the gains of the local operating points.

This method has been applied successfully for many years and is still very popular. Nevertheless, with the introduction of additional engine actuators

on current and future engines and with the possible interactions between various SISO feedback controllers, the multivariable control law design (MIMO) is progressively becoming an unavoidable technique for the next decades and will tend to replace traditional SISO methods.

Besides, the gain scheduling technique applied on SISO controllers is an ad hoc method : stability, robustness and performance cannot be guaranteed globally (Athans and Shamma, 1992). Extensive simulations must be undergone by the control laws to infer their global performances. Since recent research works on gain scheduling (Bruzelius, 2004), this matter of fact could be overcome by the use of LPV (Linear Parameter Varying) controller design techniques, which preserves the performance, robustness and stability from local controllers to global scheduled controllers.

At last, since parametrical uncertainties - heat transfers, engine modelling errors, uncaptured nonlinearities, engine deterioration, engine to engine scatters, etc… - can hardly be taken into account when using classical pole placement techniques, the robust control design methodology appears as a method particularly well-adapted to the jet engine field.

Subsequently, a major evolution of the control engineering culture in jet engine manufacturers is currently needed. This consists in developing the LPV MIMO robust controller design.

2. THE POLE PLACEMENT HEGEMONY

Despite the numerous studies that have been carried out since the latest ‘70s, the use of multivariable controls on turbofan engines still remains a relatively new and limited development. This could be explained by the higher complexity of the MIMO control for an equivalent achievable performance between SISO and MIMO controllers. Indeed, for a multivariable system with little coupling effects – a frequent case - , an appropriate designing of several SISO controllers can often satisfy properly the engine requirements. Moreover, scheduling a MIMO controller can be both a tedious and time consuming task, and the implementation of the anti-windup scheme - to prevent actuator saturations or engine protection saturations - is significantly more complex in comparison with SISO controllers.

Another possible reason that could explain the lack of consideration taken about the multivariable control is the difficulty to translate engine control requirements into criteria used for control synthesis. In early time, the multivariable design studies were limited to LQ, LQG and LTR techniques. But, quadratic optimisation criteria are actually quite far away from the immediate concerns of the engineers who are focused on the system time response and on the stability margins. Fortunately, since the beginning of the ‘90s, the development of robust control synthesis with convex optimisation algorithms has provided tools to find the best controller in terms of robustness and time response matching. Nonetheless, even in this case, the definition of criteria, which consists in specifying the different weighting filters, remains quite a tricky task for designers. This general remark about the non-explicit link between engine control requirements and synthesis criteria can also explain why advanced design methods, like H_∞ techniques, are rarely used even for SISO controllers design.

Other reasons could be found and are not exposed in this paper. But the main thing that must be underlined is that the overall designer community - used to referring to the SISO pole placement technique - still continues to work using this design method, which is well-mastered, simple to use, and can easily fulfil engine requirements. Engineers, as human beings, are naturally reluctant to change their working habits when a new method is felt as restricting. The same thing happens when it is more complex and needs more technical skills, or when it does not provide - according to them - a significant improvement in the controller performance. In opposition to this, a new design technique can be easily accepted if it clearly reduces the time spent designing, and if engineers can have a better and more precise view of what they are doing at each stage of the design process.

One way to popularise the LPV robust control technique could be to supply a single tool to engineers. This would enable them to make pole placement as well as robust control synthesis. If this tool is designed well enough and makes the pole placement faster and easier, it could become extremely helpful and be widely used. If it also contains intuitive and advanced functionalities based on LPV robust control to do the same work, it will meet all the requested conditions for the engineers to become more familiar with this new design technique. With exhaustive analysis functionalities, the contribution of each method could be easily proven. As a consequence, new controller design techniques could become trustworthy and commonly used.

Then, to reach this target, an engine control design tool has been developed at Snecma Moteurs for two years.

