DESIGN AUTOMATION OF EMBEDDED AIR COILS FOR CUBESAT ATTITUDE CONTROL

Sebastian Grau[(1)], José Manuel Diez Lopez[(1)], Debdeep Roychowdhury[(1)], Jakub Chachowski[(2)], Enrico Stoll[(1)]

[(1)] *Technische Universität Berlin, Chair of Space Technology, Marchstraße 12-14, 10587 Berlin, Germany, +49 30 314-23138, sebastian.grau@tu-berlin.de*

[(2)] *AGH University of Science and Technology, Space Technology Centre, Mickiewicza 30 Ave., 30-054 Kraków, Poland, +48 794 602 537, jchachowski@student.agh.edu.pl*

PAPER

Embedded air coils are magnetic actuators commonly used for detumbling, coarse attitude control, and angular momentum dumping in the side panels of nanosatellites. Side panels feature complex contours, pockets, through-hole connectors, vias, and keep-out areas thus making it difficult to design optimized coil geometries into them. In this work we present the automated magnetorquer design developed in the scopr of the NanoFF mission at TU Berlin. The mission aims to demonstrate formation flight with two 2U CubeSats featuring a propulsion system. The subsystems of the spacecraft negatively influence the available board area on the side panels and all side panels are different. Kitorquer, a KiCAD action plugin, detects the side panel geometry and generates an optimized coil given a set of user-defined constraints. We give a quick introduction into magnetorquer optimization, the plugin implementation, resulting coil geometries, and plugin validation based on the side panels manufactured for the NanoFF mission. We provide an outlook on the application of the plugin for the design of PocketQube side panels carried out by the authors for the HYPE mission of AGH University, Poland which has strict constraints on available board area and power budget.

1 INTRODUCTION

Magnetic control actuators are considered the baseline attitude control actuators for pico- and nanosatellites. The three main types of magnetic control actuators for such small satellites are magnetorquers (MTQs), wound air coils (WACs), and embedded air coils (EACs). Grau, Suchantke, and Brieß provide a good overview of the three different types and the optimization of their design in [1]. The optimization of EACs assumes that a rectangular area on the printed circuit board (PCB) that is the basis for an integrated CubeSat side panel is available for the optimized actuator. But to mount the panels on the satellite primary structure, to mount mechanical components on the side panel, or to connect between layers on the PCB a number of holes through the PCB are required.

The two satellites of the Technische Universität Berlin (TU Berlin) Nanosatellites in Formation Flight (NanoFF) mission are highly integrated 2 U CubeSats intended to demonstrate close-range formation flight using a resistojet propulsion system. To keep the thrust vector pointing in the required direction the attitude and orbit control system (AOCS) of the spacecraft features a redundant reaction wheel assembly (RWA) and a total of six magnetic control actuators that are grouped in three pairs of coils along each of the body axes. While the coils along the x body axis are implemented as WACs, the coils along the y and z body axes are embedded into the spacecraft's side panels.

Due to the joints of the deployable solar panels and the four-channel earth-observation payload the four side panels feature distinct cut-outs. These cut-outs make it impossible to reuse the layout of the coil used on one side panel on any of the other side panels. The cut-outs make it difficult to find a large coherent area to implement rectangular EACs. However, to efficiently detumble the spacecraft after deployment and to desaturate the reaction wheels (RWs) a comparatively large magnetic dipole moment needs to be provided by the magnetic actuators.

The limited coherent rectangular area available on the side panels makes it necessary to develop an algorithm that calculates an optimal coil layout for a given PCB geometry while at the same time obeying to some inequality constraints like maximum power consumption or minimum magnetic dipole generation. Based on the work described in [1], [2] we have developed a KiCAD plugin called KiTorquer based on an existing plugin and a Python package called Shapely.

In this work we start from recapitulating the multi-variate, model-based optimization of EACs. We describe the fundamentals behind the plugin and its application for the design of the optimized EACs of the NanoFF mission. We show the resulting optimized EAC designs that were created with the help of KiTorquer for NanoFF and validate the plugin by comparison of the design parameters and measured properties of the manufactured coils. Since the development of the plugin the NanoFF mission has been launched and an updated version of KiTorquer has been used by the Satlab team of AGH University for the optimized design of EACs for the attitude control system of their HYPE PocketQube. We end this work with a discussion of the results achieved by applying KiTorquer to real world problems and show an outlook on future development activities.

2 RECTANGULAR EMBEDDED AIR COIL OPTIMIZATION

EACs are created by tracks on a number of inner and outer layers of a multi-layer PCB. The geometry of the tracks is driven by the PCB manufacturing process. We will explain the design parameters and optimization algorithm for rectangular EACs as developed in [1], [2] in this section.

2.1 Design Parameters

The design parameters are shown in fig. 1. Outer dimensions of the PCB are given by length a and width b. Center lines of the outer tracks are separated by distances a' and b' where

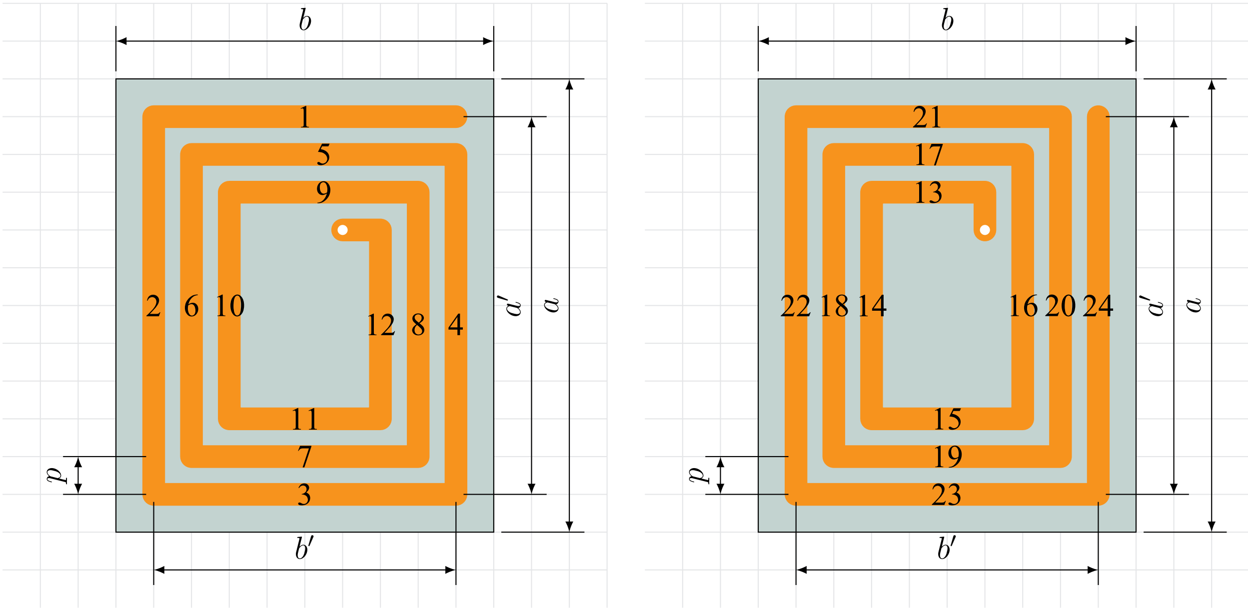
$$a' = a - w - 2d \tag{1a}$$
$$b' = b - w - 2d. \tag{1b}$$

Copper to edge clearance d, required for manufacturing, is measured between the border of the outer track and the outer edge of the PCB. Track has width w and is separated from neighboring tracks by clearance s. Track centers are therefore spaced $p = w + s$ apart. Copper thickness is denoted h. Coil windings are formed by k interleaved rectangular tracks whose geometry repeats on m layers of the PCB. The coil is operated from supply voltage U. As can be seen from fig. 1 the coil maintains its winding sense when switching layers.

