Modeling of Avionics Systems using JGrafchart and Truetime

Anna Benktson and Sofia Dahlberg

September 7, 2012

Avionics today faces many challenges, as system complexity increases. One contributing factor is the transition to more electric aircrafts, which puts high demands on system reliance and performance. The drive behind moving towards more electric aircrafts is the reduced fuel burn and emissions, which is beneficial both from an environmental and cost perspective. The increased system complexity requires better modeling tools. This motivated an evaluation of two toolsets developed at Lund University, to investigate their suitability to avionic systems.

INTRODUCTION

As control system complexity increases the ability of evaluation at an early design stage is highly desirable which is why modeling tools are used to build an abstracted model of the physical system. One gain of modeling is that it gives an idea of how the physical system would behave before the actual implementation is done. From a cost perspective it is valuable to be able to analyze a system before it is a finished product, since adjustments grow more expensive with each development phase.

SEQUENTIAL CONTROL SYSTEMS

The electric system in a civil aircraft typically consists of power generation, primary and secondary power distribution and protections. The routing of power in the primary power distribution system is performed in a sequentialized manner. Since redundancy is key in avionic systems, there typically exist many paths and combinations for the controller to consider and choose from. This makes sequential control systems difficult to design.

JGRAFCHART

JGrafchart is a mathematical modeling tool for sequential control applications. It features concepts from object-oriented programming languages and good structuring capabilities for hierarchical modeling. The aim of JGrafchart is to provide a high-level language for control applications.

APPLICATION OF JGRAFCHART

The reason for applying JGrafchart to the sequential parts of the primary power distribution system was to investigate how the object-oriented features and structuring capabilities could be used to create a generic and hierarchical model of the system.

The object oriented features in JGrafchart made it possible to model each physical component in the power distribution system as an object. Also, the structuring capabilities enabled separation of the nominal control from the fault handling.

INTEGRATED MODULAR AVIONIC (IMA) SYSTEMS

Integrated Modular Avionics (IMA) is a term used for distributed computer networks in aircrafts and a concept describing how to integrate distributed modules. In the aircraft industry today there is an increasing drive towards IMA. One reason for this is that it replaces the point-to-point networking with virtual links and thus reduces the required wiring. This is an important improvement since the wires together with isolation are the main contributor to aircraft weight. Furthermore the weight of the aircraft is a key point of cost. Another important reason for using IMA is that it increases system reliance since the network protocol used is deterministic and redundant.

In distributed computer networks it is not unusual for actuators, sensors and controllers to be on different networks with different protocols. In order to understand the effect of delay and jitter that might occur due to distribution, simulation tools are needed. Simulation tools can be used to investigate the constraints and possibilities of the system, with respect
to the control algorithm as well as the software design. Delay and jitter might lead to poor control performance, and therefore it is important to have an idea about how much interference the system can handle before becoming unstable.

**TRUETIME**

TrueTime is a simulation tool based on Matlab/Simulink used for real-time control systems. Its main purpose is to analyze the impact of latency on control performance in networked control applications. TrueTime features simulation of different network protocols as well as multi-tasking real time kernels.

**APPLICATION OF TRUETIME**

When modeling distributed systems, TrueTime can be used to build an abstracted model which still contains all critical elements in the physical system. By simulation of the abstracted model, it is possible to analyze the impact of delay and jitter on control performance. Furthermore, the results can be used to help drive requirements which are non-trivial to assess already at the design stage.

In order to apply TrueTime to IMA systems, support for the network protocols used in avionics was needed. This motivated the implementation of Avionics Full Duplex Switched Ethernet (AFDX). AFDX is basically a full duplex switched Ethernet, except that it is deterministic and redundant. TrueTime was then applied to a simplification of a typical IMA system.

**CONCLUSIONS AND FUTURE WORK**

This section briefly summarizes the conclusions drawn from applying JGrafchart and TrueTime to avionic systems.

By making use of the object oriented features in JGrafchart an intuitive design is easy to achieve. Also, the structuring capabilities in JGrafchart decrease the probability of programming errors and ease implementation of complex applications.

However, the tool needs to be further developed to better handle large control applications. It might be possible to apply JGrafchart early in the development phase to build an abstracted model, which can be used early on for analysis using model checking.

Possible extensions for JGrafchart would be to implement support for a high level version with full object orientation. This could rid the model of redundant code and improve readability. Future work could be to apply JGrafchart to several different aircrafts with slightly different electric system topology in order to build a more generic model.

TrueTime appears to be feasible to avionic systems since the tool can be used as it is today. The impression is that TrueTime can be used to drive requirements and provide value in terms of assessing what reasonable limits for requirements could be already at a design stage, which is non-trivial to do.

A next step would be to further validate this work by applying TrueTime and the AFDX implementation to a real system and evaluate the outcome.