3. THE ENGINE CONTROL DESK

ATOL is the French acronym for “ATelier contrOLe moteur” which means “Engine Control Desk”. ATOL is a Snecma Moteurs engine control tool implemented as a MATLAB® toolbox. The starting point of the development of ATOL goes back to autumn 2002 when the following fact was brought to the fore : despite the various research

studies in multivariable and robust control that have been undergone last decade at Snecma Moteurs, the design process has not evolved, and for the results of these studies to be checked and reused, a long time tedious work would be required. Indeed, technical paper reports were very often only the last trace. Besides, a research collaboration on robust multivariable control with the National French Research Centre CNRS-LAAS had begun in 2002. To maximize the fallouts of these studies on the design process, exchanging technical works and data in a structured way was needed. It rapidly appeared that a common design tool between Snecma Moteurs and the LAAS would be an attractive solution.

Today, several development versions of this tool have been completed. Each of them providing it with progressively more and more functionalities. In September 2004, the ATOL version 3.0 is capable of different tasks ranging from system identification to controller design and validation. ATOL is a generic object oriented tool, which can be adapted to any design processes : every user can customize its own design process if necessary, but some design processes are already given as a reference. Today, two different design process families are proposed for SISO controllers : one consists in a pole placement technique - carried out on local models - whose identifications have been performed automatically by ATOL itself. Then, an automated linear scheduling is done. The second process is based on LPV modelling from which an H_∞ robust controller is synthesised. Methods for MIMO controller design have not been implemented yet but developments are currently undergone at the CNRS-LAAS and at Snecma Moteurs.

To make this tool easier to use, several objects have been defined inside ATOL : model objects, controller objects, signal objects, disturbance objects, LFT objects, and many others. At each stage of the process, the properties of the different objects handled can be shown.

The identification task on ATOL is based on extensive simulation data from a Simulink® engine thermodynamic model. First, the transfer to be identified must be defined by a transfer object. Then, disturbance signals - to be applied on the model inputs for the identification - need to be described as disturbance and signal objects. All of these objects as well a point objects that represent the different operating points around which a local model is expected, are passed as parameters of the identification function. This function performs systematic disturbances on engine model inputs as well as an automated identification of local models. The identification can be customised by choosing the best method within an algorithm library. Model order can be variable and different types of models are supported : time delayed models, continuous or discrete models, transfer functions or state-space models. To enable a proper interpolation of local models, the identification base or the model structure could be parameterised. At last ATOL also provides model order reduction functionalities.

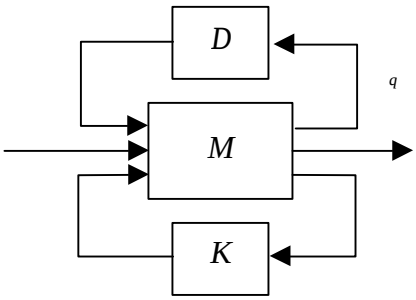

Fig. 1 –Linear fraction form of a state-space system M with its controller K. This common linear robust control setup uses also a block named Δ, which can represent different kinds of uncertainties. The LPV modelling with LFTs uses this block to describe the scheduling parameter dependency.

After having successfully identified the various local models, a global LPV model can be determined by using, for instance, LFT modelling based on polynomial approximation. The dependence of the scheduling parameter θ on the gain-scheduled LFT model is implemented by the uncertainties Δ block (fig. 1). The vector θ could be defined by an appropriate scheduling parameter study. That is the reason why ATOL provides with some specific tools in order to find the best scheduling scalar ($\dim\theta = 1$). The case where θ is a vector ($\dim\theta > 1$) still has not been implemented.

ATOL can also design controllers. It consists in finding the controller gains from local or global models. This functionality is provided by a single function called “design”. As far as the identification function is concerned, a controller design algorithm library is supported so that either the pole placement technique or a LPV H_∞ controller synthesis can be used. In this last case, ATOL intensively calls for the LMI parser YALMIP with the SEDUMI interface. Concerning the pole placement algorithm, the controller structure can be selected before the gain computation. All design algorithms provided can synthesize discrete or continuous controllers.

