29th Congress of the International Council of the Aeronautical Sciences
St. Petersburg, Russia; September 7-12, 2014

DESIGN OF GAS TURBINE CONTROL SYSTEMS AT DIFFERENT LEVELS OF ABSTRACTION

N. Kapyrin[1], Moscow, RU
[1]MAI (Moscow Aviation Institute)

Keywords: *gas turbine engine, modeling, identification, Hammerstein model, ARMAX*

Abstract

This paper presents a method we used to categorize the simplifications introduced during a gas turbine controller design process, before identifying its parameters using experimental data.

We outline the steps to simplify a nonlinear system down to a level where parametric identification is applicable

1 Introduction

The complexity of gas turbine engines and the high costs of trial runs always call for modeling approaches capable of extracting new data out of previous trials (from delivery trials, certification, etc.). A mathematical model has to be fitted upon this experimental data in a process called identification, where a model is a set of hypothesis about how the system parameters are linked to each other.

Our work aims to provide a set of categories in which to fit these hypotheses, after what each category will receive its own way of measuring its impact on the scope of the model (*abstraction metrics*), enabling us to quantify the level of specificity (or loss of generality) of the model. The current step consists in explaining the workflow, adopted in order to select specific structures from a mathematical model, which will be automatically reduced on a later stage, giving a simpler model for which a controller may be built with an analytical approach – most suitably, a linear or space-state system, because a numerical approach may prove impossible for a notably nonlinear system with hundreds of unknown variables, as a gas turbine engine model.

2 The problem

A gas turbine manufacturer may supply sufficiently precise data helping model developers to reach a good level of realism with a linear, black-box model. However when real operating conditions are studied or when the performance degradation of one specific engine must be taken into account, a careful reconsideration of the developer's information is important along with the model identification.

3 Methods

The aim of model identification is to find one model in a set, closest to experimental results and most simple in the same time. The most robust algorithms are performed on fixed-structure model in the parameters space and the process in called *parametric identification*, the most generic algorithm being ARMAX (Auto-Regression with Moving Average and eXogenous inputs). If the model structure is unknown, a *structural identification* is done, either by completing the model with an algorithm enumerating potential structures (with which the identification becomes parametric), or by introducing new elements in a flexible structure, using genetic approaches such as GMDH (Group Method Data Handling, [1]).

GMDH has spawned many adaptations, it is central to many data mining and knowledge discovery systems, including ours.

The most widely used structures in nonlinear structural identification are Volterra series, Wiener, Hammerstein models, or a combination of them. The outline in [2] on controller synthesis for Hammerstein models have been very helpful along the path of developing this approach.

4 Results

4.1 An introductory example

We use the structure shown in Fig.1, where a control input *u(t)* is passed through a nonlinear set of static equations, it's outputs go through a linear dynamic system (a transfer function or a state-space system, for example), resulting in a set of system outputs *y(t)*.

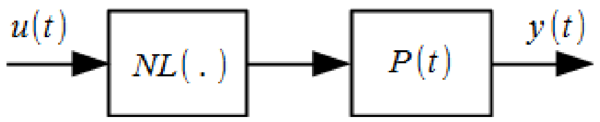

Fig. 1. General Hammerstein model

For a demonstration, we have taken a simple example from [3], where the system of equations (1) is used to illustrate a problem in gas turbine model identification. The paper presents an attempt to numerically find the values $a_1 \dots a_{14}$ by adding a polynomial weighting function upon all the parameters, gradually phased out as the identification proceeds, centered around the expected values. This is a sound strategy if we are really looking only for shifts from expected values, and it is possible only when we know how the identified parameters are related to real engine parameters, as it is very difficult to automatically set expectations on non-physical parameters.

$$\begin{aligned} \dot{\bar{n}}_c &= \varphi_c n_1 \\ \dot{n}_1 &= a_1 n_1 + a_2(\bar{n}_c^s - \bar{n}_c) \\ &\quad + a_3(\bar{n}_c^s - \bar{n}_c)^2 \\ \dot{\bar{n}}_f &= \varphi_f \bar{n}_f^s - a_6 \bar{n}_f \end{aligned} \qquad (1)$$

This system contains a controller outputting a command n_1 acting on the compressor rotor speed $\bar{n}_c$ according to a setting $\bar{n}_c^s$, while a fan rotor speed $\bar{n}_f$ is stabilized around a parameter $\bar{n}_f^s$. It also includes number of processes described by the functions φ_f and φ_c explained below, in the system of equations (2).

