Aerospace Europe Conference 2023
Joint 10th EUCASS – 9th CEAS Conference

Abstract #XXX *(to be filled by the organizers)*
Preferred Topics: FDGNCAV / SYSINT
Corresponding author: Chernetsov Dmitry
e-mail of corresponding author: dimitry.chernetsov@tu-berlin.de
Type: Oral
Status of corresponding author: Regular

Title

Model-Based Development of a Library with Standard Functions for Safety-Critical Flight Control Laws

Authors

Dmitry Chernetsov [1], Bryan Laabs [2], Robert Luckner [3], Ralph Paul[4]*
** Corresponding author*
[1] TU Berlin, Institute of Aeronautics and Astronautics, 10587 Berlin, Germany, dimitry.chernetsov@tu-berlin.de
[2] TU Berlin, Institute of Aeronautics and Astronautics, 10587 Berlin, Germany, bryan.laabs@tu-berlin.de
[3] TU Berlin, Institute of Aeronautics and Astronautics, 10587 Berlin, Germany, robert.luckner@tu-berlin.de[4]
[4]Leichtwerk AG, 38108 Braunschweig, Germany, ralph.paul@fd-tec.net

Abstract

Model-based methods can improve the efficiency of development for complex and safety-critical, embedded flight control software. The model-based paradigm allows to graphically develop software in block diagram format that is commonly used in the flight control domain. The advantages are: (i) design models serve as a universal communication mean between different stakeholders; (ii) software can be automatically generated from the design model; (iii) verification activities in the software development can start very early within the software lifecycle, as the design models are executable for dynamic testing and because they are well suited for design reviews.

MATLAB and Simulink are widely used tools in control engineering, simulation and in model-based software development. However, the use of MathWorks' Embedder Coder for code generation and its qualification according to RTCA DO-330 requires substantial additional engineering effort. On the other hand, Ansys SCADE Suite and its qualified code generator KCG can generate code with significant less effort. The project *CERTT-FBW23* has demonstrated an efficient development process for flight control software, in which control laws are designed in Simulink, the model is automatically translated into SCADE and code is subsequently generated by SCADE KCG in a qualified manner. This approach combines the advantages of both development environments [1].

After the translation of Simulink model into SCADE, it is essential to demonstrate that the two models are functionally equivalent and that any differences encountered are minor. Such differences can be reduced by developing basic blocks that only use atomic elements (plus, minus, etc.) and by replacing all blocks that are differently implemented in the Simulink and SCADE standard libraries. Furthermore, it has to be assured that the compilers either use identical mathematical libraries or that all needed mathematical functions are replaced by self-developed blocks. TU Berlin developed such model-element library that only uses elementary functions from the scope of the modelling language. In this way, nearly identical behaviour of the developed models in both modelling environments and on the target processor has been achieved.

In the projects *FCL-Methods* and *IBAS*, the experience from *CERTT-FBW23* has been used to establish a software development process according to RTCA DO-178C/DO-331 for Level C software. According to this process, flight control software for a high-altitude pseudo satellite (HAPS) airplane was developed. An important part of the projects was the development of the model-element library *"FCLib"* containing essential elements that are typically used in flight control software. It comprises filters, limiters, integrators, comparators and mathematical functions like sine, cosine, natural logarithm. The most important prerequisites for the development of the library are strict adherence of the defined software process and the use of a very limited scope of base operators (less or greater than, plus, minus etc.) in both modelling environments. The individual library modules in Simulink, SCADE and on the target were verified by test against the library requirements. The analysis of the test results showed that functional and numerical deviations were within expected and acceptable tolerances for an application in the flight control domain.

This paper describes the development process and the development of the library. Special attention is given to the specification of the library modules, design documentation, test specifications and testing of the example modules in Simulink, SCADE and on the target platform. An example for the development of library function is presented. Finally, the test results for selected modules are analysed and discussed.

References

[1] G. Walde and R. Luckner, "Bridging the tool gap for model-based design from flight control function design in Simulink to software design in SCADE," 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Sacramento, CA, USA, 2016, pp. 1-10, doi: 10.1109/DASC.2016.7778044.