The last designing stage consists in validating the gain-scheduled controller over the entire flight domain. With ATOL, step response of the closed-loop system can be shown as well as the Nyquist or Nichols or pole diagrams. ATOL can also give stability margins on each and every operating point and can characterize the maximal acceptable time delay without stability loss.

In the short-term, a MATLAB® implementation of discrete controllers will be available in an ATOL Simulink® library. These controller blocks will be compatible with the automatic critical-code generation tool used at Snecma Moteurs, and might be put in the engine control software specification that is today realised on Simulink®.

4. APPLICATIONS TO A TURBOFAN

In this section, several applications of ATOL on a jet engine are shown to illustrate the simplicity of the programming, which is rather intuitive. Some graphical results are also shown in order to have an idea of ATOL capabilities. In the following six use cases, ATOL syntax has been slightly modified so as to be more comprehensible. Each use case represents a single stage of the design process : the identification task, the scheduling parameter selection task, the global modelling task, the controller design task, the controller performances validation task and the controller re-designing task.

4.1 Identification

Here is an example of a multivariable model identification. The model to be identified is named G. The latter has the main fuel flow WF32 and the nozzle area A8 as inputs, and the fan speed XN2 and the DPQ23 as outputs (fig. 2).

Fig. 2 – Multivariable model of a military turbofan G

First, a `slkmodel` object representing the Simulink® thermodynamic engine model must be defined. This model could be a Fortran code interfaced with Simulink® by an S-function. In the following ATOL command line, input parameters of the object constructor `slkmodel` have been simplified, and only the model file name is given :

```
EngineModel = slkmodel('SimulinkModel.mdl');
```

During the second step, inputs and outputs of the model G must be defined :

```
MainSystem = transfer('G',{'WF32' 'A8'},{'XN2' 'DPQ23'});
```

Then, different disturbances needed for the model identification are to be parameterised. Each disturbance is composed of several signals to be applied on system inputs. On the MIMO system G, a WF32 fuel flow step is applied while the input A8 is frozen at its initial value. Another disturbance is also defined. It consists in an A8 nozzle area step while the input WF32 is frozen :

```
WFSignal= signal('WF32','step','amplitude',30,'time',20);
A8Signal = signal('A8','freeze','time',20);
WFStep = disturbance([WFSignal A8Signal],'duration',40);
WFSignal = signal('WF32','freeze','time',20);
A8Signal = signal('A8','step','amplitude',0.02,'time',20);
A8Step = disturbance([WFSignal A8Signal],'duration',40);
```

Later, at least one operating point object must be created. On this example, only a single operating point is given. It corresponds to the ground steady-state engine condition at a PLA of 45°.

```
OperatingPoint = point('initparameters', ...
{'Z' 0 'Mach' 0 'PLA' 45},'disturbances',[A8Step WFStep]);
```

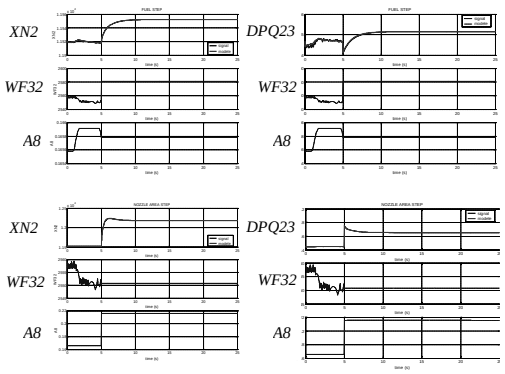

Fig. 3 – Comparison between thermodynamic engine model step responses (XN2 and DPQ23 in gray) and linearized MIMO model step responses (XN2 and DPQ23 in black). The simplified model fits exactly simulated engine data.

Finally, with all these previously defined objects, the identification of the local model could be undergone at the given operating point :

```
LocalModel = identify(MainSystem,OperatingPoint, ...
EngineModel,{'method' 'mlaas' 'order' 3 'type' 'ss'})
```

The local model returned is a 3rd order discrete state-space model. A visual validation of the identified model could be easily done (fig. 3) :

```
validation(LocalModel,MainSystem,OperatingPoint);
```

If rather than a single operating point, an operating point array is given, a local model array is outputted. In this case, each element of the model array corresponds to each operating point of the point array.

