Satellite Communication

Release 0.0.1

Jan-Philipp Herzog

Aug 11, 2023

CONTENTS:

CHAPTER

ONE

USAGE

The Program can be used via a command line interface. Execute the sat_slot_counter.py file with the relative path to a datafile containing time windows to run the analysis and find all overlapping time windows with number of satellites observed during this time.

Running the script from the /src folder. .. code-block:

```
$ python sat_slot_counter.py ../dat/satellites.dat
```

For more information:

```
$ python sat_slot_counter.py -h
```

CHAPTER

TWO

FILE IMPORTER

class importer.**File**

File importer

Contains all functions regarding reading and parsing data from file. Can be expanded to handle output as well.

classmethod load(*filename: str = 'satellites.dat'*, *filepath: str = '../dat'*) → DataFrame

Loads a .csv into a pandas dataframe with two columns: start and end. Currently only supports .csv files with two columns, containing start and end time.

More adaptive parser has to be implemented to handle multiple different time formats. Currently supported datetime format is HH:MM:SS.f

Parameters

- **filename** – str
- **filepath** – str

Returns

pd.DataFrame

CHAPTER

THREE

SLOTS

class slots.**Period**(*start: float*, *end: float*, *n_sats: int = 1*)

Period class handling time periods. A period consists of a start and end time. The n_sats counter will be increased, if two time periods overlaps, which means, that another satellite is visible during this time.

classmethod find_slots(*periods: deque*, *slots: list*)

Recursive algorithm to find overlapping periods.

If there is more than one period in the periods deque, the first two elements of the deque are taken from the deque. If these two periods are overlapping, they will be merged. This merged Period, with the overlapping time window, will then be appended to the left side of the deque and the function is called again with the updated deque. If the first two Periods do not overlap, the first of these two will be written written into the Slots list. The other will be appended back to the left side of the deque.

Once the deque is empty, all Periods, merged or not, should be in the Slots list.

The intuitive solution of comparing and merging overlapping periods emerges from a graphical solution of finding the longest vertical path through a set of time windows.

===

The graphic below shows the result after two iterations. Three Periods were merged due to the overlap in time windows. The vertical path from A to C is the longest possible vertical path originating from time slot A.

A==========|==========|A

B==========|==========|========B

C=======|==========|===C

Parameters

- **periods** – deque
- **slots** – list

Returns

overlap(*other:* Period) → bool

Checks if current period has an overlap with another period. :param other: Period :return: bool

True, if two periods overlap

overlap_period(*other:* Period) → *Period*

Generates the overlapping time period of two periods. It instantiates another Period object with an increased counter for the number of overlapping time slots merged in this period. :param other: Period :return: Period

PYTHON MODULE INDEX

INDEX