

Analysis of CLIC power consumption by hardware category via a graphical representation

Torkel Genet*

CERN Summer Student

1 Introduction

This project involved analysing the power loss estimates of the proposed Compact Linear Collider (CLIC), for the purpose representing the different power losses graphically on an illustration.

2 Motivation

The purpose of this project is to gain insights into which subsystems dissipate the most power and where in the accelerator these losses occur. Through the representation of the information in a way that is easy to understand. The CLIC is expected to consume on the order of up to 200 MW, which is significant compared to CERN's current power consumption of 200 MW in the period of May to mid-December [1], therefore it is worth to know where this power is spent in order to optimise the efficiency of the accelerator.

3 Contribution

The different figures for the power losses in the CLIC came in the form of a Spreadsheet exported from a cost tracking tool called *CostoolPBS*. The entries all have a unique hierarchical code which represents the parent entry and the entry number as you can see in Table 1.

Table 1: Example of entry structure in the PBS spreadsheet, with the different power figures omitted

level	Code	Name*	Multiplicity*	...
3	3.1.1.	Two-Beam Module Type 0 e+		...
4	3.1.1.1.	RF System	1	...
5	3.1.1.1.1.	Super-accelerating Structures	4	...
5	3.1.1.1.2.	PETS	4	...
5	3.1.1.1.3.	RF Distribution Network	1	...
6	3.1.1.1.3.5.	Loads after AS	16	...

3.1 Choice of tools

Having been given a great freedom in the choice of tools for this project, I decided to use python, for it's great number of available libraries, notably *Pandas*¹ for importing the tables from the spreadsheet and *Python Imaging Library*² for adding the figures to the graphic. For the developing environment I chose *Jupyter notebooks* hosted on CERN's *swan.cern.ch* server. This was chosen in order to facilitate sharing and quick modification of the tool that has been written.

*National Institute of Applied Science (INSA), Toulouse, France

¹<https://pandas.pydata.org/>

²<https://python-pillow.org/>

3.2 Importing the data into Python

Pandas has built-in support for importing spreadsheets in *Excel* format. When reading the Excel-file into a *Pandas dataframe* you can choose the type for each column. For this project, all the columns where either strings or floats.

The original spreadsheet was separated into multiple sheets, one for each system, plus one that regroups all the entries on level 0 through 3. In order to be able to access and sum up the different values for the project all the sheets were concatenated into one big *Pandas dataframe*. As long as the naming of the columns in the *Excel* spreadsheet is consistent this works.

The result of this is one big dataframe where all the relevant information from the spreadsheet is available.

3.3 Getting the wanted power figures

The figures that need to be extracted from the dataframe for each system are the following: RF, Magnets, Total, Share of power dissipated to water and air in the tunnel and on the surface, and change in both Drivebeam and Mainbeam power. These values are extracted from the dataframe in a manner similar to a database query using *Boolean indexing*, a logical expression applied to each entry in the dataframe in order to select the entry or not. For example, to return a dataframe of all entries related to RF in one system, the following line is used:

```
FullTable[(FullTable.Code.str[0:4] == "1.2.") & ((FullTable.Code.str[-3:] == '.1.') | (FullTable.Code.str[-3:] == '.2.')) & (FullTable.level==4)][ 'Code ']
```

This line selects all entries where the *Code* matches the system code prefix "1.2.", and ends in either ".2." or ".4." who by convention are the last digit in the code for *RF System*, and *RF Powering System* respectively. This structure is similar for the *Magnet System* and *Magnet Powering System*.

One important detail to remember when aggregating the figures for each system is to multiply each power figure for an entry with the product of all parent multiplicities to get the overall contribution of one entry.

3.4 Presenting the figures on an illustration

The graphical part of this project consists of a base illustration that is read in python with the help of the *Python Imaging Library (Pillow)* and then overlaid with text and numbers for each system.