2.2 Properties

From a satellite system point of view there are four relevant properties of magnetic actuators. **Magnetic dipole moment** μ is the quantity that generates torque via its interaction with Earth's magnetic field. It is defined by

$$\mu = I \sum_{i=1}^{m} A_i \left(k\right). \tag{2}$$



(a) Odd layer (b) Even layer

Figure 1: Embedded air coil geometry on odd and even layers (cf. [1])

According to Yoon in [3] the area enclosed by the k windings on each of the m layers

$$A_i(k) = \sum_{j=0}^{k-1} (a' - 2pj)(b' - 2pj) \,. \tag{3}$$

Current I through the coil is given by

$$I = \frac{U}{R}, \tag{4}$$

where U is supply voltage and R is resistance, depending on specific resistance ρ, cross-sectional area S, and total track length l:

$$R = \rho \frac{l}{S}. \tag{5}$$

The specific resistance

$$\rho = \rho_0 \cdot [1 + \alpha (T - T_0)] \tag{6}$$

depends on the temperature T of the wire, the temperature coefficient of temperature α, and the specific resistance ρ_0 at reference temperature T_0. Total track length

$$l = 2mk \left(a' + b' - 2p(k-1) \right), \tag{7}$$

cross-sectional area

$$S = wh. \tag{8}$$

Electrical power P is dissipated by the coil due to the supplied voltage and its Ohmic resistance:

$$P = UI = \frac{U^2}{R}. \tag{9}$$

Mass M is the mass of the copper tracks that define the coil and is the product of copper density ρ_{Cu}, total track length l, and cross-section S:

$$M = \rho_{\mathrm{Cu}} l S \tag{10}$$

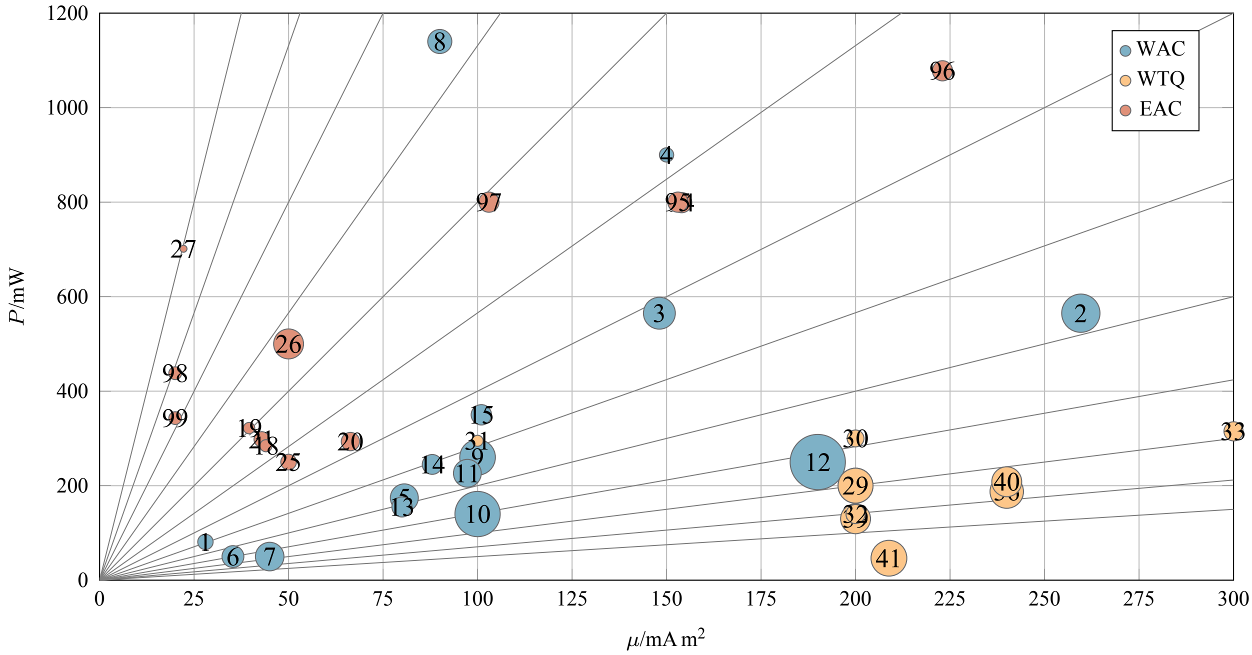


Figure 2: Overview of magnetic actuator properties

Free area A_f is the area in the center kept free of the coil. It is

$$A_f = a_f b_f \tag{11}$$

where

$$a_f = a - 2\left[(w + 2d) + (k - 1)\,p\right], \tag{12}$$
$$b_f = b - 2\left[(w + 2d) + (k - 1)\,p\right]. \tag{13}$$

The four coil properties mentioned above might not be the only ones relevant for other designs. However, in the scope in which the optimization approach described in the following section was used, these were the main ones. Due to the nature of the optimization approach it is straightforward to introduce additional coil parameters and inequality constraints.

Magnetic dipole moment μ, electrical power P, and mass M of magnetic actuators documented in the literature are plotted in fig. 2. To compare similar coils, dipole-power-ratios $\frac{\mu}{P}$ are plotted as straight lines from the origin. The lines drawn in counter-clockwise direction starting at the x axis are representing the ratios $\frac{1}{2}$, $\frac{1}{2\sqrt{2}}$, $\frac{1}{1}$, $\frac{1}{\sqrt{2}}$, and so on until the left-most line which represents a ratio of $\frac{1}{32}$. The unit of the dipole-power-ratio is $\frac{\mathrm{A\,m^2}}{\mathrm{W}}$ which can be simplified to $\frac{\mathrm{m^2}}{\mathrm{V}}$. From this it becomes clear that the efficiency of a magnetic actuator in terms of the dipole-power-ratio can be increased by either increasing the enclosed area or by reducing the voltage provided to the actuator.

2.3 Optimization

Due to the number of tracks k and the number of layers m being integer design parameters, mixed-integer linear programming (MILP) could be used for coil optimization with equality and inequality constraints. However, coil geometry is fixed by PCB dimensions a, b. Usually, the minimum copper to edge clearance will be selected for design to make the best use of the available PCB area. Copper thickness h is also only available in fixed values like 18 µm, 35 µm and 70 µm due to the manufacturing process and pooling of different PCB orders on the same panel. Typically, the supply voltage U is also available only at common values like 1.8 V, 2.5 V, 3.3 V and 5 V due to the voltage levels available on small satellite power buses. The total number of layers in the PCB is usually constrained to a small number like 6, 8 and 10. With the outer layers most likely being reserved for placing components this leaves 4, 6 and 8 internal layers m for the coil design, respectively. While track width w and clearance s have minimum values that are usually depending on the copper thickness there is neither an upper bound nor fixed lists of possible values for these two design parameters. However, due to the low voltages used for magnetic actuators the minimum clearance can be picked to achieve high magnetic dipoles during optimization. As the free area A_f can not be negative, $a_f \geq 0\,\mathrm{mm}$ and $b_f \geq 0\,\mathrm{mm}$ are inequality constraints for the problem.