4.2 Selection of the scheduling parameter

In order to find a suitable scheduling parameter for global modelling or for controller scheduling, every local models are to be described in the same base. Here is an example of local models computed into a canonical state-space realisation - the companion canonical form where the characteristic polynomial appears in the right column- :

```
LocalModels = chbase(LocalModels,'companion');
```

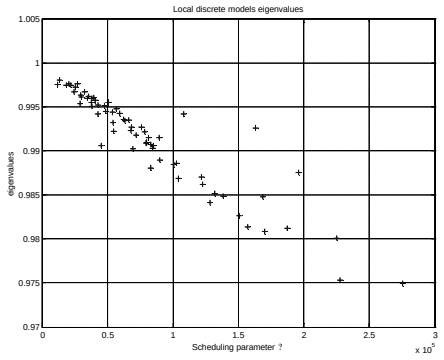

Fig. 4 – Eigenvalues of XN2 discrete 1st order local models on a wide variety of operating points in the whole flight domain. The scheduling parameter θ used is a PS32 (static pressure in the engine combustor) dependant function. One can notice that the behaviour of eigenvalues is almost linear.

In a same system of state coordinates, the variations of the parameters of 1st order local models on a given point in the flight domain are monotonic

throughout the engine working line. From these models, the best scheduling parameter θ is founded when the parametric model scatters are minimal (fig. 4).

4.3 Global modelling

When the best scheduling parameter θ has been found, a global model can be computed. ATOL supports several types of global LPV model. Of all global models, PLS (Piecewise Linear System) and LFT models are the most frequently used. A PLS model is a state-space model whose parameters are linearly interpolated from the parameters of local models. An LFT LPV model is a LFT model whose scheduling parameter dependency is implemented by the parametric uncertainties block Δ (Boyer, 2004). To create a PLS global model from a local model array, the scheduling parameter or the scheduling formula must be specified. On the following example, the PLS model outputted by the `mod2pls` function is a PS32 scheduled model :

```
PlsModel = mod2pls(LocalModels,'PS32');
```

In the same way, LFT LPV model can be created by specifying in this case the polynomial order :

```
LftModel = lsq(LocalModels,3,'PS32');
```

4.4 Controller Synthesis

ATOL version 3.0 is capable of designing SISO controller. The design could be realised thanks to the classical pole placement technique (Reberga, 2004):

```
K = design(LocalModels,{'method' 'pplaas' 'ksi' 1.7});
```

A robust H_∞ LPV controller synthesis by solving LMIs could also be used as easily as the pole placement technique :

```
K = design(LftModel,{'method' 'lmiHinf' 'W1' tf(1.15,[0.4
1]) 'W2' tf([0.1 0],[1 7]) 'W3' tf([1],[1 5])});
```

W_1, W_2 and W_3 parameters are the weighting filters used for the H_∞ robust design (fig. 5).

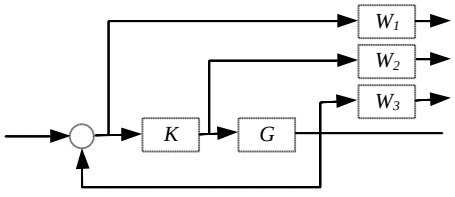

Fig. 5 – Weighting filters pattern used by ATOL for the H_∞ robust controller synthesis. W_1, W_2 and W_3 are the weighting filters, K is the controller and G the engine model.

The imperceptible difference between these two methods lies in the number of code lines called : when the pole placement technique requires less than a hundred of code lines, the LPV controller synthesis calls for more than ten thousand code lines.

