Master Thesis

Simulation and control toolkit for small satellite projects

(Programovy balik pro simulaci a navrh rizeni malych satelitu)

Adam Śmiałek

February 2020

Contents

Abbreviations

1 Introduction

Here goes short introduction with mention that this is a work about Spacecraft Control Architecture Rapid Simulator (SCARS)

1.1 Scope

The thesis covers the process of development of the toolbox for rapid prototyping of satellite's control systems. **??** describes the aim of this work and discusses the topic of prototyping tools. **??** goes into detail about the architecture of Spacecraft Control Architecture Rapid Simulator, its features and methods of implementation. Also here there are described the ways of connecting Spacecraft Control Architecture Rapid Simulator with various visualization tools. **??** explains the documentation and usage of Spacecraft Control Architecture Rapid Simulator, while in **??** there are examples showing how the toolbox can be used in real life applications. Finally, **??** discusses the conclusions from the development process and the possibilities for improvements of Spacecraft Control Architecture Rapid Simulator.

1.2 Aim

The process of effective space-related project management, from the conception of the idea, through production, to disposal, features high costs and often various unpredictable risks. Due to this, a project life cycle is usually divided into distinct phases, allowing for introduction of conducting product reviews within rigid timeframes. An example of such a workflow, adopted by most major agencies such as ESA[?] and NASA[?], is a division of the project life cycle into phases, as it can be seen on **??**. While the design of a project is often an iterative process, the phases and reviews that conclude them exist as a checkpoints, after which the design of the project is to be unchanged, on a level of details progressing as phases do. For example, as one can see, Phase B is usually ended by the Preliminary Design Review. In the case of a spacecraft, for the Preliminary Design Review, a major architecture parameters have to be defined, such as volume and weight ramifications, top-level designs of solutions for major requirements have to be presented - for a practical example: for high-resolution Earth observation mission the type of the actuators which fulfills precision requirements has to be chosen.

The aim of this thesis work is to build and provide a finished and ready to use open source product - a toolbox for small and low budget satellite

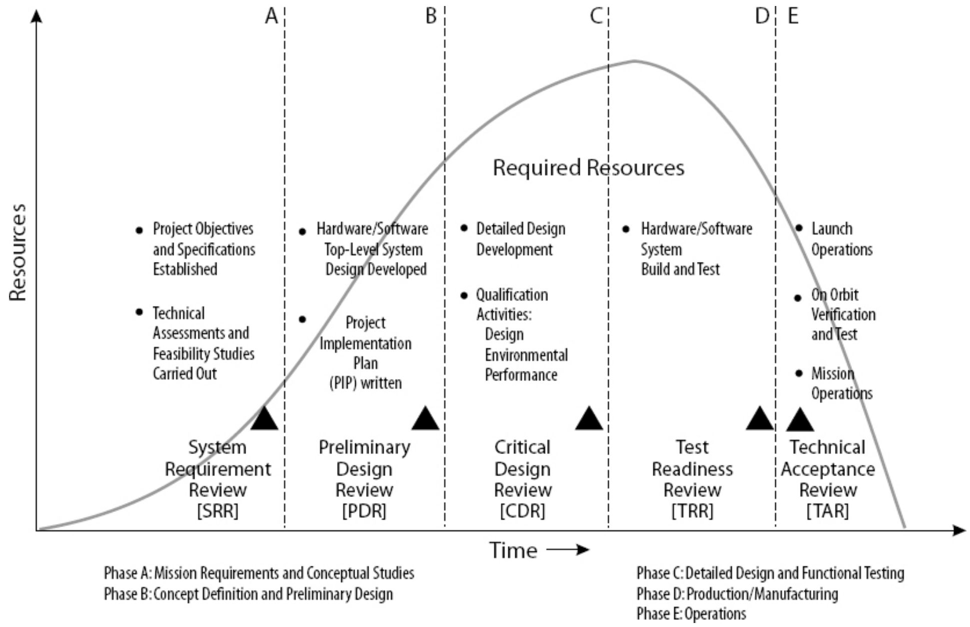

Figure 1.1: Typical space project phases and its life cycle[?]

