Санкт-Петербургский политехнический университет Петра Великого
Институт прикладной математики и механики
Кафедра «Прикладная математика»

Отчёт по курсовой работе по дисциплине«Математическая статистика»

Выполнил студент:
Митенев Александр Владимирович
группа: 3630102/70201

Проверил:
к.ф.-м.н.,доцент
Баженов Александр Николаевич

Санкт-Петербург
2020г.

Содержание

Список иллюстраций

1 Постановка задачи

Есть массив данных.Эти данные несут информацию о температуре плазмы. Требуется сопоставить сигналы в 2 каналах и получить температуру объекта (некоторый аналог пирометра на миллионы К).

2 Реализация

2.1 Получение и обработка сигнала

Получением и обработкой сигнала занимается функция *makeSignals* Рассмотрим по этапам:

1. Считывание данных
2. Выделение области для анализа, если выставлен соответсвующий флагг. Иначе область не будет изменена.
3. Установление границ указанных пользователем, так же при присутсвии соответсвующих значений.
4. Применение фильтров к сигналу.

Замечание 1. По умолчанию будет произведено выделение области для анализа без ручного указания границ.
Замечание 2. Реализация считывания сигналов и фльтров для работы с ними находится в библиотеке pyglobus.

2.2 Получение данных калибровки

Данные связанные с калибровкой обрабатываются в файле calibration.py.
Основыные функциональности:

- Считывание из файла данных о калибровке и создание по ним таблицы калибровки.
- Сопоставление занчений с температурой.

2.3 Вычисление

Расчеты производятся в функции *process*
Рассмотрим по этапам:

1. Два сигнала приводятся к единому промежутку.
2. Деление сигналов.
3. Получение результатов через таблицу калибровки.

2.4 Представление результатов

Результаты можно увидеть в текстовом файле или же вывести график. Запись в файл происходит всегда, а графическое представление может управлятся пользователем.

3 Результаты

Дальнейшие результаты соответствуют данным из файла sht38515.sht

3.1 Обработка сигнала

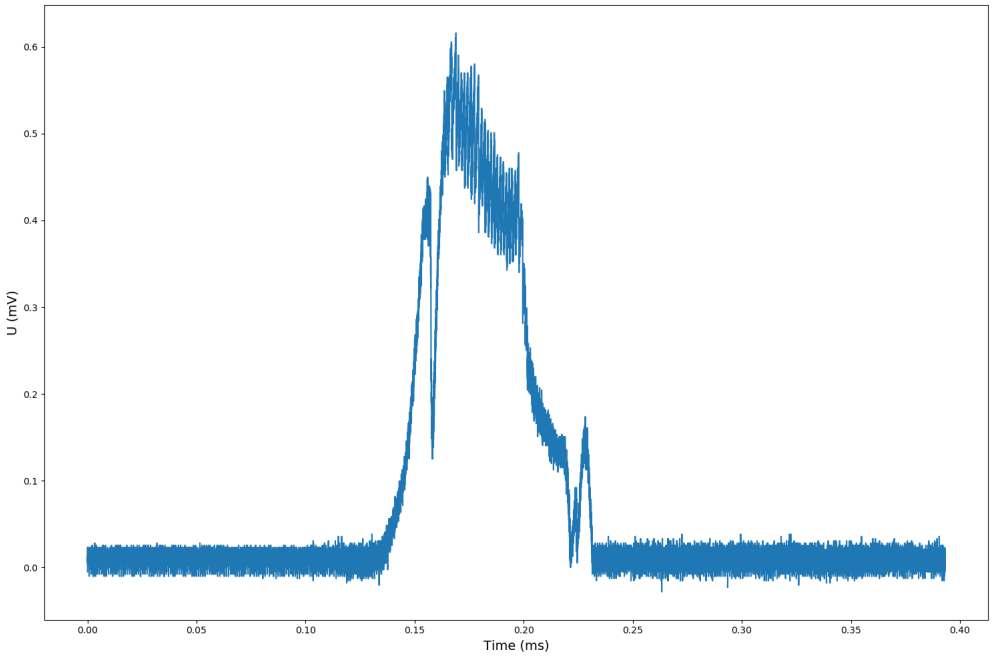

Рис. 1: Входной сигнал (sxr 80 mkm)