$$\begin{aligned} \bar{n}_f^s &= (a_9 + a_{10}\bar{n}_c^{cr} + a_{10}(\bar{n}_c^{cr})^2)\sqrt{\frac{T_{in}}{T_{H0}}} \\ \bar{n}_c^s &= a_{12} + a_{13}\delta_{ctrl} + a_{14}\delta_{ctrl}^2 \\ \varphi_c &= a_4 + a_5 \frac{p_{in}}{p_{H0}}\sqrt{\frac{T_{H0}}{T_{in}}} \\ \varphi_f &= a_6 + a_7\dot{\bar{n}}_c + a_8\dot{\bar{n}}_c^2 \\ \bar{n}_c^{cr} &= \bar{n}_c\sqrt{\frac{T_{H0}}{T_{in}}} \end{aligned} \qquad (2)$$

The compressor rotor speed setting depends on the pilot input – a lever angle δ_{ctrl}. Some values are used in a corrected form ($\bar{n}_c^{cr}$), accounting for the difference in temperature and pressure at sea-level (T_{H0}, p_{H0}) and in flight (T_{in},p_{in}). The rest are 14 parameters $a_1 \dots a_{14}$, which must be identified.

4.2 Model transformation

That system was automatically translated into a Hammerstein form through a set of automated transformations, giving a model, part of which is shown in Fig.2. The resulting nonlinear system combines the inputs into 30 new variables and is followed by a matrix combining the unknown parameters $a_1 \dots a_{14}$, entering into the same 3 differential equations from the source model (plus an equation for $\ddot{\bar{n}}_c$ included implicitly, as $\dot{\bar{n}}_c$ appears in right-hand side equations).

The resulting model has now the form:

$$\dot{y} = \sum_{i=1}^{4} NL_1^i(a_1 \dots a_{14}) NL_2^i[P\{u(t), y(t)\}]$$

Here $y = (\bar{n}_c, \bar{n}_f, n_1, \dot{\bar{n}}_c)$, the function NL_1 keeps the combinations of the parameters $a_1 \dots a_{14}$, they will be identified in the ARMAX procedure, after that each parameter will be derived from this system of equations.

The function $NL_2(.)$ is the nonlinear combination of state variables, which will be subject to simplifications.

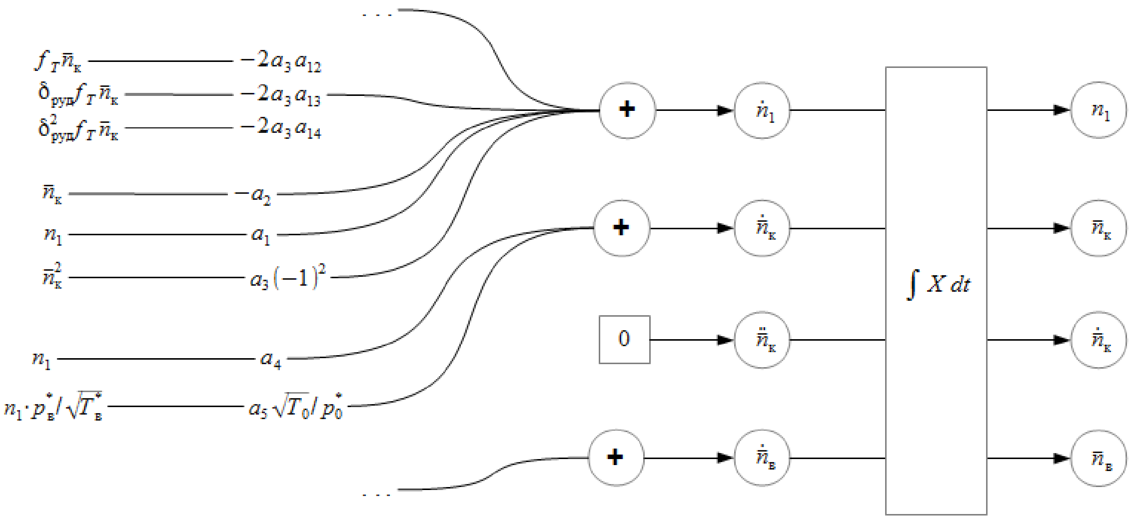

Fig. 2. A graph of equations in Hammerstein form (a part of the model)

The transformed model performs exactly the same as the initial system. An ARMAX identification is possible at this point, using the nonlinear part as a precompensator for experimental data, but we can also start modifying the nonlinear part and measuring the impact of introduced simplification on how well our model covers experimental data.

4.3 Abstraction metrics

So far we have only been able to try to introduce three of the simplifications described by B.P.Ziegler in [4], namely:

- part omission,
- replacement by a constant,
- replacement by a probability function (these blocks are only used when simulating sensors, and is equivalent to the “replace by constant” action in the model).

The forth category (part addition) is not applicable since the number of terms is intended to be limited by the basic equations. The only part to be added is a controller.

This allowed us to produce a linear model which can be easily inverted and serve to deign a PID controller. Another important outcome is that the parts of the model, that were simplified, can be simulated on their own, and their impact on the model may be taken into account by comparing simulation results of these independent models with the initial run.

