Описание головного класса OTS\_DC\_HeterogRail\_call

В комментариях в данном классе используются следующие обозначения:

Р - рельсы данного пути, рассматривается как длинная линия

ЭПС - ЭПС по данному пути

ФОТ - фидера обратного тока (отсасывающие фидера) подключенные к рельсам данного пути

ЗАЗ - заземлители к рельсам данного пути (имеет смысл учитывать заземлители с малым сопротивлением заземления менее 10 Ом)

МПС - междупутные соединители, условное название – фактически это сопротивление, соединяющее две любые точки каких любо путей (длинных линий)

МКР – метод конечных разностей, сеточный метод дискретизации дифференциальных уравнений (в данном случае уравнение длинной электрической линии) по пространству (здесь вдоль пути)

Единицы измерения в классе:

координаты и длины – км;

напряжение – В;

ток – А;

сопротивление (обычное/волновое) – Ом;

проводимость – См;

коэффициент влияния – Ом;

мощность – Вт;

продольное сопротивление рельсов – Ом/км;

переходное сопротивление рельс-земля – Ом\*км.

При инициализации экземпляра класса OTS\_DC\_HeterogRail\_call указывается в качестве параметров конструктору количество путей и через запятую количество сеток как **int**.

|  |  |
| --- | --- |
| Подклассы | Описание |
| track  представлен массивом tracks[] | Содержит параметры ОТС (рельсов) данного пути. В них же задаются точки подсоединения ФОТ, ЭПС, ЗАЗ, точечных продольных сопротивлений для каждого пути, волновое сопротивление в начале и в конце рассматриваемого участка пути. Также класс хранить внутренние параметры необходимые для расчёта МКР.  Нумерация путей в классе условно соотносится с индексом пути в массиве tracks[]. Т.е. пути условно нумеруются с 0 до N-1 включительно, здесь N это количество путей всего. Обращение к конкретному пути происходит через вызов соответствующего элемента массива tracks[].  Каждый путь (элемент массива tracks[]) должен быть соотнесен с одной конкретной сеткой (элементом массива meshes[]). |
| mesh  представлен массивом meshes[] | Содержит параметры конечно разностной сетки для дискретизации вдоль оси пути длинных электрических линий рельсов, а также внутренние процедуры для привязки элементов пути к сетке и расчета параметров схемы замещения путей. По сути, сетка задает систему координат вдоль оси пути, и может быть общей для нескольких путей.  Нумерация сеток в классе условно соотносится с индексом сетки в массиве meshes[]. Т.е. сетки условно нумеруются с 0 до N-1 включительно, здесь N это количество сеток всего. Обращение к конкретной сетке происходит через вызов соответствующего элемента массива tracks[].  С одной сеткой может быть соотнесено произвольное количество путей. |
| computing\_settings | Содержит свойства, определяющие настройки численных методов расчёта |
| err\_and\_mes | Содержит методы для считывания ошибок и сообщений, возникающих при расчёте температуры провода |

|  |  |
| --- | --- |
| Свойства | Описание |
| mps | **double[][]** – двумерный массив МПС.  Каждая строка содержит массив из пяти элементов для текущего МПС:  первый – номер пути, к которому присоединяется условно начальная точка подключения МПС;  второй – номер пути, к которому присоединяется условно конечная точка подключения МПС;  третий – координата начальной точки подключения МПС к пути в его системе координат, [км];  четвертый – координата конечной точки подключения МПС к пути в его системе координат, [км];  пятый – сопротивление МПС, [Ом];  Системы координат (а значит и сетки) двух путей соединяемых МПС могут быть разными. Номер путей МПС внутри класса приводятся к типу **int**. Номера путей соответствуют индексам путей в массиве tracks[].  По умолчанию пустой массив. Как правило, МПС расположены в точках подключения ФОТ и дополнительно могут быть в нескольких местах на межподстанционной зоне. При ответвлениях необходимо обязательно задавать МПС, соединяющий начало отходящей ветки и конкретную точку основной ветки, к которой подсоединена отходящая ветка. |