Due to the small number of actual floating point design parameters, Grau in [2] developed an optimization approach that uses only lists of possible integer values for the variable design parameters that specialize to scalar values in case there is only one choice. For a coil described by n design parameters, the properties p_i are the n-dimensional matrices

$$\text{size}\,(p_1) = m_1 \times m_2 \times m_3 \times \ldots \times m_n \tag{14}$$

$$\text{size}\,(p_2) = m_1 \times m_2 \times m_3 \times \ldots \times m_n \tag{15}$$

$$\vdots$$

$$\text{size}\,(p_l) = m_1 \times m_2 \times m_3 \times \ldots \times m_n \tag{16}$$

where p_1 through p_l are the l properties and m_1 through m_n are the number of elements in the arrays v_1 through v_n that describe the coil design parameters. Using Python, properties p_1 through p_l are calculated and constraints are enforced like

$$\hat{p}_i = \{x \in p_i |\, (p_1 \geq \bar{p}_1) \cap (p_2 \leq \bar{p}_2) \cap (p_3 \leq \bar{p}_3) \cap \ldots\} \tag{17}$$

which will return all elements of matrix p_i for which similar elements in p_1, p_2, and so on obey the rules given by $p_1 \geq \bar{p}_1$, $p_2 \leq \bar{p}_2$, and so on. In this respect, $\bar{p}_i$ are the boundary constraints imposed on the set of p_i, and $\hat{p}_i$ is a subset of elements that obey the limits. If $p_1 = \mu$, $p_2 = P$, and $p_3 = M$ then

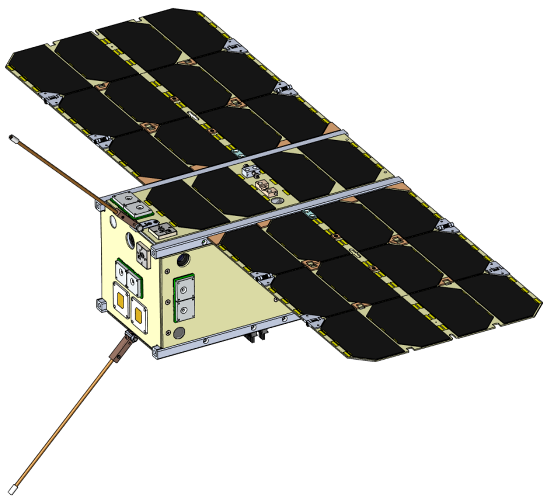
$$\hat{\mu} = \left\{x \in \mu |\, (\mu \geq \bar{\mu}) \cap \left(P \leq \bar{P}\right) \cap \left(M \leq \bar{M}\right)\right\}$$
$$\hat{P} = \left\{x \in P |\, (\mu \geq \bar{\mu}) \cap \left(P \leq \bar{P}\right) \cap \left(M \leq \bar{M}\right)\right\}$$
$$\hat{M} = \left\{x \in M |\, (\mu \geq \bar{\mu}) \cap \left(P \leq \bar{P}\right) \cap \left(M \leq \bar{M}\right)\right\}$$

and $\hat{\mu}$, $\hat{P}$, and $\hat{M}$ would contain only entries of the initial matrices that obey the previously set constraints. Now looking for minimum or maximum values in $\hat{\mu}$, $\hat{P}$, or $\hat{M}$ leads to coil designs that fulfill all of the constraints and that are optimal by means of one of the coil properties.

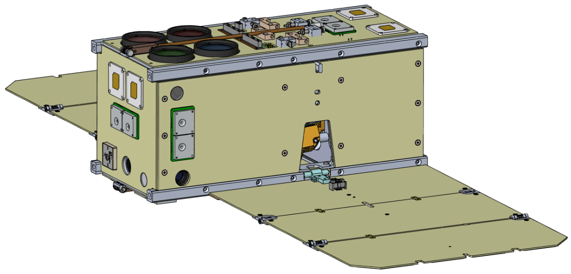
This optimization has already been used in the design of the 0.25 U CubeSats of the Berlin Experimental and Educational Satellite (BEESAT) 5-8 and 10-13 missions and the highly-integrated side panels (HISPs) of the TU Berlin Picosatellite Experiment 7 (TUPEX-7) Rocket Experiments for University Students (REXUS) mission. For these missions, rectangular coils were made possible by keeping certain areas on the internal layers of the side-panel PCBs free from through-holes. For the NanoFF mission, that we will talk about in the following section, it was not possible to keep the PCB peripheral area free from cut-outs and through-holes.

3 THE NANOSATELLITES in FORMATION FLIGHT MISSION

The NanoFF mission is a formation of two 2 U CubeSats intended to demonstrate formation flight in Low Earth Orbit (LEO) [4]. Both spacecraft were deployed from an Orbit Transfer Vehicle (OTV) into



(a) NanoFF spacecraft with solar panels and UHF antennas deployed, Z+ direction facing down



(b) NanoFF with solar panels deployed, Z+ direction facing up

Figure 3: CAD model of the NanoFF spacecraft in differen configurations

a carefully designed Helix orbit in January 2024, which ensures collision safety. As of this writing, both CubeSats are healthy and undergoing commissioning operations for all the subsystems. Manual formation control activities using a resistojet propulsion system has already commenced. Upcoming activities involve deploying the solar panels and performing autonomous formation control, using the propulsion system and differential drag.

3.1 Mission Objectives

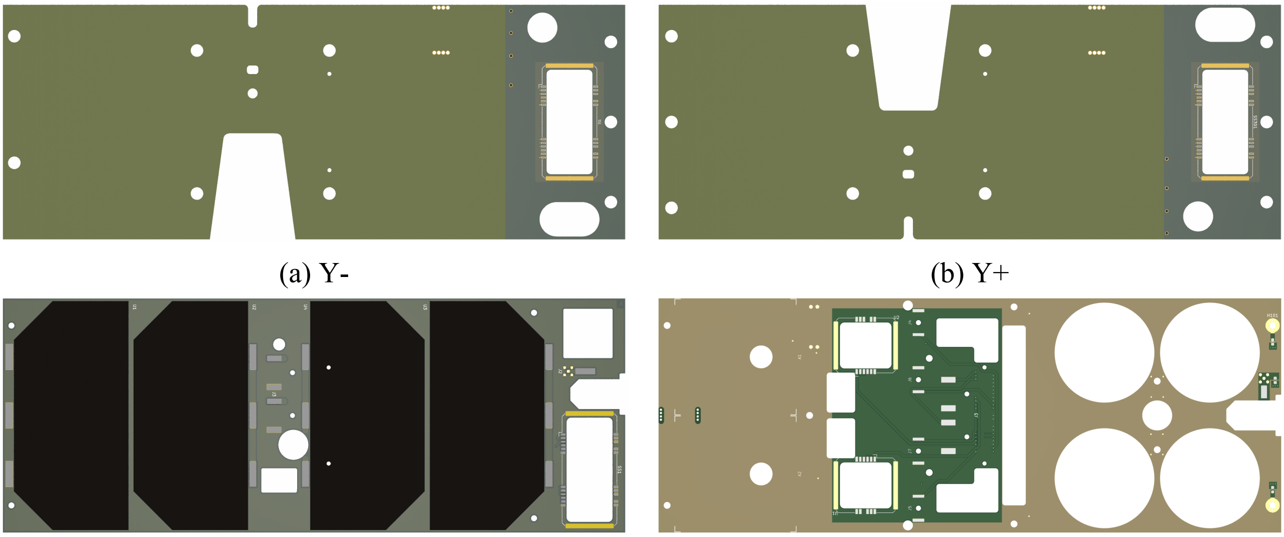
The primary mission objective is to fly a helix orbit. The satellites can be moved along their orbits in order to change the distance and to demonstrate different application scenarios. In the course of the mission, further formations are to be adopted in which one of the satellites will autonomously maintain its relative position to the other.

