

## Um Estudo de Software simulador AOCS para CubeSats

CAMARGO, L.<sup>1,2</sup>, LANDIM, D.<sup>3</sup>, ESSADO, M.<sup>4</sup>, LIMA, J.<sup>1</sup>, GOMES, C. <sup>1</sup>,  
MATTIELLO-FRANCISCO, F. <sup>1</sup>, DURÃO, O.<sup>1</sup>

<sup>1</sup>Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brasil

<sup>2</sup>Aluna de Doutorado do curso de Engenharia e Gerenciamento de Sistemas Espaciais - CSE.

<sup>3</sup>Universidade Federal de São Paulo, São José dos Campos, SP, Brasil

<sup>4</sup>EMSISTI Sistemas Espaciais & Tecnologia, São José dos Campos, SP, Brasil

lazarocamargo@inpe.br

**Resumo.** *Este artigo apresenta o estudo realizado sobre um software de simulação de atitude, determinação e controle de órbita para Cubesats. O trabalho foi desenvolvido como parte das atividades do Curso de Inverno ETE/INPE 2017 durante o mini estágio prático, no contexto do Programa NanosatC-BR – Desenvolvimento de Cubesats. Para o desenvolvimento e aprofundamento do tema na área de Sistemas Espaciais, buscou-se cumprir as atividades previstas no plano de estágio: pesquisa, análise, execução e documentação das práticas realizadas. São apresentados os aspectos relacionados a modelagem, parâmetros, fluxo de operação e funcionalidades do software ISIS AODCS.*

**Palavras-chave:** AOCS; Determinação de órbita; Controle de Atitude; Cubesat; Simulação.

### 1. Introdução

Este artigo é resultado das atividades desenvolvidas no mini estágio técnico realizado durante Curso de Inverno em ETE - Engenharia e Tecnologia Espaciais do INPE - Instituto Nacional de Pesquisas Espaciais, no ano de 2017, que tem como objetivo oferecer aos alunos de graduação algumas noções básicas de Engenharia e Tecnologia Espaciais, em especial, Engenharia de Sistemas, projeto, montagem, testes e operação de satélites.

O curso de inverno possui duração de três semanas, com apresentação de palestras nas áreas de computação, ciência espacial e observação da terra e aplicação prática oferecida através da realização de mini estágios. Ainda, possibilita a realização de visitas técnicas ao Laboratório de Integração e Testes -LIT, Centro de Controle de Satélites (CCS) e à Divisão de Eletrônica Aeroespacial (DEA). O estudo realizado sobre um software de simulação de atitude, determinação e controle de órbita para plataformas de satélite do tipo Cubesat foi desenvolvido no LabSim – Laboratório de Simulação do INPE, no âmbito do Programa NanosatC-BR-Desenvolvimento de Cubesats. São apresentados os aspectos relacionados a modelagem, parâmetros, fluxo de operação e funcionalidades do software ISIS AODCS.

### 2. Metodologia

No desenvolvimento do trabalho buscou-se cumprir todas as atividades previstas no plano de estágio: pesquisa, análise, execução e documentação das práticas realizadas.

Porém, algumas das atividades não puderam ser totalmente executadas devido a restrições de tempo.

As atividades consistiram em:

- Modelar componentes do simulador por meio de diagramas UML;
- Configurar o ambiente de simulação;
- Executar simulação com variação de parâmetros;
- Elaborar relatório técnico das atividades.

As ferramentas usadas durante o estágio são descritas a seguir:

- Manual fornecido pelo desenvolvedor - ISIS.I-AODCS.UM.001\_Solar Simulator User Manual\_v1.0.pdf
- Software Simulador AOCS em C/C++ para plataforma de satélite 1U, 2U e 3U.