| Методы | Описание |
| --- | --- |
| calc\_ots() | Метод для расчёта мгновенной схемы ОТС. Происходит на основе заданных свойств. Данный метод необходимо запускать каждый раз, как только меняются исходные данные, т.е. новая мгновенная схема или изменившиеся параметры рельсов. И только после следует этого считывать результаты.  *Входные параметры:*  ***(основной метод)*** Нет  ***(переопределённый метод)*** **double**[] массив поисковых токов в МПС. Токи в таком массиве задаются в таком же порядке как заданы их параметры в массиве mps в головном классе. Итерационный расчёт начинается с данных начальных значений токов. В основном методе эти начальные токи равны нулю.  Переопределённый метод может быть полезен, если в основном методе расчёт не сошёлся, чтобы начать уже не от нулевых значений.  *Выходные параметры*:  Значение **boolean** равное true, если расчёт завершился корректно, и false в противном случае.  *Возможные ошибки и исключения*:  В случае возникновения ошибки при выполнении метода, ее наличие и соответствующее сообщение об этом будет находиться в подклассе err\_and\_mes. |
| init\_OTS() | Метод инициализирует ОТС. Происходит проверка корректности ввода исходных данных. Создаётся расчётная сетка, заполняются матрицы трехдиагональные ленточные для каждого пути, матрица влияющих коэффициентов напряжений в МПС от токов в них.  Данные метод необходимо вызывать обязательно перед серией расчётов, когда уже заданы параметры рельсовой цепи (продольное сопротивление, переходное рельс-земля), заданы МПС и ЗАЗ для каждого пути. При изменении ФОТ или ЭПС этот метод не требует запуска, т.к. параметры самой ОТС не меняются при этом, меняется нагрузка.  Но если, к примеру, произошло изменение электрических параметров рельсов или МПС, ЗАЗ, то метод перед расчётом необходимо заново вызвать.  *Входные параметры:*  Нет  *Выходные параметры:*  Нет |
| get\_U\_rail() | Геттер для вывода напряжения в рельсах первого и второго пути. Необходимо запускать данный метод только после расчёта, т.е. после выполнения calc\_ots().  *Входные параметры*:  1 **int** – номер пути, для которого нужно вывести параметр.  2 **double[]** – массив координаты точек [км]. Если этот параметр не указан, то вывод по координатам узлов сетки  *Выходные параметры*:  Массив **double[]** содержащий напряжение в рельсах в [В] по первому или второму пути в указанных координатах.  *Возможные ошибки и исключения*:  Если после метода calc\_ots() исходные данные оказались с ошибкой или этот метод не выполнялся ни разу, выводится пустой массив. В этом случае наличие ошибки и соответствующее сообщение об этом будет находиться в подклассе err\_and\_mes.  Если при указании массива координат они выходят за пределы расчётной сетки, то метод выводит Null |
| get\_I\_rail() | Аналогично get\_U\_rail() но выдаёт ток в рельсах в [А] |
| get\_I\_grnd() | Аналогично get\_U\_rail() но выдаёт ток в земле в [А] |
| get\_I\_poisk() | Геттер для вывода значения токов в поисковых точках в МПС.  *Входные параметры*:  Нет  *Выходные параметры:*  Массив типа **double[]** содержащий значения токов в МПС  *Возможные ошибки и исключения*:  Если после метода calc\_ots() исходные данные оказались с ошибкой или этот метод не выполнялся ни разу, выводится пустой массив. В этом случае наличие ошибки и соответствующее сообщение об этом будет находиться в подклассе err\_and\_mes. |
| get\_P\_ots() | Геттер для вывода значения мгновенной мощности потерь в ОТС в Вт  *Входные параметры*:  Нет  *Выходные параметры:*  Величина мощности потерь типа **double** в Вт.  *Возможные ошибки и исключения*:  Если после метода calc\_ots() исходные данные оказались с ошибкой или этот метод не выполнялся ни разу, выводится значение -1. В этом случае наличие ошибки и соответствующее сообщение об этом будет находиться в подклассе err\_and\_mes. |
| set\_I\_poisk\_no\_call() | Сеттер задает значения в МПС без выполнения непосредственного решения системы уравнений, т.е. в отличие от метода calc\_ots() расчёт выполняется гораздо быстрее, т.к. никакого итерационного поиска токов не происходит. Позволяет воспроизвести состояние ОТС фактически сразу без расчёта, если каким-то образом известны токи МПС.  Удобно сохранять результат расчёта ОТС через get\_I\_poisk(), а после воспроизводить через данный сеттер.  *Входные параметры:*  **double**[] массив токов МПС. Токи в таком массиве задаются в таком же порядке как заданы их параметры в массиве mps в головном классе.  *Выходные параметры*:  Значение **boolean** равное true, если расчёт завершился корректно, и false в противном случае.  *Возможные ошибки и исключения*:  В случае возникновения ошибки при выполнении метода, ее наличие и соответствующее сообщение об этом будет находиться в подклассе err\_and\_mes. |