3.2 Attitude Determination and Control System

Driven by the requirements of the propulsion system, the NanoFF spacecraft feature redundant AOCS nodes containing micro electro-mechanical system (MEMS) magnetic field sensors, sun sensors, angular rate gyroscopes, and self-developed star trackers for attitude determination. A tetrahedral RWA provides the torque for thruster pointing, while three pairs of magnetic actuators are used for detumbling after deployment and angular momentum dumping. Due to the high packing density of the NanoFF spacecraft, wound torquerods (WTQs) could not be used. Instead, a pair of WACs on the body x axis was used, allowing for coil mounting between the two main avionics boards. Additionally, EACs are integrated into the four larger side panels on the body y and z axes.

3.3 Side Panels

Figure 3a shows the NanoFF spacecraft solar arrays and ultra-high frequency (UHF) antennas with deployed and its Z+ axis pointing down. On the spacecraft body the central solar panel is the Z- side panel. Besides the four solar cells it is equipped with the antenna joint and release mechanism, a double sun sensor, and a global navigation satellite system (GNSS) antenna. Figure 3b shows the NanoFF spacecraft with only solar panels deployed and its Z+ axis pointing up. The Z+ side panel is equipped



(a) Y-

(b) Y+

(c) Z-

(d) Z+

Figure 4: Top view of NanoFF side panels in the KiCAD 3D viewer

with the release mechanism and joint of a UHF antenna, the release mechanisms for both solar arrays, the umbilical connector, and the optics of the multi-spectral camera.

Figures 4a and 4b show the as-built Y-/Y+ side panels in the KiCAD 3D view. The Y panels are mainly characterized by the trapezoidal cut-out required for the solar panel joint and the cable connecting the panel to the electrical power system (EPS). The two circular holes and the small slit next to the larger cut-out create a more or less complete interruption between the left and right part of the side panel. On the right part, there are additional three large cut-outs for the star tracker, the double sun-sensor board, and the retro-reflector. The Y side panels are nearly symmetrical twins except for the row of through-holes on the left edge of the panels.

The as-built configuration of the Z- side panel shown in fig. 4c has the least area missing. There is only a small rectangular cut-out for the antenna deployment mechanism and the retro-reflector in the center of the board. And three cut-outs on the right edge of the side panel where the GNSS antenna, the UHF antenna, and the double sun-sensor are mounted.

The Z+ side panel of which the as-built configuration is shown in fig. 4d has the most area missing. This holds especially true for the right edge, where the four openings for the camera heads, the UHF antenna joint, and the retro-reflector are located. But also a large fraction of the area in the center of the board is missing, where the umbilical connector, the release mechanisms for the UHF antenna and solar panels, and two single sun sensors are located.

Overall, it can be seen that the side panel area is heavily constrained by the cut-outs required for the different components. The following section describes the development of the KiTorquer plugin that was used to auto-generate optimal coil designs for the NanoFF mission.

4 THE KITORQUER PLUGIN

We developed the KiTorquer plugin to be able to design optimal magnetic actuators on geometrically heavily constrained side panel PCBs. This section defines the intended functionality of the plugin and explains its implementation.

4.1 Plugin Functionality

The KiTorquer plugin shall provide the following functionalities:

- Find the outer and inner contours of the PCB, the vias, the through-hole pads, and copper keep-out zones;
- Calculate the positions of the vias required for connecting the different layers of the coil and allow the user to pick the via size and annular ring width;
- Layout the coil windings on the selected layers and take care of maintaining winding sense in odd and even layers:
- Calculate the electro-magnetic properties of the layn-out coil;
- Allow the user to define inequality constraints for coil parameters like required minimum magnetic dipole or allowed maximum power consumption and to define design parameter ranges for the coil design parameters;
- Draw the coil in the selected layers;

4.2 Plugin Implementation and Usage

KiTorquer is implemented as a so-called *Action Plugin* in KiCAD (cf. [5]). The plugin can be run by navigating to *Tools → External Plugins → KiTorquer*. Coil design parameters are currently entered by a mix of built-in dialogs in KiCAD and the `__init__.py` file located at the root folder of the plugin. Via size, annular ring width, and copper to edge clearance can be set from the *Board Setup* dialog under *Design Rules → Constraints*. All other design and optimization parameters need to be set from the `__init__.py` file. The optimization runs are also controlled from the same file.

KiTorquer is further structured into two main components: the `EmbeddedAirCoil` class that finds the available area for the coil design, places the vias, lays out the coil windings, and calculates the electromagnetic properties of the coil is the one, the other is the `CoilDrawer` class that is used to draw the coil windings on the selected layers in the PCB. Both classes are using functionalities from the KiKit [6] plugin for KiCAD. KiKit is a Python library, KiCAD plugin, and a command line interface (CLI) tool intended for the panelization of regular and oddly shaped boards and to automate several other tasks in a standard KiCAD workflow.

The panelization functionality in KiKit is based on functionalities provided by Shapely [7], a Python package for set-theoretic analysis and manipulation of planar features using functions from the GEOS library [8]. The KiTorquer `EmbeddedAirCoil` class extends the substrate detection provided by KiKit by the detection of vias, through-hole pads, and copper keep-out zones. The polygon that represents the milled interior and exterior geometries is buffered by the copper to edge clearance. The resulting polygon is split in interior and exterior polygons. The exterior polygon is then used to start the coil generation process by buffering by half of the track width. The resulting polygon represents the outermost winding of the coil. For the following winding, a combination of the last winding and the interior extracted at the beginning of the process is buffered by track width plus track separation. The exterior of the resulting buffered polygon is kept, its interior parts are discarded. If by buffering the combination, any of the original interior contours are merged into the buffered polygon, their initial representations are deleted and not considered anymore. This process is repeated until the desired number of windings is achieved.

The polygons representing the windings are then used to calculate the coil parameters as described in section 2.2. Shapely provides methods to calculate the area enclosed by a polygon and also the length of the contour of the polygon. The calculated values for the area and length can be used directly with eqs. (3) and (5) to calculate the other properties of the coil. The calculated values are then used to start the optimization process. Currently, the optimization is implemented in the `__init__.py` file and will move to its own class in the future. For the optimization, a set of variable parameters can be defined and are looped over. During each loop, the coil geometry is generated based on the current set of variable parameters. The relevant calculated parameters are stored together with the design parameters and kept for the final optimization. As described in section 2.3, if one of the calculated parameters is violating the boundaries, it is discarded from the list of possible coil designs. In the end, the user can select from the possible designs based on maximum or minimum criteria, e.g. the design with the maximum magnetic dipole moment.