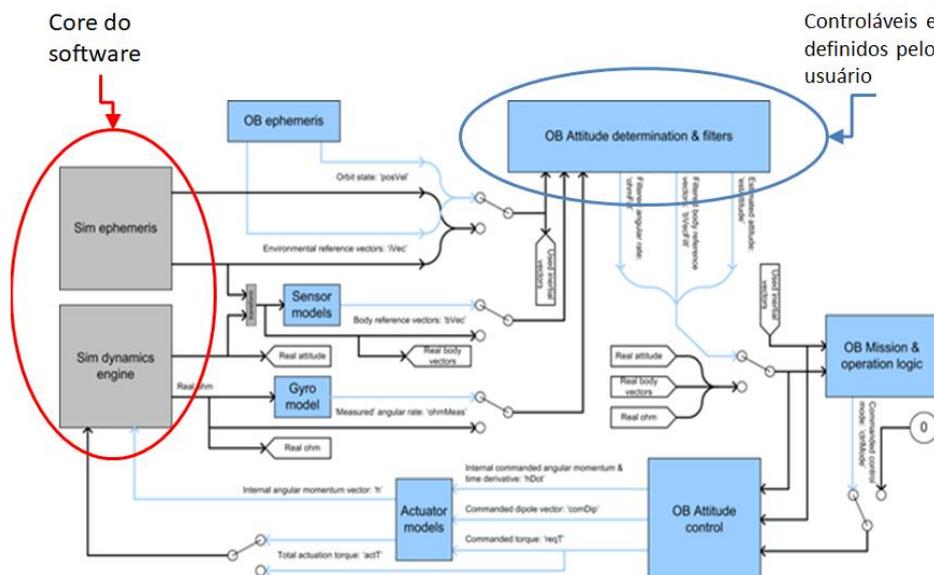
### 3. Software de Simulação AOCS

O Software ISIS - Attitude & Orbit Determination and Control Simulator (I - IADOCS) é um framework para simulações de controle de órbita e atitude.

- Entrada: Especificações do projeto;
- Processamento: Funções definidas pelo usuário;
- Saída: Gráficos, Tabelas, Animações.

O diagrama de funcionalidades do software é apresentado na Figura 01.

Figura 01. Diagrama de Funcionalidades do Software.

