COMPARING MATRIXx AND MATLAB FOR MODELING, DESIGNING AND SIMULATING FLIGHT CONTROL SYSTEMS

Gilberto da Cunha Trivelato, M.Sc.
EMBRAER – Empresa Brasileira de Aeronáutica S. A.
gtrivel@embraer.com.br
P.O. Box 8050
12227-901 - S. José dos Campos – SP – Brasil

Marcelo Lopes de Oliveira e Souza, Ph.D.
INPE – National Institute for Space Research
DMC-Division of Space Mechanics and Control
marcelo@dem.inpe.br
Av. dos Astronautas, 1758
12227-010 - S. José dos Campos – SP – Brasil

Abstract. In this work we compare the MATRIXx and the MATLAB environments from the perspective of modeling and simulating aerospace vehicles, specially aircrafts and satellites. Some criteria are adopted, including ease to install and use, platform demanded, friendly interfaces, resources for modeling and simulation, stop and go, re-run, generation and presentation of results, code generation, code documentation, etc. We will discuss and show their respective capabilities and reasons to become the de facto standards in the aerospace industry.

Keywords. Modeling Environment, Simulation Environment, Aerospace Vehicles.

1. Introduction

Since its beginning, the aerospace industry became one of the main users and beneficiaries of simulation, stimulating and permitting the development of tools even more powerful. Successively, it took advantage of the use of: physical simulation for design and training; simulation in analog computers for control systems; and, in the age of digital computers, digital simulation in the remaining areas of knowledge involved in the diverse phases of development of aerospace vehicles. Today, simulation is recognized even more as an essential tool for the diverse phases of product development such as: specification, design, development, production, tests, accreditation, maintenance, training, and marketing.

The occurrence of use of new materials, of new applications, of reduction in weight and consumption, of the search for optimizing such vehicle designs, and the need of reducing the production cycle, created new problems of distinct nature, that did not exist or that were neglected before, and that can be solved with the support of simulation and rapid prototyping, including those simulations with hardware or even pilots in the loop. It also became fundamental the reuse of models of the diverse systems along the diverse phases of the product as a way of reducing time and costs, increasing the reliability, the ease of accreditation, the fidelity for training, the reproduction and investigation of faults or accidents, etc.

2. Requirements of the aerospace industry

For the adequate use of simulation environments in the production cycles of an aerospace vehicle it is necessary to observe the important aspects of each phase with respect to those environments. The current designs in this area may be developed by teams distributed even in diverse places around Earth, which requires standardization and universality as a necessary requirement for the specification of systems, to have the same meaning among the diverse teams. In some areas, depending on the quality of the models used, the design phase merges with the specification phase. The interoperability of models and of the generated codes is essential for prototyping the systems under diverse technical aspects and in diverse places during the development phase, and to allow adequate environments for the integration and tests of subsystems. These same models must be reused in production and tests devices in the production line. All these prototypes, in the whole or in part, can and must be used for the accreditation of the product under the correspondent institutions (FAA, JAA, CTA, etc.), for training the members involved during the program, and for training the customers in the operation and maintenance of the final product. It must be possible the reuse and incorporation of legacy codes that were already verified, validated and accredited. With the inclusion of these requirements in the beginning of the specification phase, we have in the end truly simulated systems available for the investigation of failures and for maintenance of the product developed. The simulation software can be used...
for marketing or for support to the sales during product development and in the after-sale. The use of environments that meet these requirements from product conception to after-sale represents significant savings with benefits in the product reliability and in the reduction in the development time.

3. **Models of aerospace vehicles**

To be used in all phases of development of an aerospace vehicle, the models simulated must include all pertinent systems and must allow the adjustment of the complexity of each subsystem according to the application desired. For example, to meet these requirements, the simulation of an aircraft must contain: the models of the atmosphere, engines, aerodynamics, motion, other subsystems, landing gear, controls, graphical user interfaces-GUI (pilot displays, monitoring interfaces or data communication interfaces), and initialization routines. A common generic model of an aircraft is shown in Figure 1.

