User's manual

1 — Last update: 2017/07/26

Dzikie Kuny

Table of Contents

1. Introduction

This document describes the Python_BST_Project application for generating commands (increase, decrease) for power control on stations and terminals to match target for signal strength. Each chapter describes different modules of the application that showcases specific functionality and provides instructions on how to install, run and use the application depending on operating system.

2. Python_BTS_Project application

This section is divided into two chapters:

2.1. Default behaviour

The basic features of the program include:

- accepting user input through stdin
- calculating response accordingly to Power Control Algorithm v2
- printing commands to terminal
- saving measurements in database

The extra features of the program include:

- configuration file for pre setting chosen parameters
- log file which generates information about running application
- saving data to the database via http server
- handover algorithm implementation which enables to switch to other terminal for achieving better signal quality
- graphs generation based on calculated data from power control algorithm

2.2. General Syntax

The developers of the application put a lot of effort for this program to require as little time and knowledge as possible. Therefore, the only file that requires your input is basic configuration file – **config_file.txt**. If you want to change the parameters for the algorithm please modify them in there before running the algorithm. The values are set after the “is equal to” sign and should be positive or negative integer. No commas, dots or quotes are allowed here – they may cause the program to crash.
Example of the correct values:
target = -75
hysteresis = 3
maxInc = 8
maxDec = 4
values = 4
missing = 1
http = 0

Apart from this, the basic syntax you need to know is the command line syntax in order to be able to select the correct input file and run the algorithm. Please see “Running the application” for more information.

3. Expected input

The expected input is .txt file with mixture of measurement, in columns:

- direction of transmission:
  DL, downlink, measurement comes from MS, concerns specific cell (S0, N1, N2…), commands are send for BTS of S0 cell only
  UL, uplink, measurements come from BTS, concern serving cell (S0) and mobile terminal (MS)
- cell identity: S0 (serving cell), Nx (neighbor cell, not used in PCA2)
- MS identity, e.g. MS222, MS667 etc.
- Signal strength e.g. -78 [dBm], correct values -45..-95
- Signal quality (0..5) representing quality measured by error rate, measured only for serving cell (S0)

Example:

DL	S0	MS776	-66	1

It is possible for some data or measurements to be missing. In this case the algorithm interpolates.

4. Expected output

The calculated output is formatted in the columns:

- direction of transmission:
  DL, downlink, commands are send for BTS of S0 cell only
  UL, uplink, for mobile terminal (MS)
- MS identity, e.g. MS222
- Command: INC (increase), DEC (decrease), NCH (No Change)
- Step for increase or decrease

Example:

DL	S0	MS776	INC	8
UL	S0	MS776	NCH	

The calculated output is being displayed in your terminal window, saved in the database and saved in .txt files, using MS identities as distinguishers.

5. Running the application

Please select your platform:

Windows

Linux

OsX

5.1. Windows

Installation and setup instructions:

Download the program files from https://github.com/peterb91/Python_BTS_Project.git, and save them on your local drive.

Open cmd line:
Ensure you have installed python 2.x and 3.x interpreters on your station (command: *where python*).
If python 2.x or 3.x is missing, download required version of python from: https://www.python.org/getit/ and install it following instructions on the website.
Ensure you have Flask installed (follow this guide: http://flask.pocoo.org/docs/0.12/installation/#windows-easy-install)
Ensure you have requests installed (command: *python3 -m pip install requests*)

Install additional program Gnuplot for graph creation:
download gp426win32.zip from the following site:http://sourceforge.net/projects/gnuplot/files/

Extract the file into “Name of folder”. Open “My Computer” and find the file wgnuplot. It should be somewhere like: C:/Documents and Settings/username/Name of folder/gnuplot/bin
click on the icon for gnuplot.exe and install the program. During the installation you should check the “Add application directory to your PATH environment variable” option.
now you can use gnuplot in cmd line using command: *gnuplot*

Go to the directory where all files were downloaded (command: cd directory_name_path eg. cd C:/Documents and Settings/username/Python_BTS_Project.