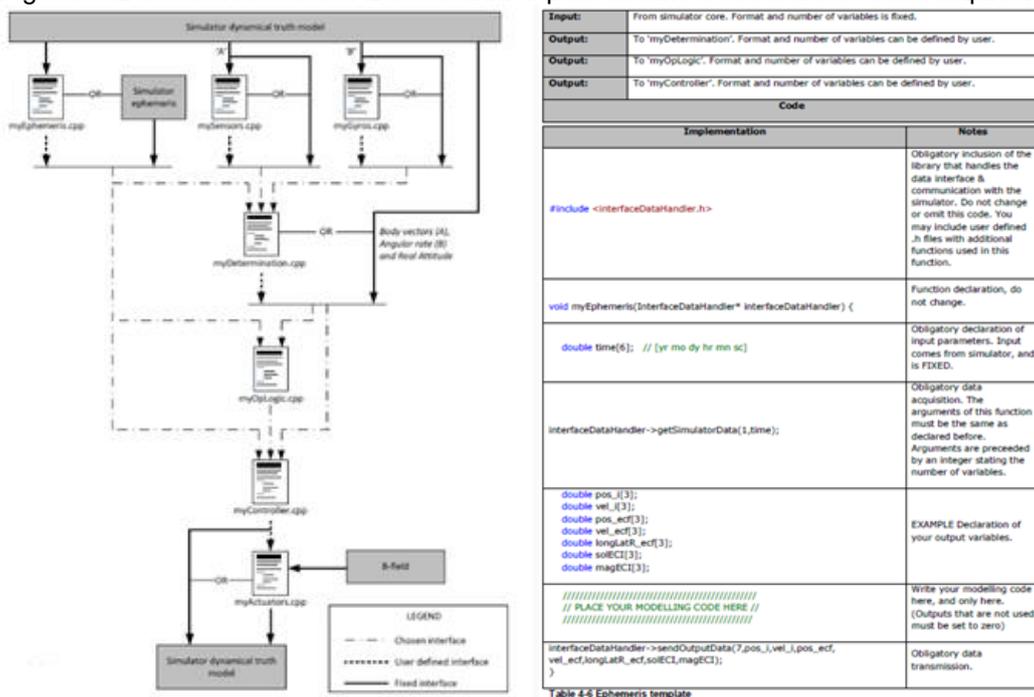


Fonte: Adaptado do Manual - ISIS.I-AODCS.UM.001\_Solar Simulator User Manual

O software permite ao usuário implementar seu próprio código em C/C++. Tornando possível modelar e desenhar a arquitetura ADCS própria de seu hardware, implementar biblioteca de funções. Existem 7 categorias de arquivos em C/C++ de código que podem ser definidos pelo usuário, não obrigatoriamente todos devem ser implementados, eles são independentes, myEphemeris.cpp, mySensors.cpp, myGyros.cpp, myDetermination.cpp, myOplogic.cpp, myController.cpp, myActuators. É importante observar que apesar de não ser obrigatório implementar todos os códigos para o processamento de dados, alguns códigos são dependentes, e realizam atividades de processamento de dados que geram valores de entrada para os outros arquivos. A Figura 02 exemplifica como os modelos ADCS podem ser definidos pelo usuário.

A estrutura de cada arquivo definido pelo usuário, conforme Figura 3, obedece uma rotina de processamento de dados com parâmetros de entrada e saída pré-definidos e imutáveis.

Figuras 02 e 03. Modelos de ADCS definidos pelo usuário e estrutura de cada arquivo.

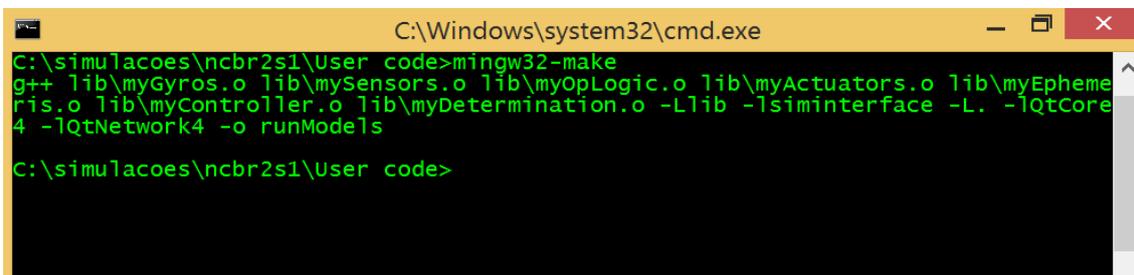


Fonte: Manual - ISIS.I-AODCS.UM.001\_Solar Simulator User Manual

Neste exemplo o arquivo myEphemeris.cpp recebe de entrada valores do próprio software que podem ser ajustados no ambiente gráfico (parâmetros de órbita Kepler, TLE, numéricos). O código executa uma rotina de processamento de acordo com o modelo definido pelo usuário no campo identificado. As variáveis de saída não podem ser alteradas, e o código retorna uma saída padrão conforme última linha definida no modelo. Esta saída serve de input aos outros modelos, conforme diagrama apresentado na Figura 2.

Após editados todos os arquivos de modelo de usuário, um arquivo único chamado “Makefile” deve ser compilado no diretório raiz onde todos os arquivos de modelo de usuário devem-se encontrar. Este arquivo Makefile irá reunir todas as funções dos arquivos de modelo de usuário e irá gerar um executável denominado runModels.exe. O executável fica em processo de loop, aguardando uma chamada do framework para realizar o cálculo das simulações e devolver os valores processados ao framework. A Figura 04 mostra a compilação dos arquivos de modelo do usuário e geração do runModels.exe.

Figura 04. Compilação dos arquivos de modelo do usuário e geração do runModels.exe.

