**Быстрый старт**:

В основной папке проекта содержатся две исполняемые программы **Processing** и **loop\_modelling**. Для быстрого старта следует запустить **Processing**, выбрав эксперимент с помощью переменной data. После того, как в workspace появится структура feloop запускаем **loop\_modelling**. Если после исполнения **Processing** не закрывать plot петли, то модельная петля построится прямо на графике с экспериментальной.

На данный момент хорошей показательной петлей является петля из Results\_2022\_PMN20PT\_PZT4\_1 с file\_number = 418. Она выставлена по умолчанию.

**Processing**

**Описание работы:**

В начале программы выбираются данные (переменная data) для работы из 5 предложенных вариантов (выбрать один, остальные закомментировать). В зависимости от переменной data, программа загружает данные и строит петли по одному из пяти блоков, разделённых перегородками следующего вида: **%-----**. Почти в каждом блоке данные загружаются уникально, т.к. данные не стандартизированы, и структура петель везде разная. В каждом блоке есть возможность построения петель.

Во всех блоках, кроме блока для data = 'Results\_2022\_PMN20PT\_PZT4\_1/' данные загружаются с помощью функции load\_loops\_files (data). Исполнение данной функции требуется лишь при первом запуске блока, после чего все необходимые данные для построениея петель лежат в Workspace и строчку с фунцией можно закомментировать. Эти данные содержат все измеренные петли при всех температурах и частотах для выбранного эксперимента.

В блоке data = 'Results\_2022\_PMN20PT\_PZT4\_1/' следует выбрать характеристики образца Sample.H и Sample.S (толщина и площадь образца соответственно) в зависимости от того, какой эксперимент строится (PMN или PZT). Подробнее про эксперименты см. файл ‘Структуры данных Results\_2022\_PMN20PT\_PZT4\_1'

Каждый блок в конечном итоге формирует структуру feloop, содержащую поля E и P. Эта структура должна лежат в Workspace при запуске **loop\_modelling**.

**Возможные улучшения:**

- Сделать разделение данных для Results\_2022\_PMN20PT\_PZT4\_1 на PZT и PMN в отдельные папки

- Сделать так, чтобы функция load\_loops\_files (data) выполнялась лишь один раз при пустом Workspace

- Для каждого блока сделать **удобный** вывод информации об образце и эксперименте (на каких частотах и температурах проводились измерения, амлитуда поля)

**loop\_modelling**

**Описание работы:**

В начале программы feloop загружается в функцию **Data\_for\_hysterons(feloop)**, которая выдаёт массивы двух порогов переключения (X,Y) и уровеней поляризации (Ps) гистеронов, используемых для моделирования данной петли (подробнее работа функции будет описана ниже).

Далее, в зависимости от выбранных данных, следует выбрать параметры образца Sample.area и Sample.thickness. Есть также возможность включения и отключения негативных эффектов (токи утечки, диэлектрические потери и шумы) с помощью переменной Sample.negative\_effects.

Далее выбираются характеристи треугольного сигнала для моделирования амплитуда и период Amp и T\_period. Функция voltage\_triangle ответственна за модельную генерацию треугольных импульсов.

Далее функция experiment (Sample, FEset, waveform) выдаёт нужные данные для построения модельной петли: field и P\_int.

В теле программы можно выбрать, какую петлю строить: обычную, по МДВ или токовую.

**Возможные улучшения:**

- Сделать автоматискую подгрузку данных об образце

- Модельные токовые петли не соответствуют реальным, особенно при увеличении периода измерения в реальном эксперименте (модельные сильно выше реальных).

- Функция Experiment была написана для усреднения данных при генерации рандомных гистеронов. Надо оформить покрасивее или вообще убрать.

- Разобраться с периодом модельного поля. Сейчас слишком медленно строятся петли на больших периодах. На малых периодах петли недетальные.

**Data\_for\_hysterons**

**Описание работы:**

На вход поступает петля feloop, которую хотим промоделировать. В начале часть данных обрубается, чтобы не было вторых проходов, где поляризация не меняется. Затем берём производную от поляризации, встроенной в матлаб функцией diff – получаем токовые петли. Затем встроенная функция movmean помогает избавится от шумов в токовых петлях. Параметр m отвечает за гладкость – чем он больше, тем меньше шумов в исходных петлях. Этот параметр выбирается автоматически исходя из размера массива E.p.

Далее с вероятностью 50% выбирается максимум тока(поляризации) либо на отрицательной ветви, либо на положительной. После этого ищется максимум на противоположной ветви, в зависимости от выбранной на первом шаге и оба максимума сравниваются: меньший из максимумов записывается в переменную Final\_P и её значение отнимаются от каждого максимума. После этого в переменные X, Y (пороги переключения гистерона) записываются текущие поля E при которых находились максимумы, а в переменную Psat (поляризация гистерона) записывается текущее P\_final. После этого начинается следущая итерация. Процесс продолжается до тех пор, пока все максимумы не будут “стравлены” до нуля, чему соответствует условие Final\_P == 0.

Таким образом функция разбирает петли по ступенькам, величина которых соответствует величине максимумов в токовых петлях. Эти ступеньки и есть гистероны.

**Возможные улучшения:**

**-** Подумать как лучше автоматизировать задание m для movmean

- Подумать, почему, если не рандомизировать выбор первого максимума, токовая петля не всегда стравливается до конца.