4.4 A larger model

In order to give further support for the approach, a Modelica textual model has been built and the transformation has been tested, with limited success due to the large amount of equations in the model (100 equations, 115 variables for a one-rotor turbojet). A typical gas turbine model contains several piece-wise linear datasets, the most important one being the compressor map.

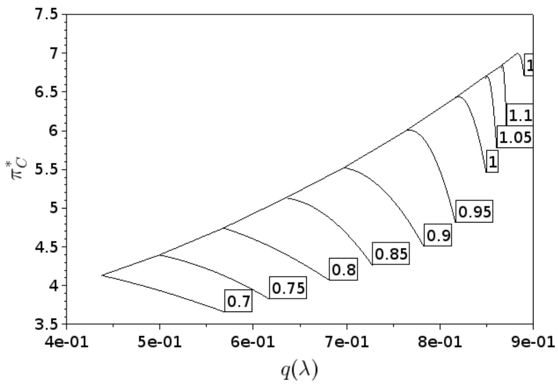

Fig. 3. Compressor map

A way to incorporate this data in our study was found by fitting it into a polynomial expression through least-square fitting in Scilab. The map is used to find the compression ratio π_c^* from relative rotor speed $\bar{n}_c$ and relative flow density $q(\lambda)$. It has been fitted to the following function with less than 5% error and used in the simplification process.

$$\pi_c^*(\bar{n}_c, q(\lambda)) = 0.0198\bar{n}_c^2 q(\lambda)^2 - 0.037\bar{n}_c q(\lambda)^2 - 3.22\bar{n}_c^2 q(\lambda) + 6.67\bar{n}_c q(\lambda) + 0.0154 q(\lambda)^2 + 70.3\bar{n}_c^2 - 3.17 q(\lambda) - 163\bar{n}_c + 90.04$$

We note that the polynomial coefficients may be used as weights for the terms of the function, and indicate that some of the terms may be safely removed. The interpolated and simplified dataset is shown in Fig. 4, next to the initial compressor map.

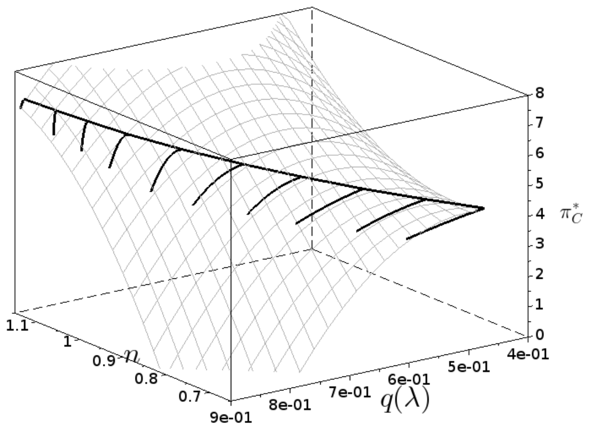

Fig. 4. Interpolated compressor map

More results are given in the poster accompanying this article.

4.5 A meta-model

The last part of this approach, essential to the simplification process, as it helps handling structural simplifications based in theory, is the construction of a meta-model.

All the basic physical equations for a gas turbine model are described in widespread literature, for example [5]. A meta-model designed in this work holds these equations together in a type of class diagram.

Its has been designed and constructed using the yEd graph editor, allowing for automated transformation to be done using scripts.

5 Discussion and Conclusion

A model's correctness is based on experimental data, and likewise experimental data has no predictive power without a model. The introduction of new hypotheses (physical equations or observations) or new data from different sources increases the predictive power of a model, in the same time making correctness analysis more difficult. The outcome of this study may give a tool to control this process.

The aim of this work is to automate model simplification, hence the approach it is designed to handle any number of equations as one big system, keeping the aggregated model readable.

Only the most basic tools were used, in order to allow for more interoperability and easier introduction in the teaching process, where this work has the most application at this stage.

References

[1] Madala H.R., Ivakhnenko A.G. *Inductive Learning Algorithms for Complex Systems Modeling*. 1st edition, CRC Press, 1994.

[2] Guo F. *A New Identification Method for Wiener and Hammerstein Systems*. PhD Thesis, Forschungszentrum Karlsruhe, 2004.

[3] Kolpakov V.F. An approach for gas turbine model identification. *NTK*, Vol. 1, No. 1, pp 210-216, 1993.

[4] Ziegler B.P., Praehofer H., Kim T.G. *Theory of Modeling and Simulation*. 2nd edition, Academic Press, 2000.

[5] Shlyakhtenko S.M. *Theory and design of gas-turbine engines*. 2nd edition, M.Mashinostroyenie, 1987.

6 Contact Author Email Address

nikolay.kapyrin@gmail.com

Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS 2014 proceedings or as individual off-prints from the proceedings.