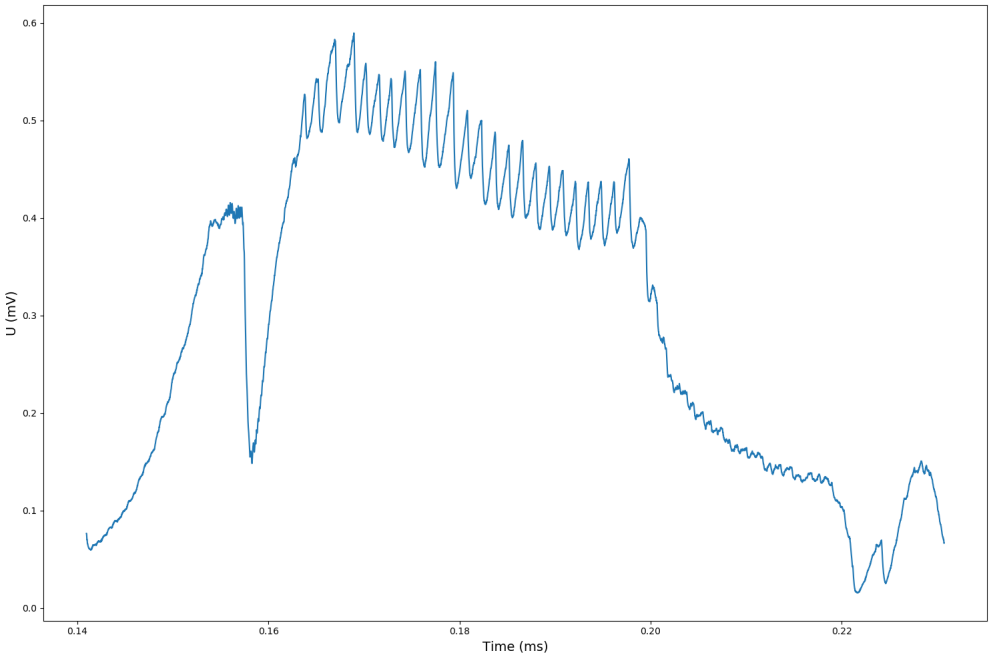

Рис. 2: Сигнал (sxr 80 mkm) после обработки по умолчанию

3.2 Значение температуры

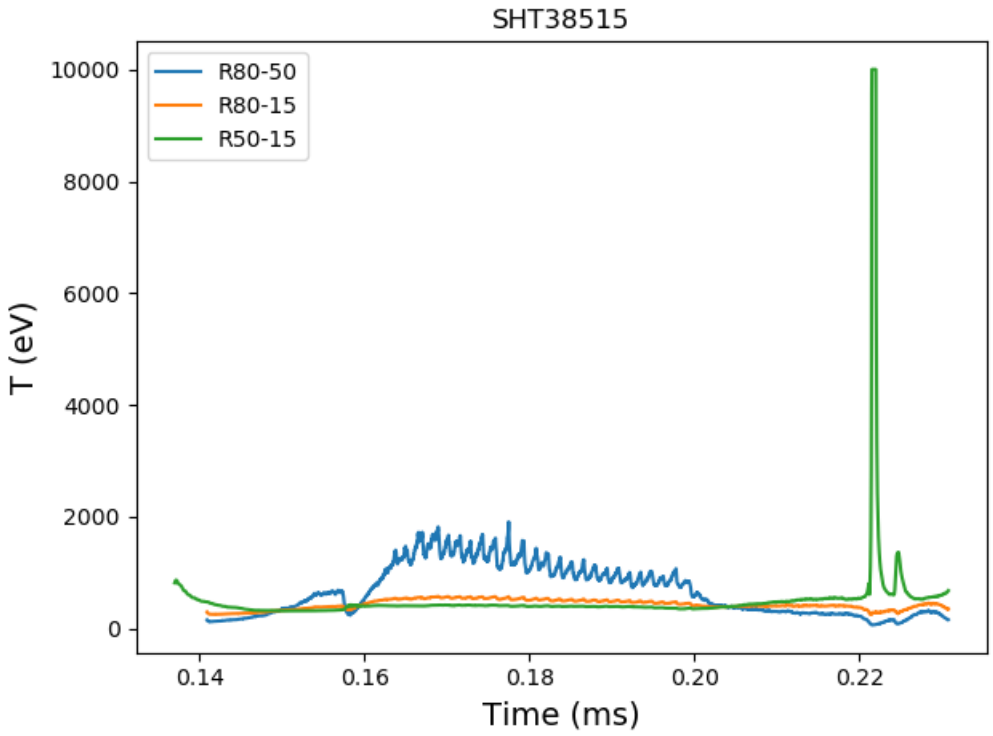

Рис. 3: Результаты

4 Варианты использования

Этапы описанные в пункте "Реализация"выполняются в функции *mainProcess* с данной сигнатурой:

def mainProcess(shtNum, nums, ROI=True, borders=False, left=LEFT, right=RIGHT, graphics=False)

- shtNum - номер sht файла (в представленных результатах использовался "38515")
- nums - массив с номерами (могут быть 80, 50, 27, 15) из названий сигналов, например, "SXR 80 mkm". Результаты собраны для значений

[80, 50, 15].