4.5 Performance validation

After computing the controller gains, the most frequently used diagrams in the control designers community are available : Bode diagrams, Nichols diagrams, Nyquist diagrams, … (fig. 6). These diagrams can be plotted by a very simple syntax :

```
OpenLoop = series(K, LftModels);
nichols(OpenLoop);
ClosedLoop = feedback(K, LftModels);
nyquist(ClosedLoop);
step(ClosedLoop);
pzmap(ClosedLoop);
```

The syntax would be exactly the same with the `LocalModels` model array. Gain and phase margins could also be immediately computed by ATOL.

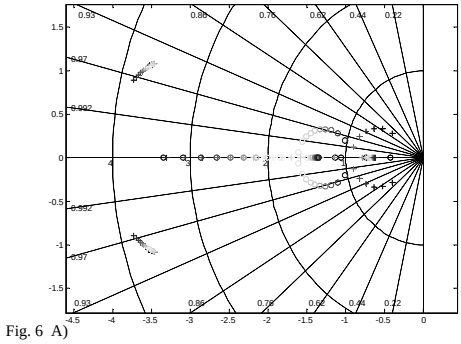

Fig. 6 A)

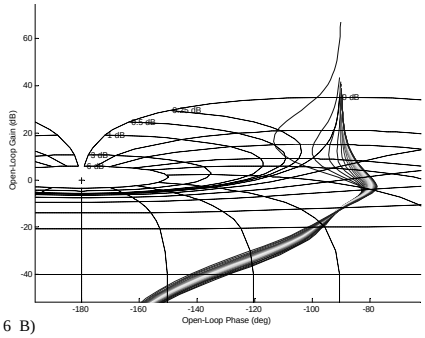

Fig. 6 B)

Fig. 6 – Closed-loop characteristics of a fan speed controlled by an LPV robust controller, whose gains have been synthesized by optimisation algorithms based on LMIs. A) pole and zero location. B) Nichols chart. Both figures represent the

4.6 Controller re-engineering or re-using

Another great contribution of ATOL lies in the simplicity of a controller re-designing. Very often, data used for the first design of a controller, such as local models, are not recorded by any given ways.

After several years of operating life, some updates of the engine control software involving the control laws might occur. Also, sometimes existing control laws are reused for other purposes or for controlling derivative versions of an initial engine. For these different cases, a closed-loop analysis must be carried out. Because design data were not kept, model identification needs to be carried out again as well as the closed-loop computation in order to infer the controller behaviour. This could take several hours of work. Thanks to ATOL, it only takes a few minutes to check closed-loop performances and robustness. Indeed, existing

modellings of controllers used at Snecma Moteurs are available on ATOL, and a controller object can easily be created from the discrete gains implemented in the control software:

```
% Controller object creation
PS32TAB = eval(LocalModelsXN25,'PS32');
GHPMA1 = 4.968;
GHPMA2 = -4.692;
K1 = PS32TAB* GHPMA1;
K2 = PS32TAB* *GHPMA2;
Controller = crdcontroller('PI',0.02,[K1 K2],PS32TAB);
```

Besides, the identification task is automated on ATOL, and identification data are recorded in what could be called a data base managed by ATOL itself. Eventually, identified models could be persistent objects and could easily be reloaded. The given example describes a re-use attempt of an XN25 limitation controller (see fig. 7).

```
% Open-loop and closed-loop computations
OpenLoopXN25 = series(Controller, LocalModelsXN25);
nichols(OpenLoopXN25);
ClosedLoopXN25 = feedback(Controller, LocalModelsXN25);
step(ClosedLoopXN25);
```

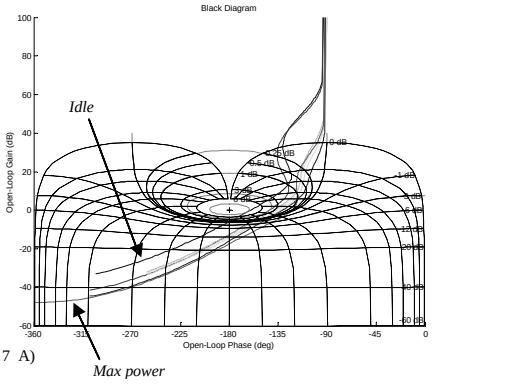

Fig. 7 A)