To draw the optimized coil, the user needs to select the position of the vias that connect the layers of the coil to each other. The *tab* footprint, which resembles an arrow, from the KiKit plugin is placed so that it points from the outside of the PCB perpendicularly towards the edge of the PCB. This so-called annotation footprint is then detected by the `CoilDrawer` class. The required vias on the outside and the inside of the coil are then placed, where the line along the arrow intersects the outermost and innermost winding of the generated coil design. Actually, one additional winding is added to the inside of the coil which is then used to place the vias enclosed by the coil windings. When the vias are placed, the windings are broken up close to the vias and the first winding is connected to the first outside via on one end while the other end is connected to the next winding. This process is repeated until the last winding is connected to the first inside via. On the next layer, the vias and windings are connected in the inverse order starting from the first inside via and ending at the second outside via. Thus the winding sense is kept constant on all layers. This two steps are repeated until all layers selected by the user have been used to place the coils.

4.3 Plugin Distribution and Further Development Plans

While the core of the KiTorquer plugin has been developed within the scope of the NanoFF mission, it might be interesting to others to use the plugin for their small satellite projects or other applications that require optimized coil designs on geometrically constrained PCBs. Therefore the KiTorquer project is hosted on GitLab[1]. The project page holds information on the installation of the plugin and the issue tracker is active and waiting for your feedback.

We hope to develop the plugin further and provide a proper graphical user interface (GUI) in the future. The GUI shall allow to select which design parameters are fixed an which ones are variable and it shall be possible to set ranges for the variable parameters. Also, the GUI shall allow to carry out the optimization and drawing of the coils in two separate steps. We wish to generate plots of the found solutions and enable to user to pick interactively from the those solutions to draw the coil.

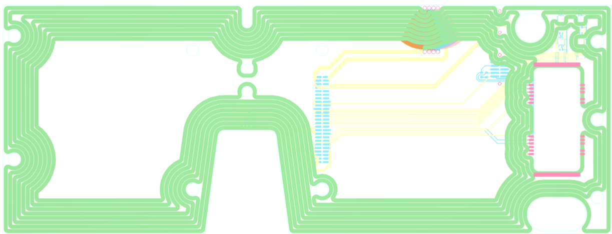
5 AUTOMATED EMBEDDED AIR COIL DESIGNS

This section describes the generated coil designs for the NanoFF and HYPE missions.

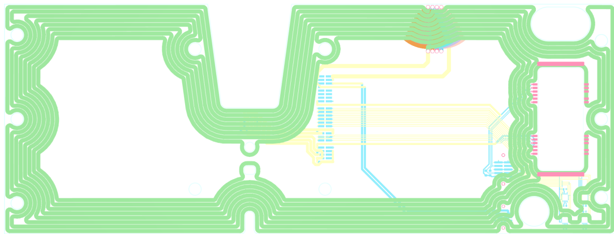
5.1 NanoFF

In this section we introduce requirements and constraints for the coil optimization and show the results for the NanoFF spacecraft.

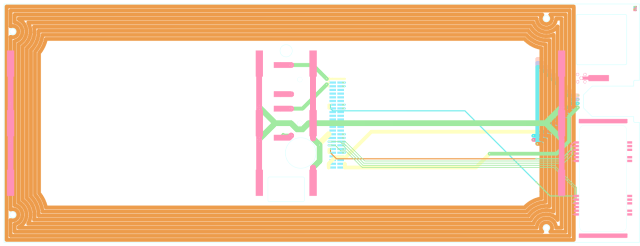
[1] `https://gitlab.com/space-stuff/kitorquer`



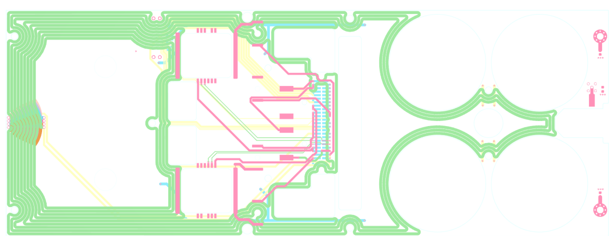
(a) Y-



(b) Y+



(c) Z-



(d) Z+

Figure 5: Generated coil layouts of NanoFF side panels in the KiCAD 3D viewer

5.1.1 Requirements and Constraints

The side panels have multiple functions on the NanoFF satellites. Power from the solar arrays is routed to a board-to-board connector that mates with the power control unit (PCU) board. Two sun sensors are connected on each of the Y-, Y+, Z- and Z+ panels, their power and data lines are routed to the AOCS nodes via the same board-to-board connector. Furthermore, the Z- panel routes the power and data signals from the satellite bus to the propulsion module. Both Z panels contain release mechanisms for the UHF antennas and Z- panel contains additional release mechanisms for the deployable solar panels. Due to the release mechanism design, the total thickness of the Z panels needs to be 1.1 mm ± 0.1 mm. The AOCS design requires a magnetic dipole moment of 300 mA m^2 per axis, it was decided to try to split the required magnetic dipole as evenly as possible between the two panels per axis. In order to support the electrical functions and to achieve the required dipole on each panel, the panels were manufactured using PCB high density interconnect (HDI) 1-8-1 technology. This results in a total of ten layers, with laser-drilled microvias connecting each surface layer with the internal layer directly below of it.

5.1.2 Optimized Embedded Air Coil Designs

Figure 5 shows the copper layers of the side panel PCBs designed for NanoFF containing the optimized air coils. Table 1 holds the numerical values of the generated coil designs. It can be seen, that all generated coils follow the outer contour and move around the various holes in the PCBs nicely.

The main limitations on the Y axis side panels in figs. 5a and 5b are the opposing recesses on the outer contour where the hinge for the solar panel sticks through and the neighboring mounting holes. This effectively separates the Y axis side panels into two halves, greatly reducing the amount of magnetic dipole moment per panel. In the area, where the sun sensors, laser reflector, and star tracker are mounted, the plugin is able to layout the coil in an area, that would not be usable otherwise.

The Z- panel shown in fig. 4c is mostly covered with solar panels and antennas on the outside and thus offers the largest coherent area for a coil. Therefore, the generated coil design shown in fig. 5c looks

Table 1: Comparison of coils created for the NanoFF mission with the help of KiTorquer

mission		NanoFF				HYPE	
side panel		Y-	Y+	Z-	Z+	def.	Y-
index in fig. 2		94	95	96	97	98	99
mill separation	mm	0.3	0.3	0.23	0.3	0.3	0.3
track width	mm	1.444	1.448	1.293	1.044	0.340	0.300
nominal track separation	mm	0.130	0.130	0.130	0.130	0.175	0.175
minimum track separation	mm	0.107	0.106	0.108	0.106	0.155	0.156
copper thickness	µm	35	35	35	35	35	35
winding count		8	8	9	9	19	23
layer count		7	7	6	7	6	4
voltage	V	3.3	3.3	3.3	3.3	3.3	3.3
temperature	°C	20	20	20	20	20	20
enclosed area	m^2	0.637	0.632	0.681	0.425	0.153	0.193
track length	m	35.166	35.242	27.198	25.415	17.619	19.889
cross-section	mm^2	0.043	0.043	0.045	0.031	0.012	0.010
resistance	Ω	13.622	13.613	10.085	13.617	24.844	31.784
current	A	0.242	0.242	0.327	0.242	0.133	0.104
magnetic dipole	$A\,m^2$	0.154	0.153	0.223	0.103	0.020	0.020
power	W	0.799	0.800	1.080	0.800	0.438	0.343
dipole-power-ratio		0.193	0.192	0.206	0.129	0.046	0.058

like a rectangular coil. The benefit of using the plugin in this case is, that it is not necessary to route the coil manually around the four mounting holes in the corners of the panel.