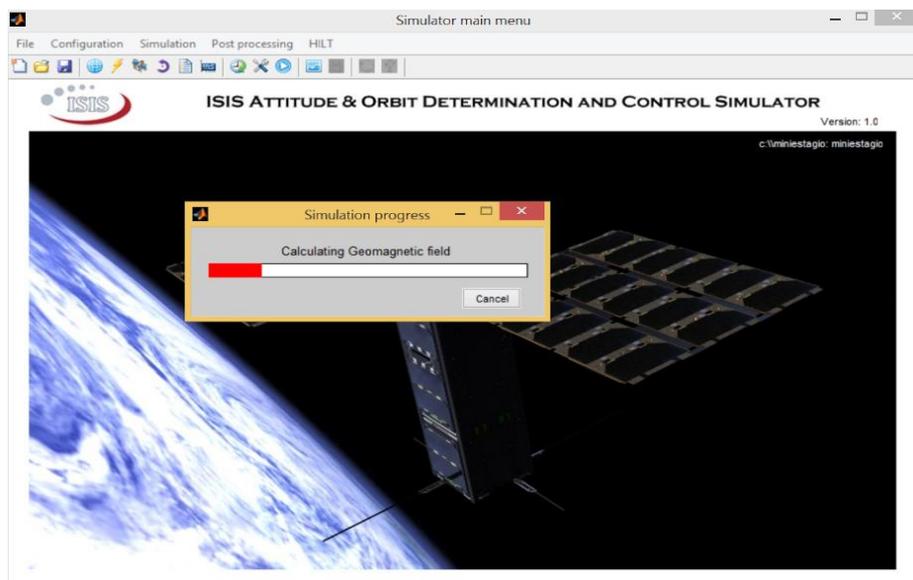


```
C:\Windows\system32\cmd.exe
C:\simulacoes\ncbr2s1\User code>mingw32-make
g++ lib\myGyros.o lib\mySensors.o lib\myOpLogic.o lib\myActuators.o lib\myEphem
ris.o lib\myController.o lib\myDetermination.o -Llib -lsiminterface -L. -lQtCore
4 -lQtNetwork4 -o runModels
C:\simulacoes\ncbr2s1\User code>
```

Fonte: Autor com Software Simulador AOCS em C/C++ para Cubesats.

Na Figura 5 é pode-se ter um exemplo do processamento dos dados de simulação realizado por meio do runModels.exe

Figura 05. Processamento dos Dados de Simulação.

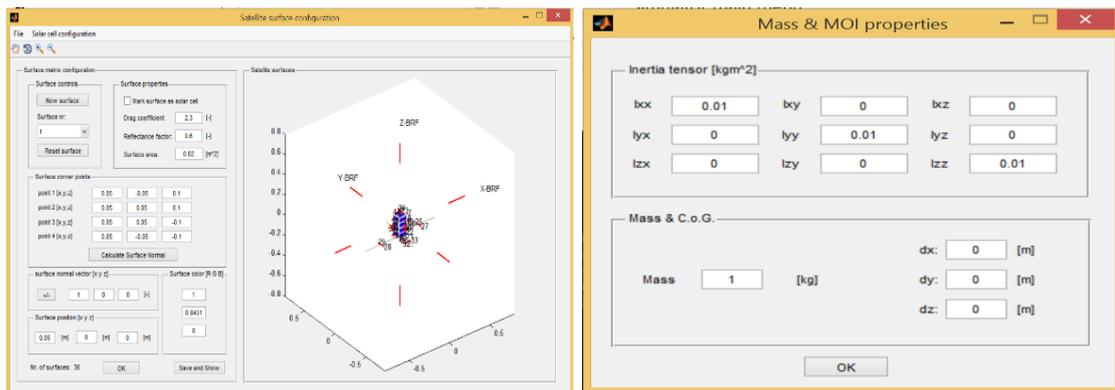


Fonte: Autor com Software Simulador AOCS em C/C++ para Cubesats.