projects. The toolbox features allow for a initial design of spacecraft's Attitude Determination and Control System, which means that they allow for, i.a. simulation of spacecraft orbit, testing the feasibility of various actuation methods and testing of feasibility of different control algorithms. That software would then allow smaller and inexperienced teams of spacecraft designers to better prepare for design milestones like Preliminary Design Review, when there is not enough time to create a full simulation of their spacecraft Attitude Determination and Control System subsystems.

refine the list of examples this

While the toolbox is by itself a tool for practial use, the thesis also serves as as a review of available solutions, so it can be used by future control engineers as a learning material.

also, add some lines about how it can be used for learning in general (like some university courses)

put here an example from my experience with BEXUS

1.3 Prototyping tools

refer to CAE software

1.4 Small spacecraft simulations

1.5 Already existing tools

... This is to show that, while the toolbox for Attitude Determination and Control System prototyping or development is not an innovative idea, but since some existing solutions are not best fit for the job of rapid prototyping and others are unavailable or paid, there is a need for a new, open-source one.

Should I put "advantages" and "disadvantages" here at the end of every point?

1.5.1 MATLAB CubeSat Simulation Library

how to write that it was considered to expand this rather than develop own toolbox

CubeSat Simulation Library is a part of Aerospace Blocks created by MathWorks Aerospace Products Team. Using it one can model motion and dynamics of CubeSats and nano satellites. It provides the most basic features, like the simulation of pre-set attitude scenarios, basic actuators and sensors models and integration with MATLAB's Virtual World visualization tools.

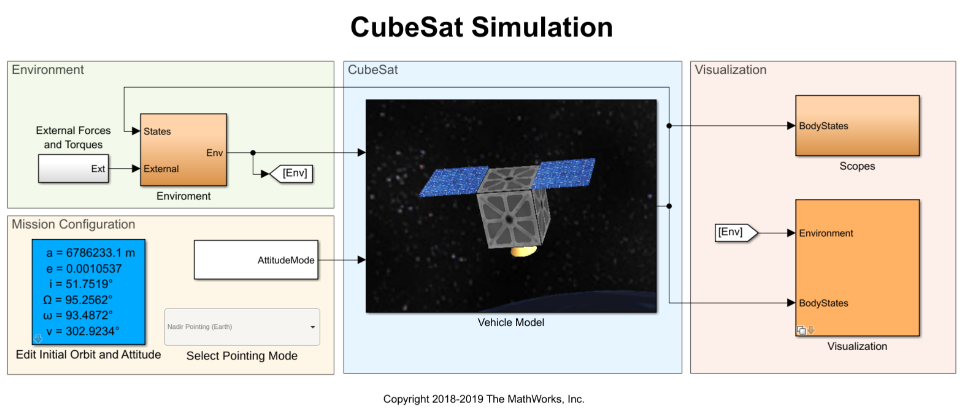

Figure 1.2: Top-level view of the example project of the MATLAB CubeSat Simulation Library

This library, while conceptually most similar to the Spacecraft Control Architecture Rapid Simulator, it lacks some functionalities. For example, for actuators, it provides only general models for perfect and second-order actuators. In Spacecraft Control Architecture Rapid Simulator, the actuators are full models, which allows not only for reducing the number of layers of abstraction between the user and the simulation, but also for things like calculation of energy expended by the actuator. Also, this toolbox is sparsely documented - while most functionalities are described within their Simulink block masks, there is no comprehensive guide about how to use them in own models.

does this explanation belong here?

1.5.2 PrincetonSATELLITE Spacecraft Control Toolbox

PrincetonSATELLITE Spacecraft Control Toolbox is a commercial solution for building spacecraft Simulations. I contains over two thousand functions for attitude and orbit dynamics, simulation, estimation, analysis and design. This is the most robust and comprehensive toolbox available, includes online API, well written documentation and additional modules for unique applications like formation flying, fusion propulsion or solar sails. Yet this is a paid solution and even the cheapest option - CubeSat Edition - may be out of price range for smaller teams.