The base illustration is an adapted version of the CLIC illustration found in the CLIC Conceptual Design Report [2] retraced in order to obtain a modifiable vector-image in order to be able to adapt the illustration more easily in the future.

The resulting image can be found in Appendix A.1.

4 Future improvements

Since the CLIC is still under development, the power figures are not yet available for all systems. Updating the illustration with new numbers should be a relatively simple task. An other future addition would be to make the same type of illustration for the klystron based option of the CLIC. This would allow a quick overview and comparison of the two options.

Acknowledgements

I wish to thank my supervisor Steinar Stapnes (ATS-DO), and Alexej Grudiev (BE-RF-LRF) for their help and explanations regarding the project. I also want to thank the Summer Student Team for their warm welcome on my first day, and their following attention to taking care of the summer students.

Bibliography

- [1] Powering CERN, <https://home.cern/about/engineering/powering-cern>
- [2] A Multi-TeV linear collider based on CLIC technology: CLIC Conceptual Design Report, edited by M. Aicheler, P. Burrows, M. Draper, T. Garvey, P. Lebrun, K. Peach, N. Phinney, H. Schmickler, D. Schulte and N. Toge, CERN-2012-007

Appendix

A Output illustration with figures as of 25/08/2018

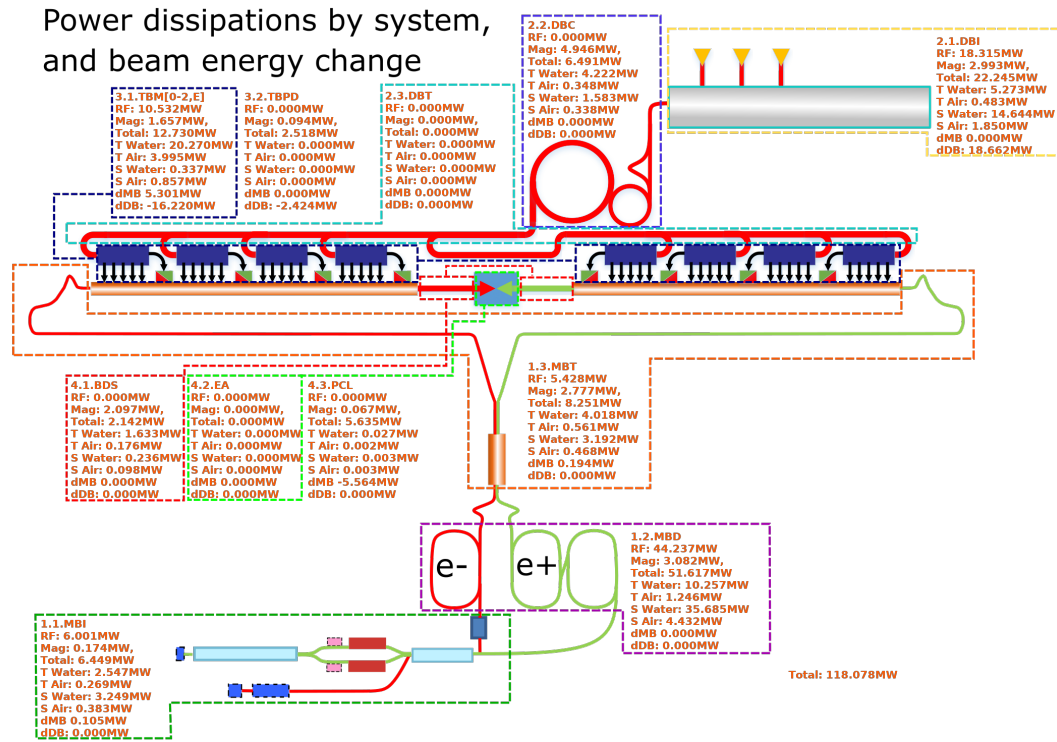


Figure A.1: Output illustration with figures as of 25/08/2018