#### 4. Lições Aprendidas

Com as definições dos modelos de usuário, por fim é possível ajustar todos os parâmetros que serão dados ao core do programa para realizar os processamentos de dados de simulação. Várias funções em ambiente gráfico permitem o ajuste de valores de parâmetros. As Figuras 06 e 07 apresentam alguns exemplos.

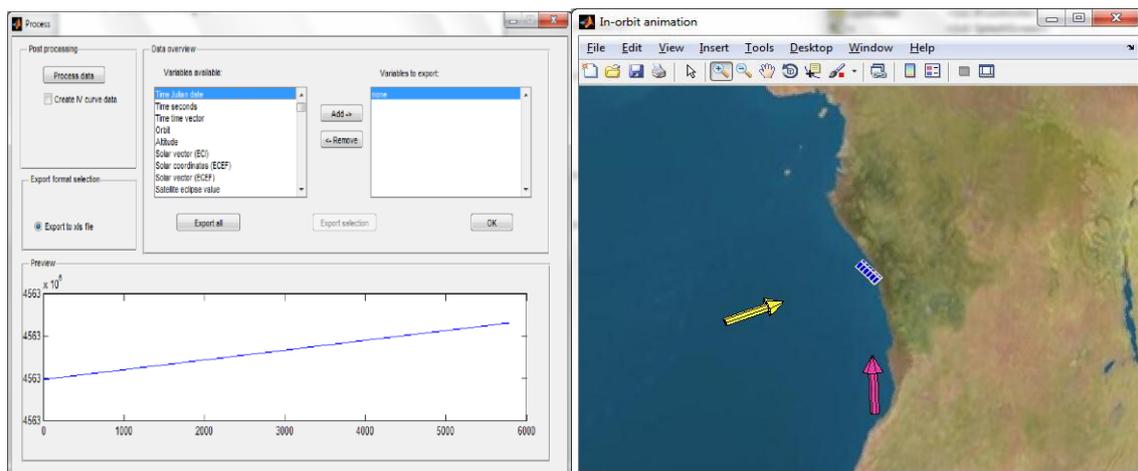
Figuras 06 e 07. Configurações de superfície do painel solar e propriedade de massa do satélite.



Fonte: Autor com Software Simulador AOCS em C/C++ para Cubesats.

Na Figura 08 observa-se o padrão de saída de dados do simulador, onde pode-se selecionar os valores simulados desejáveis e obtê-los na forma gráfica ou exportada em tabulação através de um arquivo csv. Após o processamento destes dados é possível realizar animações da simulação de órbita e controle de atitude do satélite, conforme Figura 09.

Figuras 08 e 09. Saída de dados do simulador gráfica e animação.



Fonte: Autor com Software Simulador AOCS em C/C++ para Cubesats.

As lições aprendidas no período de estágio serão úteis e aplicadas às próximas fases de desenvolvimento do satélite NanosatC-Br2.

#### 4. Conclusões

Este trabalho apresenta os aspectos relacionados a modelagem, parâmetros, fluxo de operação e funcionalidades do software ISIS AODCS.

É possível, nos modelos de definições do usuário utilizar a ferramenta conforme especificações de cada projeto. Ainda, utilizar diversos parâmetros pré-definidos de órbita e controle de atitude criando diversos cenários de simulação variando esses parâmetros.

A flexibilidade do código para modelagem dos parâmetros de órbita, atuadores, sensores e outros permite implementar funções e modelos matemáticos de forma objetiva.

O desenvolvimento deste trabalho foi importante como exercício de aprendizado durante o Curso de Inverno/ETE do INPE.

*Agradecimentos: Curso de Inverno ETE/INPE e EMSISTI Sistemas Espaciais & Tecnologia, Capes, CNPq, Programa NanosatC-BR.*

#### Referências

Manual do fabricante, ISIS.I-AODCS.UM.001\_Solar Simulator User Manual\_v1.0.pdf