1.5.3 PROPAT Toolbox

PROPAT is is a small set of functions in Matlab to simulate and propagate orbit and attitude of an Earth's satellite, developed by the single person as an open-source toolbox. Several functions allow to transform between orbit and attitude coordinates and for propagation or rigid body attitude. PROPAT contains only MATLAB scripts, which while useful and can be used as a part of the simulation, do not combine into a model of a whole spacecraft's Attitude Determination and Control System subsystem.

1.5.4 GAST Toolbox

The GAST toolbox is the result of the consolidation of several toolboxes available in Guidance, Navigation, and Control Systems Section of European Space Research and Technology Centre, such as the old AOCS Toolbox, the SpaceLAB library, the ViSiLib library, the ATPE simulator, and the PAV simulator. In addition to consolidating these toolboxes, new models were developed for the GAST toolbox according to the needs of the section. **??** shows a pictorial representation of the consolidation of the toolboxes of TEC-ECN. This software was developed in Guidance, Navigation, and Control Systems Section in 2008, but since it is a product of European Space Agency, it is not available for use for wider audience.

1.6 Modules available on MathWorks MATLABCentral

describe these modules in two sentences

1.6.1 SAT-LAB

https://www.mathworks.com/matlabcentral/fileexchange/63344-sat-lab-a-matlab-graphical-user-interface-for-simulating-and-visualizing-keplerian-satellite-orbits

1.6.2 Satellite Orbit Modeling

https://www.mathworks.com/matlabcentral/fileexchange/54877-satellite-orbit-modeling

1.6.3 Satellite Orbits: Models, Methods and Applications

https://www.mathworks.com/matlabcentral/fileexchange/54840-satellite-orbits-models-methods-and-applications

1.6.4 Smart Nanosatellite Attitude Propagator (SNAP)

https://www.mathworks.com/matlabcentral/fileexchange/68652-smart-nanosatellite-attitude-propagator-snap

Also, write somewhere that while the Toolboxes/Simulations like these are present within papers online, they are nowhere to be found to download and it is hard and time-consuming to recreate them from papers.

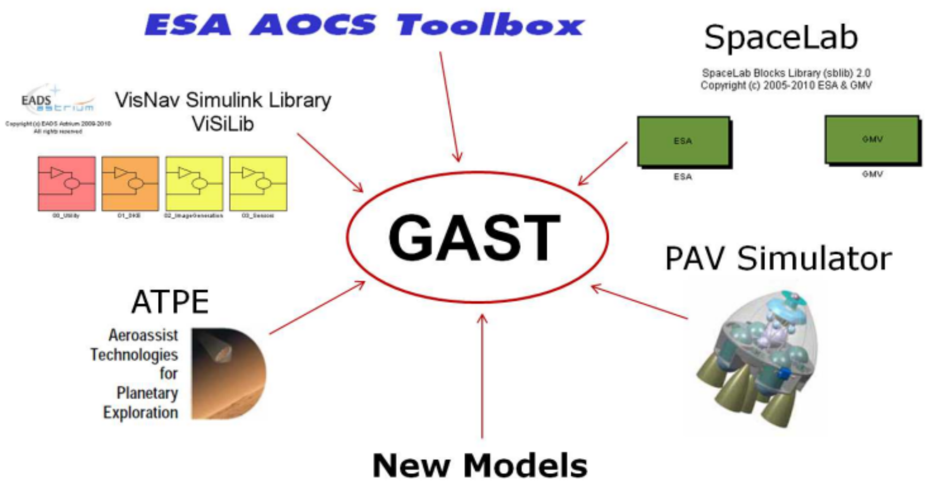

Figure 1.3: Representation of the consolidation of TEC-ECN toolboxes

2 Spacecraft Control Architecture Rapid Simulator (SCARS)

short description of the chapter

something about the Simulink Add-Ons that are used by the toolbox

describe briefly what is the input and the output of the whole toolbox

2.1 Features

2.2 Architecture

Spacecraft Control Architecture Rapid Simulator is divided into two parts: 1) Parts Library and 2) Modular Simulation. The Parts Library contains Simulink subsystems, which can be connected to form simulations of various complexity and for multiple scenarios. The other is a Modular Simulation, which can be set up with either MATLAB command line scripts or graphical user interface.