Before running the algorithm, you are able to modify basic configuration file – config_file.txt, where you can modify target value, hysteresis, maximum increase and max decrease values and other details, as well as you can set if the program should send the data to the database via http or not (it is much faster with http disabled). If you want to send data via http, please read this chapter:
Saving data to the database via http

Run the program in cmd console using command: *type file.txt | python3 source.py* (where “file.txt” is file with data from terminals and stations and “source.py” is running file included in program pack.

For generating graphs based on calculated data please use following command: *python2 chart_creation.py*

5.2. Linux

Installation and setup instructions:

Download the program files from https://github.com/peterb91/Python_BTS_Project.git, and save them on your local drive.

Open terminal:

Ensure you have installed python 2.x and 3.x interpreters on your station (command:_ whereis python_).
Ensure you have Flask installed globally (command:_ pip3 install flask_)
Ensure you have Requests installed (command: *pip3 install requests*)

If python 2.x or 3.x is missing, download required version of python from:" https://www.python.org/getit": https://www.python.org/getit/ and install it following instructions on the website. For linux – ubuntu installation use command:_ sudo apt-get install pythonx_.x (where x.x is version)

Install additional program for graph creation (command: *sudo apt-get install Gnuplot*)

Go to the directory where all program files are stored cd folder_name_path eg cd /usr/Python_BTS_Project/

Before running the algorithm, you are able to modify basic configuration file – config_file.txt, where you can modify target value, hysteresis, maximum increase and max decrease values and other details, as well as you can set if the program should send the data to the database via http or not (it is much faster with http disabled). If you want to send data via http, please read this chapter:
Saving data to the database via http

Run the program using command:_ cat file.txt | python3 source.py._

For generating graphs based on calculated data please use following command: *python2 chart_creation.py*

The most important is to make sure that file.txt exists and contains input data.

5.3. OsX

Installation and setup instructions:

Clone the repository" https://github.com/peterb91/Python_BTS_Project.git": https://github.com/peterb91/Python_BTS_Project.git to any directory (command: git clone "https://github.com/peterb91/Python_BTS_Project.git":https://github.com/peterb91/Python_BTS_Project.git)
Open terminal:
Ensure you python 2.x and 3.x interpreters installed on your station (command: *whereis python*).
Ensure you have Flask installed globally (command:_ pip3 install flask_)
Ensure you have Requests installed (command: *pip3 install requests*)

If python 2.x or 3.x is missing, download required version of python fromhttps://www.python.org/getit/ and install it following instructions on the website. If you are using homebrew you can use commands_ brew install python_ and *brew install python3*

Install additional program for graph creation – Gnuplot – from binary" http://ricardo.ecn.wfu.edu/pub/gnuplot/": http://ricardo.ecn.wfu.edu/pub/gnuplot/ or using homebrew: brew install gnuplot

Go to the directory where all program files are stored *cd folder_name_path* eg cd /usr/Python_BTS_Project/

Before running the algorithm, you are able to modify basic configuration file – config_file.txt, where you can modify target value, hysteresis, maximum increase and max decrease values and other details, as well as you can set if the program should send the data to the database via http or not (it is much faster with http disabled). If you want to send data via http, please read this chapter:
Saving data to the database via http

Run the program using command: *cat file.txt | python3 source.py* (where "file.txt" is file with data from terminals and stations and "source.py" is running file included in program pack).

For generating graphs based on calculated data please use following command: *python2 chart_creation.py*