![Generic model of an aircraft](image)

**Figure 1. Generic model of an aircraft.**

It should also be possible to include and to exclude models of specific parts whenever necessary (e.g. of the electric system). A fundamental property of these simulated systems is to allow us to change blocks of the same subsystem, but with different levels of fidelity. A practical example on the need of different levels of simulation of the same subsystem can be verified in the use of the simulated electrical system when we are analyzing the control system of an aircraft or of a satellite: in the first case, the batteries can be modeled and simulated by state machines with ON and OFF levels; while in the second case, the battery charge level must be included in the model because it changes the capacity of the actuators (ex.: magnetic coils) then changing the control parameters. The investigation on the effects of the sampling period of aerospace vehicle control systems on the vibration modes of flexible structures had its importance increased recently due to the use of larger and lighter structures in satellites and due to the design of lighter and thinner aircraft wings.

4. **Modeling and Simulation Environments**

An study of diverse simulation languages and environments devoted to the control systems area was presented by Linkens (1993). There, diverse authors clarify the origins of the main environments currently available and the bases of new ones that appeared later. A typical application was presented by Papini (2000), involving the two environments most used in the aerospace industry (MATRIXx e MATLAB), upon which
we will concentrate our comparison efforts, despite the profusion of softwares available on the market for this area.

4.1. MATRIXx

A vision of the modeling and simulation environment of the MATRIXx family and of their components is presented in Figure 2. The Xmath is a work environment for the design and analysis with graphical visualization containing a package with programming language, debugger, and mathematical, model and function libraries. The SystemBuild is a modeling and simulation graphical environment that runs simultaneously with Xmath. The models are constructed from the block diagrams of its own library, from a specific module for state machine or from blocks developed by the user. The Autocode is a module that generates the C or ADA codes from the block diagrams automatically. The generated code is optimized for the use in real time control systems or in hardware in the loop simulations. If the system is distributed in different tasks with different frequencies then the Autocode groups them and organizes the tasks automatically, and generates the code in the form adequate and ready to work with the scheduler and the dispatcher of the real time operating system used.

![Figure 2. Architecture of the MATRIXx family. Source: Integrated Systems (1995).](image-url)

The DocumentIt allows the automatic generation of project documentation from the block diagrams, detailing the software design and supporting the hierarchical structure used in the SystemBuild. These documents include tables of block inputs and outputs, state transition diagrams and global variables. Basic models of documents can be included by the user, and they can be produced in the ASCII, Interleaf, Microsoft Word e Framemaker formats. By the user request, they can be produced complying with the DOD-STD-2167A standard. The RealSim is a hardware environment that allows the rapid prototyping and implementation of systems modeled in real time operating systems integrated with analog and digital I/O boards. The communication between the host and the target is via Ethernet and allows the download of codes
4.2. MATLAB

A vision of the modeling and simulation environment of the MATLAB family and its components is presented in Figure 3.

![Figure 3. Architecture of the MATLAB family. Source: The MathWorks(1996).](image)

The MATLAB basically is a computation, programming and visualization environment similar to MATRIXx containing its own language and enriched with specific tools (toolboxes) that are developed by third parties. The Simulink is a modeling and simulation graphical environment in which the models are constructed from block diagrams. It contains a specific module for the generation of state machine diagrams and another module for discrete simulation. The Real-Time Workshop is a module that automatically generates the C or ADA codes from the block diagrams. The code generation is not possible if we use MATLAB functions inside the Simulink blocks. The Report Generator allows the automatic generation of project documents from the block diagrams. They include figures, tables, state transition diagrams, and are produced in the HTML, RTF, XML e SGML formats. There are more than 200 toolboxes developed by third parties for the MATLAB environment, and in particular, more than one developed for data acquisition and stimulation with analog and digital I/O boards that allow rapid prototyping and implementation in real or virtual time.