library screenshot

2.2.1 Parts Library

2.2.2 Modular Simulation

2.3 Orbit dynamics

2.4 Environment

Environment block is responsible for outputting few parameters, which are then used by the Satellite subsystem

- Gravity model
- Partial atmosphere?
- Sun's relative position and Earth's shadow
- Magnetic model

2.4.1 Frames of Reference

Used: ECEF, NED
Transformation from Keplerian Elemenets to Kartesian Coordinates

2.4.2 Earths's Gravity Model

Simulink Aerospace Blockset **Spherical Harmonic Gravity Model**, EGM2008 Planet's Model

Citation for orbital elements:[?]

The body is assumed to be rigid?

2.5 Actuators

2.5.1 Thrusters

https://www.cubesatshop.com/wp-content/uploads/2017/04/ENP-IFM-Nano-Thruster-Product-Overview.pdf https://blog.satsearch.co/2019-07-10-cubesat-thrusters-an-overview-of-in-space-propulsion-products-for-small-satellites

Parameters: Thrust range; Nominal thrust: (find a way to model change?) Specific impulse (also ranges?) Max propellant Total impulse Power (at nominal thrust?) Mass Dimensions? Hot standby Power Time delay to control In both directions?

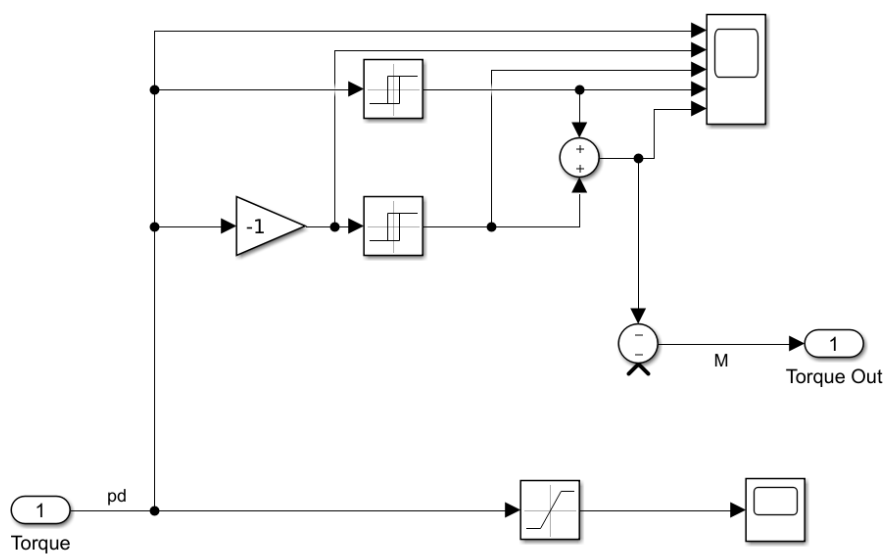

Figure 2.1: Simulink Thrusters model

Thrusters parameters:

2.5.2 Reaction Wheels

Ideal to real: Bearing Noise, Transport Delay, Saturation, Quantization.