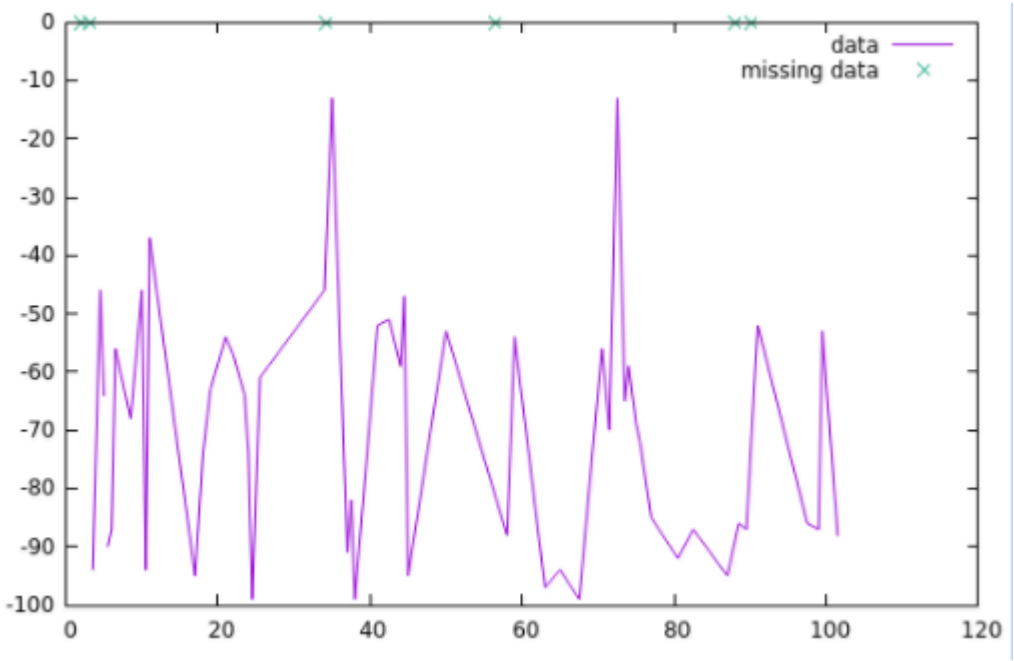
The most important is to make sure that file.txt exists and contains input data.

6. Additional features

This application contains some additional features tot be used if needed. Please navigate to the subchapter for full description:

6.1. Graphs generation

In order to create graph from input data for chosen MS identity you have to run graph_creation.py. Ensure you have installed Python 2.x and Gnuplot application. Please edit graph_creation.py file by updating lines 20 and 21: ‘data1’ and ‘data2’ with file name according to terminal name you want to see on graph. (e.g. data1=“ms_MS222.txt”; data2=“missing_MS222.txt”, where MS222 is MS identity name). The srcipt will create png file: chart.png with resolution 400×300 pixels. Chart example below:



On X axis is a time [sec], on Y axis: signal strength [dB], if there is some missing values for given MS identity name the green cross is drew with value 0.

6.2. Handover algorithm

Additional functionality added into power_management function which handles switching between terminals (six neighbors named N1, N2, N3, N4, N5 or N6). It calculates average from given last number of values from configuration file (“values”) and compares it to strength of best chosen neighbor cell. If neighbor performance is better than current terminal by more than 3 points it sends handover request (HOBC) to the standard output.

6.3. Configuration file

The text file contain set of value which are used for calculations of final commands.
User can set the values (by overwriting them) in config_file.txt before running the application.
config_file.txt:
target = -75
hysteresis = 3
maxInc = 8
maxDec = 4
values = 4
missing = 1
http = 0

Please, do not use extra characters – use integers only, without letters or quotation marks.

6.4. Log file

Log file called myapp.log is created after running script logs.py. In order to run logs.py you need Python 2.x or 3.x and python logging library. Log file contains date and time, and logging info messages.

6.5. Saving data to the database via http

This application enables saving data to the databes via http. To make it work make sure you have Flask and Requests installed (as described in this section). Then:
1. use configuration file and set http to 1:
http = 1
2. start local server by running http_server.py (python3 http_server.py) in a separate terminal window (this script must be running for the whole time). Then, when you have server running start application in a normal way.

If you want to change the name of the database created via http please navigate to http_server.py and change the path on the 6th line. If you are using directory outside tmp please make sure it exists first. Another thing is to restart the server every time you make some modifications in the file.

7. Authors and contact

In case you have any questions regarding the project and you can not find the answers in the project’s documentation please contact one of the developers:

Teresa.Bury[at]ust-global.com
Piotr.Bednarz[at]ust-global.com
Patrycja.Brzeska[at]ust-global.com
Katarzyna.Fedevych[at]ust-global.com

8. Copyrights

No access rights required for the program. The program is open source software.