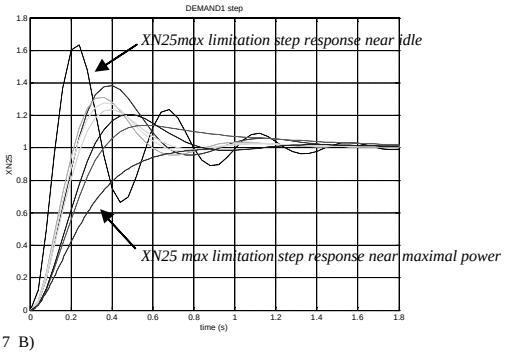

Fig. 7 B)

Fig.7 – Closed-loop characteristics of an existing controller that limitates the maximal high pressure spool speed - XN25 limitation -. At the defined XN25 limitation value - near maximal power -, the controller step response B) and robustness A) is satisfying. But at intermediate engine powers a loss of stability and of the damping factor are shown . This study realized with ATOL indicated in only a few minutes that this controller was not suitable for controlling the high pressure speed XN25 at idle.

5. CONCLUSION

In this paper, the ATOL tool and its functionalities have been briefly presented. The easy use and the high process speed both make of ATOL a very helpful tool for jet engine control design. Today, two designing processes are supported by this tool : classical pole placement technique can be used as well as an LPV robust controller synthesis. These controller design algorithms are developed only for SISO systems, but a MIMO implementation is in the progress of being developed. The implementation of some other design algorithms - local LQ and H_∞ syntheses for instance - are planned to increase the number of algorithms provided in the synthesis method library. Specification of additional parametrical uncertainties on LFT models will also be possible in future versions of the LPV robust H_∞ synthesis algorithm. Concerning the identification functionalities, these will be increased very soon. Suitable identification algorithm for the use of real noised engine data will be added. In the future, ATOL could also be used as an efficient data base managing tool for the identification signals.

6. ACKNOWLEDGEMENTS

The authors thank Snecma Moteurs for founding this work and for giving the publication permission. They would like to thank the CNRS-LAAS, and particularly Jacques BERNUSSOU and Didier HENRION for their help. They also gratefully acknowledge David BOYER - ENSAE SUPAERO engineering student - for his remarkable work on the LPV H_∞ controller synthesis implementation inside the ATOL tool.

NOMENCLATURE

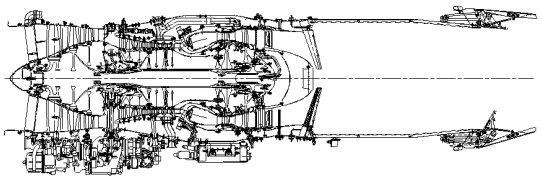

Fig. 8 – Cross section of a typical twin spool mixed flow military engine.

XN2 : fan speed or low pressure core speed	A8 : nozzle area
XN25 : high pressure core speed	Z : altitude
DPQ23 : fan throttling parameter	XM : inlet Mach number
PS32 : combustor static pressure	PLA : power lever angle
	WF32 : fuel flow

REFERENCES

Athans Michael, Shamma Jeff S. (1992). Gain Scheduling: Potential Hazards and Possible Remedies. *IEEE Control System Magazine (No. 12 / pp 101-107)*

Boyer David (2004). Commande LPV robuste de turboréacteur. *Snecma Moteurs Report*

Bruzelius Frederik (2004). Linear Parameter-Varying Systems. *PhD Report – Chalmers University of Technology*

Reberga Luc (2004). Rapport sur la création et l'analyse du modèle complet de turboréacteurs. *CNRS-LAAS Report*

Scoles Richard J. (1986). FADEC, Every Jet Engine Should Have One. *SAE Technical Paper 861802*

Spang Austin H., Brown Harold (1999). Control of Jet Engine. *Control Engineering Practice (No. 7 / pp 1043-1059)*