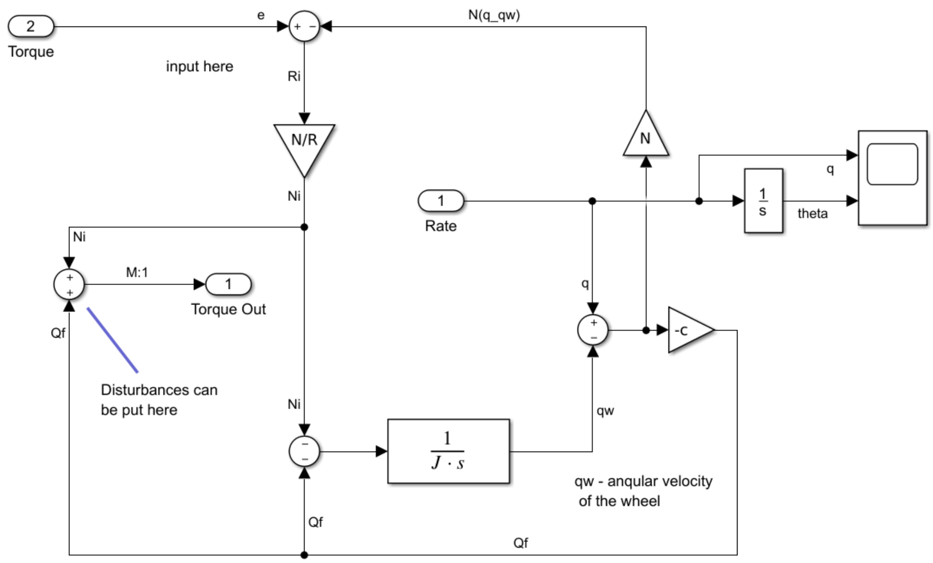

Figure 2.2: Simulink Reaction Wheels model

2.5.3 Gimbaled Momentum Wheel

2.6 Sensors

2.6.1 GPS

Proporties: update rate reference location horizonstal position accuracy vertical position accuracy velocity accuracy decay factor ? failure rate?

2.6.2 Infra-Red, Optical and Radar Sensors

2.6.3 accelerometer

accelerometer misalignment

2.6.4 Gyros

Parameters: change over temperature zero rate level change over temperature [dps/*C] (but this can biased to get rid of the error, so is not necessary to model) senistivity Measurement range Angle Random Walk (same Parameter as FFT and Power Spectral Density)

2.6.5 Inertial Measurement Units

2.7 Control Methods

2.7.1 PID

2.7.2 LQR

2.7.3 Quaternion Feedback Control

Look into the topic

2.7.4 Analisys Tools

2.8 Visualization Tools

2.8.1 MATLAB Virtual World

2.8.2 FlightGear

2.8.3 Orbiter

2.8.4 Kerbal Space Program

3 Documentation of SCARS

3.1 MATLAB Masks and help

3.2 Website?

4 Example of usage

4.1 Setting up spacecraft ADCS architecture in SCARS

4.1.1 Basic made up spacecraft

4.1.2 PW-Sat2

Magnetorquers
Two modes of control: Detumbling Control Mode; Sun Pointing Mode
https://pw-sat.pl/wp-content/uploads/2014/07/PW-Sat2-C-01.00-ADCS-CDR.pdf

4.2 Examples of tests possible with SCARS

4.2.1 Control system robusntess analisys

4.2.2 Long term simulation

4.2.3 Contingency scenarios

5 Conclusions

Notes

- Aims:
  - To provide a toolbox/rapid prototyping software
  - For use for smaller projects, like CubeSat teams, without access to commercial tools
  - For use for first Feasibility Study, or testing and comparison of different acutators
  - For learning purposes (i.e. analogue to FCS course)
- Add database of example/know actuators/sensors
- Parameters of the block (for example: power consumption)
- Add methods of analisys of the control systems, ie. bode, pole plots
- Methods for optimization
- Already existing tools:
  - MATLAB CubeSat Simulation Library
  - PrincetonSATELLITE Spacecraft Control Toolbox
  - PROPAT Toolbox
  - GAST Toolbox
- What would differ SCARS from these tools:
  - Ease of use
  - Limited to only certain cases
  - Open source
  - Good documentation

Timeline

- May 17: All actuators and sensors models included in Simulation
- May 24: Control algorithm implementation and tuning
- May 31: GUI for setting up and launching the Simulation
- **June 07: Finishing Simulation and additional features** (e.g. visualisation, detailed environment)
- June 21: Text of the thesis in 3/5, changes and additions to the toolbox, research on public forums
- June 28: Finish writing the thesis
- **Onward from June 29: Time for corrections**
- July 14: Finish of database collection for element picker
- **July 31?: Subsmission deadline**

Actuators Checklist

- Reaction wheels touchup and parameters
- Allow for thrusters mode change from bang-bang to controlable
- Add this damned solar sail maybe?
- Add the magnetometers with parameters
- Add some sensors with parameters