- ROI - булевская переменная, отвечающая за выделение области для анализа (region of interest) у сигнлов. По умолчанию выставлена в True.

- borders - булевская переменная, отвечающая за выставление границ промежутка, указанных в следующих двух переменных. По умолчанию False.

- left - значение левой границы промежутка. По умолчанию LEFT=0.14

- right - значение правой границы промежутка. По умолчанию RIGHT=0.2

- graphics - булевская переменная, отвечающая за вывод графика на экран. По умолчанию False.

Разберем наглядно значение этих переменных.

- ROI = True, borders = False (по умолчанию)

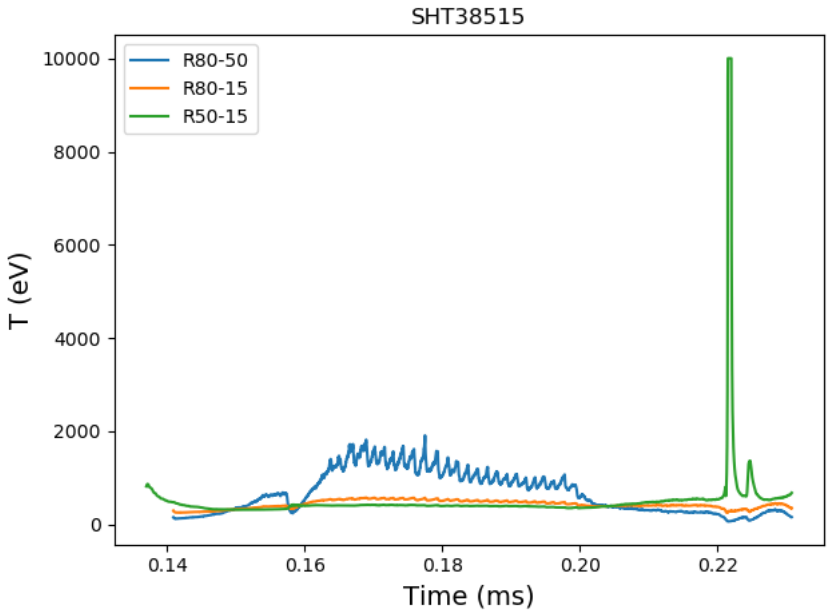

Рис. 4: Результаты (по умолчанию)

- ROI = True, borders = True, left = 0.14, right = 0.2

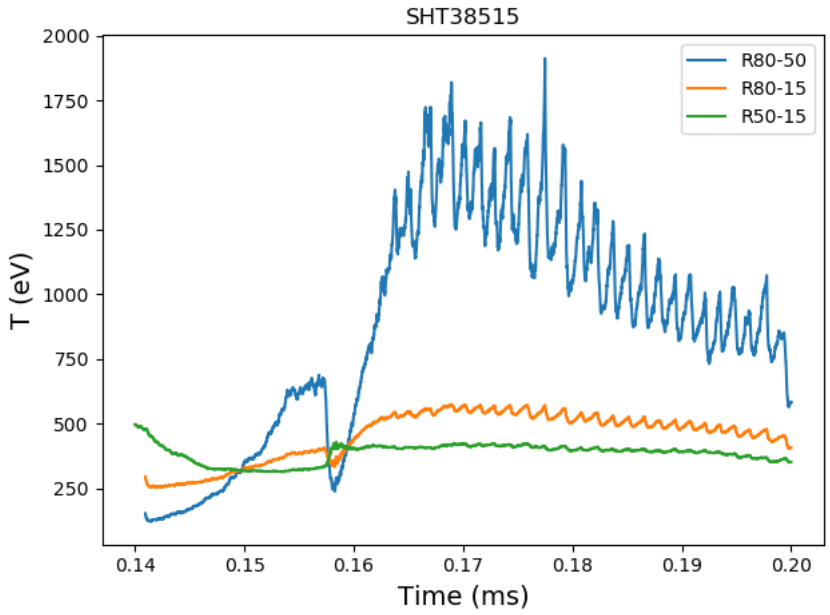

Рис. 5: Результаты (ROI = True, borders = True, left = 0.14, right = 0.2)

Заметим, что в данном случае от ROI ничего не зависело, т.к. вручную установленный промежуток полнстью входил в границы области для анализа.

