Czech Technical University
Faculty of Electrical Engineering
Department of Control Engineering

CTU Diploma Project review- 2nd reviewer's evaluation of master thesis with title "S imulation of Attitude and Orbit Control for APEX CubeSat" by Space Master student Niels de Graaf.

I find that the goal of the thesis project fulfills the requirements of a master thesis in space technology. The work concerns implementation of an open source, low cost demonstrator platform to be used for developing and testing onboard software on physical hardware. In this thesis the focus is on the AOCS and the system bus communication (CAN bus).
A simple microcomputer system including a CAN-bus and two microcomputers communicating with each other (and a PC) is set up, but the main part of the project concerns writing software modules in python for the orbital simulation and the communication over the CAN-bus.
The student is using available python libraries for orbital propagation, low level CAN communication, serial communication and plotting, and is adding code for perturbations (presented in Ch.2).

The design of the Hardware-Software testing facility are shown in the UML tree in Figure 3.2. The system is de-composed into two components, which in turn is aggregated into more components. I find the decomposition (the levels and objects in the tree) indicated by this figure relevant, but I do not understand the choice of aggregation for the "leaves" in the tree, where direct links exists between the objects in the Hardware simulator and the orbital simulator (I assume that the arrows represents associations (links) between the objects). Moreover, I cannot see that this design is shown in the final implementation of the system.

The main concern regarding the project is the software design and implementation. A very important aspect of software design is to clearly define the responsibilities of different parts of the software and the interfaces.   General guidelines for software design includes decomposition and modularization, encapsulation/information hiding, and separation of concerns.
Ths means that the complete implementation shall be divided into modules (functions, components, objects etc) with clear responsibilities (functionality) and well defined interfaces. This is not the case for the software implemented. Also, literals and constants (such as orbital elements, satellite parameters, chosen perturbation etc) are spread throughout the code; this should be gathered in a configuration, or system initiation file.
One example is in the orbit_data_sim.py code, where the perturbation algorithm is hard-coded into the code, and the density etc is given as numbers in the code. This means that if parameters are changed, the user needs to search for all places in the code where parameters are set, which is error prone. It is also advisable to gather information regarding the asteroid etc into one structure, instead of using many variables.
It is also of importance to document the code, using file headers and extensive commenting. When equations are used, it is good practice to include references to literature in the code.

Regarding the thesis format, small editing, language and spelling mistakes are present in the thesis.

Summary:
The student has put in a sufficient effort into the task.

The main objectives was to create a model of the dynamics of the APEX CubeSat subjected to the disturbance for the asteroid environment to plot its trajectories using python libraries developed for orbital dynamics. Attitude and Orbital control applications should be programmed on a microcontroller to calculate the new trajectories for the simulation. An Interface should be made to send simulated data using the standard protocols developed for spacecrafts and using the communication bus used on the APEX spacecraft. The result of the thesis fulfills the first objective, and the second objective concerning basic orbital control. The third objective is not fulfilled (using standard protocols), only low level, basic communication test routines using CAN-messages are implemented.

The result of the thesis project may contribute to a future solution to the problem addressed, since a basic, working testing environment for further development is set up. But, the software needs to be re-designed before developing the system any further.

Based on the review above I recommend to grade the thesis by C(good). The oral presentation is still to be graded.

This review serves solely for the purposes of the diploma project defense at CTU. LTU official evaluation for the SpaceMaster double degree will follow the thesis defense and may differ from this review report and suggested grade.

Kiruna, September 25 2020

Dr. Anita Enmark
Luleå University of Technology