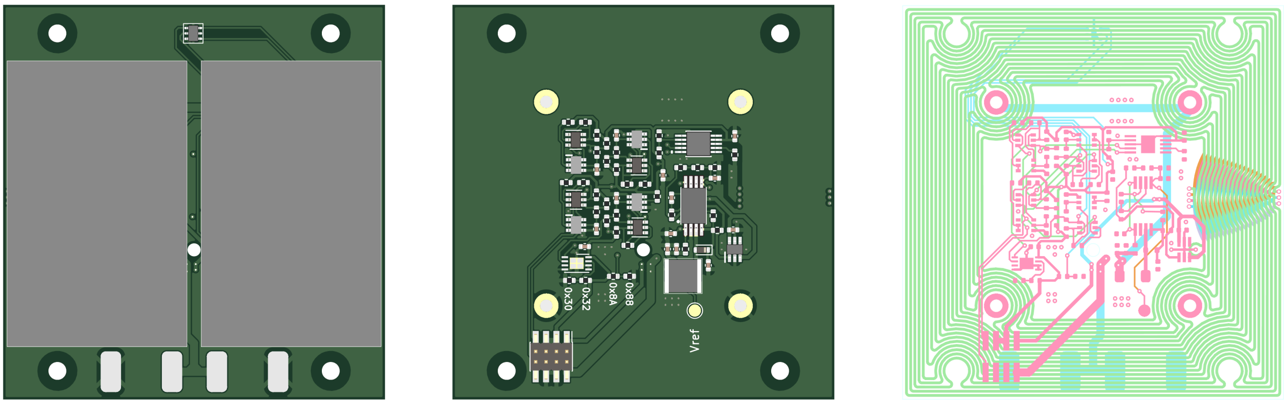
The Z+ panel features the most challenging geometry for the coil layout. As can be seen in fig. 4d, there is a large number of holes in the PCB. From fig. 5d it can be seen that even the plugin is not able to layout more than three windings in the right half of the panel. The coherent area in the left half of the panel is efficiently made use of but even then the magnetic dipole is only 103 $mA\,m^2$ – which is the smallest of all four embedded coils.

5.2 HYPE

HYPE is a 1 p PocketQube[2] mission developed by the Satlab student research group located at the Space Technology Centre of AGH University of Krakow, Poland. The spacecraft is set to launch on Transporter-11 in July 2024 and features a camera mounted to a deployable boom dubbed selfie stick. The spacecraft features overall dimensions of 64 mm × 58 mm × 51.6 mm in accordance with the PocketQube standard. There are two main types of side panels: the so-called default side panel that is referenced by index 98 and the Y- panel, referenced by index 99. Confer fig. 2 and table 1 for data on the two different types of side panels.

The outside, inside, as well as internal and external copper information of the default side panel is shown in fig. 6. In contrast to the side panels on NanoFF, this side panel has the coil placed closed to the perimeter of the PCB. The main interference is with the four mechanical mounting holes and the four through-holes used for soldering the contacts of the solar cells. The coil uses a total of six out of the available eight layers of the PCB. Components for maximum power point tracking (MPPT),

[2]See `https://www.albaorbital.com/pocketqube-standard` for documentation on the PocketQube standard

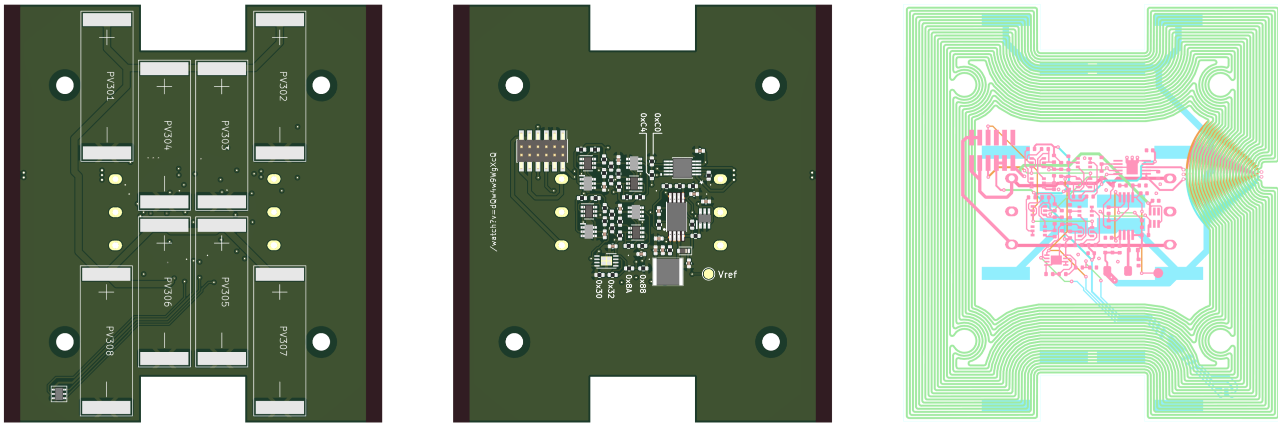


(a) Outside with solar cells and photo diode

(b) Inside with electrical components

(c) Copper on internal and external layers

Figure 6: Views of the HYPE default side panel



(a) Outside with solar cells and photo diode

(b) Inside with electrical components

(c) Copper on internal and external layers

Figure 7: Views of the HYPE Y- side panel

temperature and ambient light sensors, the coil driver, and protection of the inter-integrated circuit (I2C) bus are concentrated on the center of the board. The center is also the only area, where vias are allowed to connect through the PCB. The coil is optimized to create 20 mA m^2 of magnetic dipole moment at 20 °C and 3.3 V and consumes 438 mW.

Figure 7 shows the outside, inside, as well as internal and external copper information of the Y-side panel. This PCB is the sliding backplate that is used to store and deploy the PocketQube. The backplate features tabs on the left and right edge where the top and bottom surfaces of the PCB are free from copper and solder resist. Slots are added at the top and bottom edge of the PCB to be able to store the antennas while the spaecraft is located inside the deployer. The outside is again populated with solar cells, while the inside is populated with the same components as the default side panel. In fig. 7c it can be seen, that the coil follows the perimeter of the PCB closely and that it is routed around the mounting holes where necessary. As vias are forbidden where the tabs are located on the sliding backplate PCB, a keep-out area was used in combination with the KiKit tab footprint to move the vias inside the allowed area. The position is indicated by the notch in the perimeter of the coil on the right edge in fig. 7c. This coil is also optimized to create 20 mA m^2 of magnetic dipole moment at 20 °C and 3.3 V but consumes only 343 mW. Reduced power consumption is attributed to the increased enclosed area: while the coil on the default side panel encloses an area of 0.153 m^2, the coil on the Y-side panel – which is longer and wider – encloses an area of 0.193 m^2.