4.3. Analysis of functions

We tried the MATRIXx 5.0 and MATLAB 5.1 versions. Both softwares were installed easily with demo licenses supplied by their owners. The MathWorks has an specific version of MATLAB for students and universities, and it is also possible to use floating licenses. The documentation provided by The Integrated Systems (1995) and by The MathWorks (1996) is abundant and of good quality. In the case of MATLAB there still are manuals of the many toolbox as well as many text books of courses that include such environment as support. Our limited experience can be summarized in Table 1. The interfaces are friendly and of easy learning. The resources available in both environments for modeling and simulation are equivalent, but are distributed in a slightly different way. In the area of control systems design we highlight the module for iterative design of control systems presented by MATRIXx. The transformation between continuous time and discrete time models in both ways is automatic in the MATRIXx environment with changes of block or superblock properties; while in the MATLAB environment we must enter in the “edition mode” and add samplers and holders to the system. In the MATRIXx environment specific modules are offered for the areas of Control Systems and Systems Identification while in the MATLAB environment such functions are offered through toolboxes, making it possible the acquisition of similar functions developed by different companies. The simulation results can produce 2D or 3D graphical objects that can be manipulated by the user in both environments. The MATLAB presents an wide set of graphical functions that practically meet all necessities, despite in the aeronautics area it is more important the connectivity with visual environments specific for the generation of displays (ex. VAPS of Virtual Prototypes).
We compared C language codes automatically generated in the AutoCode with similar codes of equivalent models generated by third parties in the RTW of MATLAB 6.0. In AutoCode it is necessary to define if we will emphasize the comprehension (by generating basic modules) or the code performance (by generating one single module). To have a separated code for each basic module or block it is necessary to generate the code of each block individually, otherwise we would have the code of the entire system as a single block. The variables are the same defined in the SystemBuild, and the generated code is of easy understanding due to the comments inserted in a clear way. These serve as additional documentation to the one already generated for the description of the blocks. The codes generated by both environments are not accredited by standard 178-B level A (life critical), despite there is a version of the F15-E aircraft using control laws directly generated by the AutoCode. Prudêncio (2000) presents the use of codes directly generated by the AutoCode as inspiration for a version in OCCAM2 of the control laws of a satellite studied by INPE. In the training aspects, MATLAB presents a larger availability of instructors due to its widespread dissemination in universities, and due to its commercialization policy of aggregating toolboxes developed by third parties. They allow the integration of legacy codes and can have blocks built from such codes.

5. Conclusions

Both environments analyzed showed satisfactory for the modeling and simulation of aerospace vehicles. The graphical interfaces are satisfactory, mainly in the MATLAB. But particularly in the aeronautics area, the more adequate is the connectivity with specific tools that automatically generate the code that really can be put on board of an aircraft (ex. VAPS). In this way, the representation developed will be evaluated from the viewpoint of “human factors” and of use during all the development phases of the product. The integration of such environments with synthetic environments is important for the Systems Integration and Tests Area of very complex systems as military aircrafts and space shuttles. It is possible to not loose the investments made in legacy systems (ex.: engines, aerodynamics) written in FORTRAN language but that are validated already. There is a large number of libraries available, particularly for MATLAB, with constant updates and tests by thousands of users distributed in different areas. With the continuous use of both environments we perceive specific points where a mutual contribution could happen for the mutual improvement, that certainly will happen due to the competitive market where they are inserted. Their dissemination is easy in universities and research centers considering the similarity and easiness of intuitive use of the graphical interfaces based on block diagrams. This has the power of liberating the user from repetitive tasks, allowing him to concentrate his efforts in the essence of the problems to be solved. These environments are becoming de facto standards in the aerospace area due to their dynamic capacity of meeting the ever growing needs; and from our point of view, they already became essential tools due to the reduction of deadlines and prices of projects in such area. For the aerospace industry, due to the big number of lines of code (between 0.5 and 3 million or more) and due to the complexity of the embedded programs, it is of fundamental importance the accreditation of the code generated from the models according the requirements of the accreditation institutions, particularly the
standard 178-B, level A. The benefits provided from the use of such environments will be observed in
crescent levels proportional to the levels of integration of the diverse parts of the program. But even so, they
will still be far from the possibilities if there is no an effective and general plan that put modeling and
simulation as a tool to be used and mainly reused in all phases of the program, or among the diverse programs
of a company or group of companies associated in the aerospace area.

6. Acknowledgements

We thank the following persons for the useful discussions on the technical questions presented here:
Sérgio G. Cavalcanti, Ph.D., from Opal-RT Technologies Inc. (http://www.Opal-RT.com), for his experience
in modeling and simulation of aircrafts in both environments; Silvano V. Prudêncio, M. Sc., from EMBRAER
(http://www.embraer.com.br), for his experience in modeling and simulation of artificial satellites in
MATRIXx; e, Adenilson R. da Silva, M.Sc., from INPE (http://www.inpe.br), for his experience in the use
and development of applications for simulations using the MATLAB environment. The ideas and opinions
expressed here reflect the limited experiences of the authors which are the solely responsible for them; and are
not and should not be considered as positions from EMBRAER and/or from INPE.

7. References

USA.
Proceedings of AIAA 2000 GNC, AFM, and MST Conferences, Denver, CO, USA.
Autônomo de um Satélite”, Dissertação de Mestrado, São José dos Campos, SP, Brazil, 167 pp.