- ROI = False, borders = False

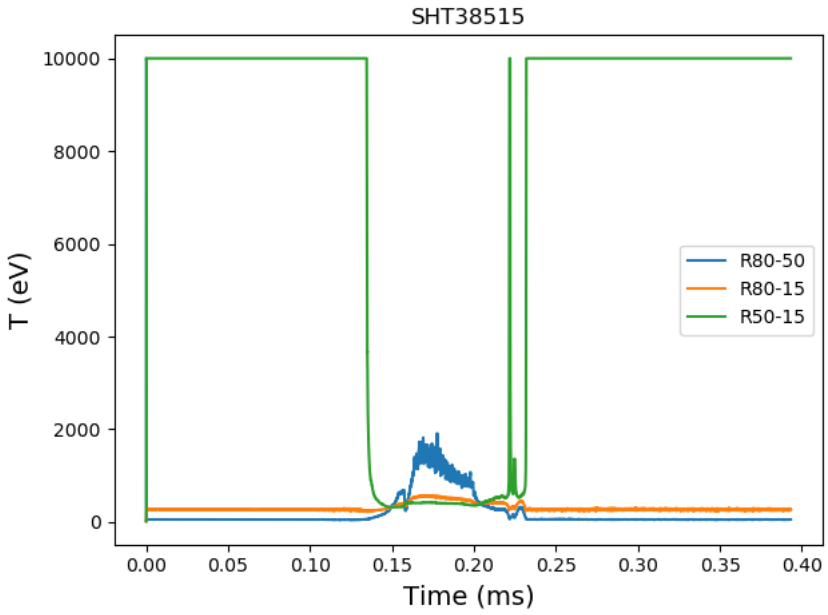

Рис. 6: Результаты (ROI = False, borders = False)

- ROI = False, borders = True, left = 0.14, right = 0.3

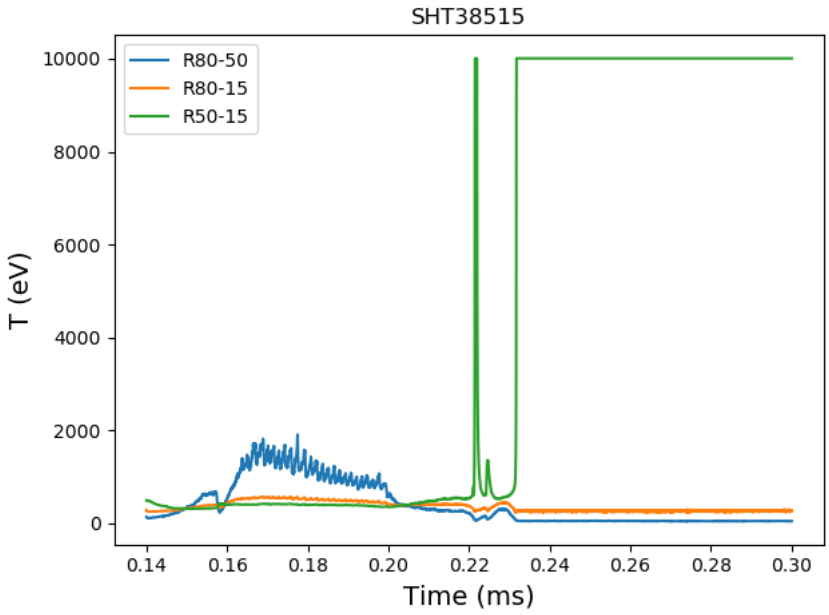

Рис. 7: Результаты (ROI = False, borders = True, left = 0.14, right = 0.3)

Подводя итог, можно сказать, что, если ROI=True, то границы интервала будут выставленны автоматически. borders=True добавляет возможность ручного управления границами с помощью переменных left и right. При использовании комбинации будет выбрана наибольшая левая граница и наименьшая правая.

5 Обсуждение

Полученные результаты совпадают с результатами работы программы globus2018, из чего можно сделать вывод о правильной работы программы.

6 Приложения

- Репозиторий с исходным кодом: https://github.com/mitenevav/Mathematical-statistics/tree/project
- Реализация pyglobus: https://github.com/dev0x13/globus-plasma