Table 2: Measured and derived properties of the coils of the three NanoFF spacecraft

(a) measured currents [mA]

FM	Y-	Y+	Z-	Z+
A	223.12	221.88	221.62	210.12
B	224.50	220.50	221.75	207.25
C	223.38	223.62	220.00	209.12
ref.	242	242	327	242

(b) derived magnetic dipole moments [mA m^2]

FM	Y-	Y+	Z-	Z+
A	0.14	0.14	0.18	0.09
B	0.14	0.14	0.18	0.09
C	0.14	0.14	0.18	0.09
ref.	0.154	0.153	0.223	0.103

The dipole-power-ratios of the HYPE coils is 0.046 and 0.058, respectively. This is significantly lower than the 0.129 which is the lowest dipole-power-ratio of the NanoFF coils. As explained in section 2.2, it would have been possible to improve the dipole-power-ratio by increasing the enclosed area or reducing the coil supply voltage. For the presented side panels, the enclosed area could not be further increased without interfering with the components and vias placed at the center of the PCBs. It was also decided against introducing additional step-down converters to achieve voltages below 3.3 V as not to increase the number of electrical components in the spacecraft and thereby increase the complexity and risk of failure.

6 PLUGIN VALIDATION

In this section, measurements taken on the ground and in space with the NanoFF spacecraft are used to validate the plugin and the coils created with it.

6.1 Measurements during Integration

During integration of the NanoFF spacecraft, the coils were measured using the onboard sensors. Results are listed in table 2 for the flight models (FMs) A, B, and C together with the reference values for all Y and Z side panels. Measured currents in table 2a show a good agreement for the two Y panels – values differ by about 8 % from the reference – and relatively good agreement for the Z+ panel – values differ by about 14 %. However, the Z- panel is off by about 32 %. A similar behavior is seen in table 2b. The magnetic dipole in the table is derived from the current values in table 2a by multiplying with enclosed areas of 0.637 m^2, 0.632 m^2, 0.804 m^2 and 0.425 m^2 for the Y-, Y+, Z-, and Z+ side panels, respectively. The values for the Y- and Y+ panel are again in good agreement with the values differing by 10 % from the design value. Also Z+ is showing a good agreement with a difference of 12.6 %. Again it is the Z- panel that is off by about 19.3 %.

It is currently unclear to us, what caused the differences between the design parameters and the measured values of the coil embedded in the Z- panel. Effects like the temperature of the coil during measurement differing from the design temperature or biased calibration of the integrated sensors might cause the differences between design and measured values for the other three coils. We will investigate what has caused the differences and will provide an update of this work with our findings. However, the measurements showed, that the coils in the Y+/- side panels are nearly identical, which we take as a confirmation, that the optimized design of EACs is possible.

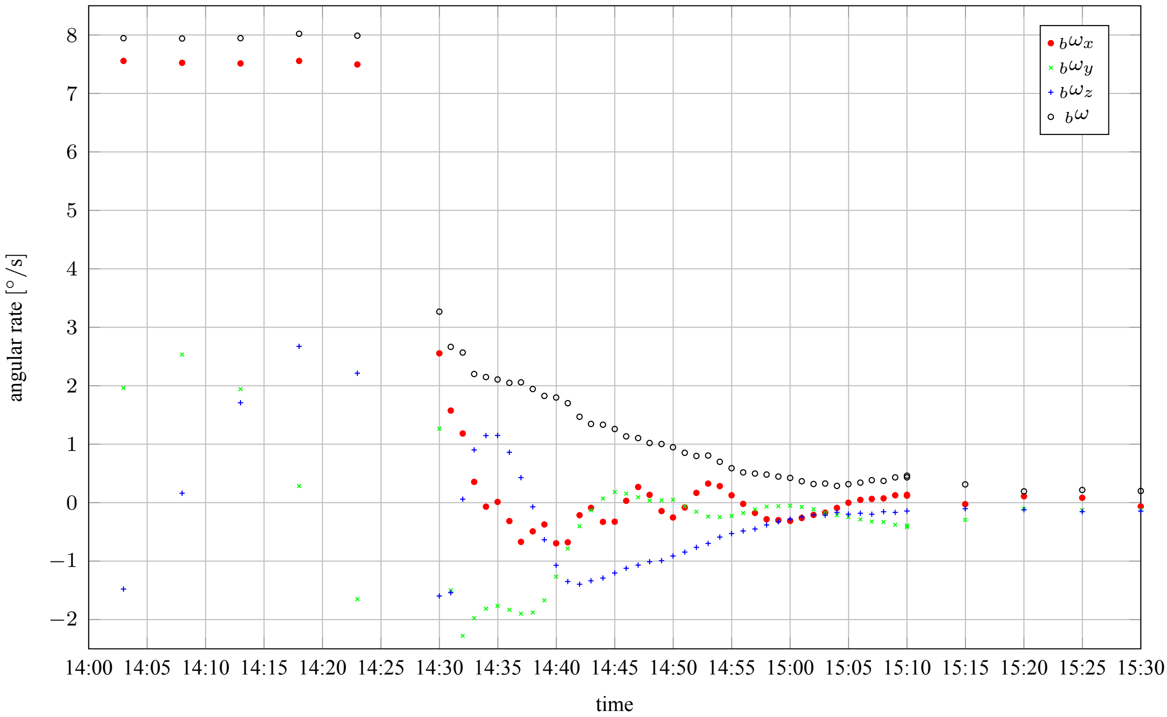


Figure 8: Effect of B-Dot on NanoFF-B on 2024-01-11

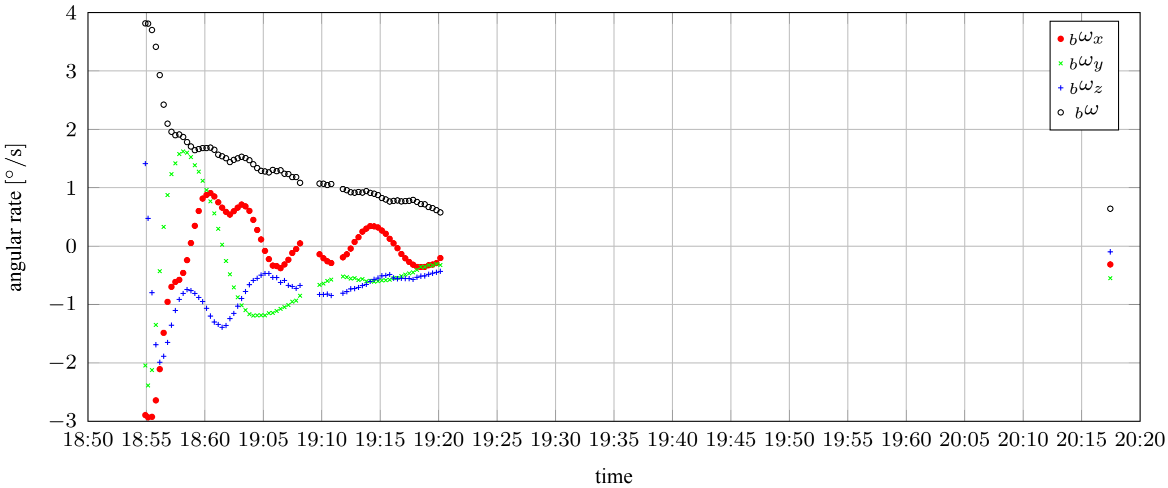


Figure 9: Effect of B-Dot on NanoFF-A on 2024-02-05

6.2 In-Orbit Results

The EACs in the side panels are primarily used for initially detumbling the spacecraft after deployment. Later during the mission the coils are mainly foreseen for parallel desaturation of the reaction wheels. For detumbling, a combination of magnetic field sensors and the coils is used to reduce the rate of change of the measured geomagnetic field below a certain threshold. In this section, we show the effectiveness of the auto-generated magnetic coils of the NanoFF mission for detumbling after orbit insertion in January 2024.

After NanoFF-B's deployment on January 10, 2024, B-Dot was activated within 24 hours, effectively reducing the total angular rate from 8 °/s to under 0.5 °/s within just one orbit (see fig. 8). Following NanoFF-A's deployment on January 15, 2024, the total angular rate was approximately 5.8 °/s. Over the course of 3 weeks, natural damping gradually decreased this to around 4 °/s. Subsequently, after B-Dot activation, the angular rate was further reduced to approximately 0.5 °/s within half an orbit (see fig. 9). Despite the discrepancies found between design and measured coil parameters, the data from the detumbling of the two spacecraft shows that they are performing well on orbit, which we take as a further confirmation for the successful optimization.

7 DISCUSSION and OUTLOOK

In this work we present the development of the KiTorquer action plugin for KiCAD that allows to generate optimized EACs for magnetic attitude control in heavily constrained PCBs in an automated fashion. The plugin provides a significant advantage over the manual design of EAC, greatly reducing the time to draw the coil while at the same time providing a way to come up with an optimal design. We were able to quickly implement a demonstrator of the automated coil design by building on the KiKit plugin, which can be used for PCB panelization in KiCAD. The KiKit plugin in turn is based on Shapely, which is a Python package for set-theoretic analysis and manipulation of planar features. While not being fully fledged, KiTorquer has already been used for the coil design of the NanoFF spacecraft of TU Berlin and the HYPE spacecraft of AGH University. During the design of the coils for HYPE the plugin has seen tremendous improvement.

However, there are still some caveats. Due to the fact that the coil geometry is based only on linear elements, it is not possible to maintain a precise track separation without slowing the coil generation down too much. In order to meet the manufacturer requirements of having a certain minimum track distance, we currently use a larger track separation and calculate the actual minimum track distance. The calculated minimum track distance must be monitored to fulfill the manufacturer's requirements. One possible solution is to rework the coil generation engine to support native KiCAD PCB elements in order to mitigate the problem with the minimum track separation. However, this is tedious to implement as it is then necessary to reimplement the functionalities provided by the KiKit plugin and Shapely. Besides the planned improvement of the GUI, the rewrite of the coil generation algorithm is on our roadmap.

We hope, that other nanosatellite projects find our work helpful. And we are looking forward to use being made of our plugin for applications that we could have not foreseen while creating the plugin. Enjoy!

ACKNOWLEDGEMENTS

This work is supported by the German Aerospace Center (DLR) with funds provided by the Federal Ministry for Economic Affairs and Energy (BMWi) under the grant numbers 50 RU 1701 and 50 RU 1803.

References

[1] S. Grau, I. Suchantke, and K. Brieß, "A comprehensive study on magnetic actuator design for cubesat missions," in *Proceedings of the 68th International Astronautical Congress*, (Adelaide, Australia), 2017 (cit. on pp. 1–3).

[2] S. Grau, "Contributions to the advance of the integration density of cubesats," Ph.D. dissertation, 2019. DOI: `https://doi.org/10.14279/depositonce-7293` (cit. on pp. 2, 5).

[3] Z. Yoon, “Drei-Achsen-Stabilisierung eines Pico-Satelliten mit Magnetspulen,” German, M.S. thesis, Berlin, 2006 (cit. on p. 3).

[4] S. Weiß, N. Korn, R. Wolf, Y. Lim, D. Roychowdhury, and F. Baumann, “NanoFF: A 2u-CubeSat formation flight mission,” in *70th International Astronautical Congress*, 2019 (cit. on p. 5).

[5] “Pcb python bindings.” (2023-09-21), [Online]. Available: `https://dev-docs.kicad.org/en/apis-and-binding/pcbnew/` (visited on 02/20/2024) (cit. on p. 8).

[6] “Kikit - automation for kicad.” (2023-09-16), [Online]. Available: `https://yaqwsx.github.io/KiKit/v1.4/` (visited on 02/20/2024) (cit. on p. 8).

[7] “Shapely.” (2024-02-16), [Online]. Available: `https://github.com/shapely/shapely` (visited on 02/20/2023) (cit. on p. 8).

[8] GEOS contributors, *GEOS coordinate transformation software library*, Open Source Geospatial Foundation, 2021. [Online]. Available: `https://libgeos.org/` (cit. on p. 8).

Glossary

AGH University AGH University of Krakow (Polish: Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie) 2, 11, 15

AOCS attitude and orbit control system 1, 6, 10

BEESAT Berlin Experimental and Educational Satellite 5

BMWi Federal Ministry for Economic Affairs and Energy 15

CLI command line interface 8

CubeSat A smallsatellite form factor 1, 5, 6

DLR German Aerospace Center 15

EAC embedded air coil 1, 2, 6, 13–15

EPS electrical power system 7

FM flight model 13

GEOS GEOS is a C/C++ library for computational geometry with a focus on algorithms used in geographic information systems (GIS) software. It implements the OGC Simple Features geometry model and provides all the spatial functions in that standard as well as many others. GEOS is a core dependency of PostGIS, QGIS, GDAL, Shapely and many others. 8

GNSS global navigation satellite system 6, 7

GUI graphical user interface 9, 15

HDI high density interconnect 10

HISP highly-integrated side panel 5

HYPE 2, 9, 11, 13, 15

I2C inter-integrated circuit 12

KiCAD KiCAD is a free software suite for electronic design automation 2, 7, 8, 15

KiKit KiKit is a Python library, KiCAD plugin, and a CLI tool to automate several tasks in a standard KiCAD workflow like: panelization of both, regular and oddly shaped, boards; automated exporting manufacturing data based on manufacturer presets; multi-board project in KiCAD; building board presentation pages 8, 9, 12, 15

KiTorquer KiTorquer is a KiCAD action plugin developed based on the KiKit KiCAD plugin and the Shapely Python package for the design of optimized embedded air coils on geometry constrained CubeSat side panel 2, 7–9, 15

LEO Low Earth Orbit 5

MEMS micro electro-mechanical system 6

MILP mixed-integer linear programming 4

MPPT maximum power point tracking 11

MTQ magnetorquer 1

NanoFF Nanosatellites in Formation Flight 1, 2, 5–7, 9–11, 13–15

OTV Orbit Transfer Vehicle 5

PCB printed circuit board 1, 2, 4, 5, 7–13, 15

PCU power control unit 10

PocketQube A picosatellite format factor 2

Python Python is a high-level, general-purpose programming language 2, 5, 8, 15

REXUS Rocket Experiments for University Students 5

RW reaction wheel 2

RWA reaction wheel assembly 1, 6

Shapely Shapely is a Python package for set-theoretic analysis and manipulation of planar features using functions from the well known and widely deployed GEOS library 2, 8, 15

TU Berlin Technische Universität Berlin 1, 15

TUPEX-7 TU Berlin Picosatellite Experiment 7 5

UHF ultra-high frequency 6, 7, 10

WAC wound air coil 1, 6

WTQ wound torquerod 6