Подкласс track

| Свойство | Описание |
| --- | --- |
| r | **double[][]** – двумерный массив функция погонного сопротивления рельсов вдоль пути. В каждой строке два значения: 1- координата км, 2 -величина погонного сопротивление рельсов Ом/км до данной координаты.  По умолчанию не задан. Необходимо обязательно задавать. |
| rp | **double[][]** – двумерный массив функция переходного сопротивления рельсы-земля вдоль пути. В каждой строке два значения: 1 – координата км, 2 – величина переходного сопротивления Ом\*км до данной координаты.  По умолчанию не задан. Необходимо обязательно задавать. |
| R\_tch | **double[][]** – двумерный массив таблица сосредоточенных точечных сопротивлений в рельсах на данном пути. В каждой строке два значения: 1- координата расположения км, 2 -величина сосредоточенного сопротивления Ом. (к примеру: разрыв рельсовой сети, дефектный неизолированный стык с высоким сопротивлением, дефект в средней точке ДТ с повышенным сопротивлением).  По умолчанию пустой массив. Можно не задавать. |
| fot | **double[][]** – двумерный массив ФОТ подключения соответственно к данному пути.  Каждая строка содержит массив из двух элементов для текущего ФОТ:  первый – координата точки подключения к рельсам, [км];  второй – величина тока ФОТ, [А];  Величина тока ФОТ принимается положительной, если ТП в выпрямительном режиме, т.е. ток затекает из рельсов на ТП через ФОТ.  По умолчанию пустой массив. Необходимо обязательно задавать. |
| eps | **double[][]** – двумерный массив ЭПС соответственно на данном пути.  Каждая строка содержит массив из двух элементов для текущего ЭПС:  первый – координата точки нахождения на рельсах, [км];  второй – величина тока ЭПС, [А];  Величина тока ЭПС принимается положительной, если состав потребляет ток из контактной сети, т.е. ток втекает в рельсы из ЭПС.  По умолчанию пустой массив. Необходимо обязательно задавать. |
| zaz | **double[][]** – двумерный массив ЗАЗ соответственно наданном пути.  Каждая строка содержит массив из двух элементов для текущего ЗАЗ:  первый – координата точки подключения к рельсам, [км];  второй – сопротивление заземления ЗАЗ, [Ом];  Имеет смысл учитывать заземлители с малым сопротивлением заземления менее 10 Ом. Сопротивление множества высокоомных заземлителей целесообразно учитывать через переходное сопротивление рельсы-земля, путём пересчёта вторичных параметров длинной линии.  По умолчанию пустой массив. Можно не задавать. |
| Rv0 | **double** – величина волнового сопротивления в начале пути, т.е. в первом узле сетки [Ом].  По умолчанию = – 1. Это означает, что волновое сопротивление будет рассчитано автоматически при выполнении процедуры init\_OTS() в базовом классе на основе r и rp для данного пути.  При необходимости следует изменить это значение. Волновое сопротивление условно имитирует бесконечно продолжающуюся однородную линию за этой точкой. Если фактически путь в этой точке начинается/заканчивается, то волновое сопротивление должно быть большим по величине. К примеру, если этот путь отходящая ветка, то волновое сопротивление в начале линии должно быть большой величины. Другой случай это тупик, когда дальше даже не электрифицированного пути далее нет. |
| Rvn | **double** – величина волнового сопротивления в конце пути, т.е. в последнем узле сетки [Ом].  Аналогично Rv0 |
| num\_mesh | **int** – номер сетки (фактически индекс элемента в массиве meshes[]) с которой ассоциирован данный путь.  По умолчанию равен 0. Обязательно надо указывать для каждого пути. |

Точки ЭПС и ФОТ являются точками с заданным током. ЗАЗ при дискретизации по сетке естественным образом входят в трёхдиагональную ленточную матрицу. ЗАЗ в этом случае не являются поисковыми точками. Только МПС это поисковые по току точки. Расчёт ОТС сводится к нахождению тока в этих элементах.

Подкласс mesh

| Свойство | Описание |
| --- | --- |
| X\_beg, X\_end | **double** – начальная и конечная координата участка расчёта км.  По умолчанию 0 и 20 км. |
| dX | **double** – шаг сетки по длине в км.  По умолчанию 0,1 км. |

| Методы | Описание |
| --- | --- |
| get\_X() | Геттер возвращает массив координат узлов сетки в км.  *Входные параметры:*  Нет  *Выходные параметры*:  **double[]** – массив координат всех узлов сетки в км. |

Подкласс computing\_settings

|  |  |
| --- | --- |
| Свойство | Описание |
| convergence\_U | **double** – допустимая невязка величины напряжения [В], усреднённая по всем точкам граничного условия, в которых происходит итерационный поиск величины тока втекающего в эти точки. Эти точки МПС, ЗАЗ1 и ЗАЗ2  По умолчанию равно 0.01 |
| max\_number\_iterat | **int** – максимальное число итераций при расчёте величины тока в поисковых точках методом итераций по величине невязки напряжения.  По умолчанию равно 1000 |
| initial\_damping\_factor | **double** – начальное значение шага в методе Ньютона при расчёте величины тока в поисковых точках через решение системы алгебраических уравнений.  По умолчанию равно 0.7 |

|  |  |
| --- | --- |
| метод | Описание |
| get\_current\_state\_solver | Геттер возвращает текущее состояние решателя. Эти данные обновляются при выполнении метода calc\_ots()  *Входные параметры*:  Нет  *Выходные параметры:*  массив **double[3]** содержащий 3 элемента  1 – количество выполненных итераций.  2 – среднее значение невязки по напряжению на всех поисковых точках [В] на последней итерации.  3 – **double** значение коэффициента распространения в рельсах первого пути величина коэффициента демпфирования в методе Ньютона на последней итерации |

Подкласс err\_and\_mes

|  |  |
| --- | --- |
| метод | Описание |
| get\_data\_error() | Геттер для считывания признака наличия ошибки во входных данных. К примеру, когда не заданы ЭПС или ФОТ. Данная ошибка может быть критичной.  *Входные параметры*:  Нет.  *Выходные параметры:*  Значение **boolean** признака ошибки во входных данных. Если true ошибка есть, false – ошибки нет. |
| get\_solver\_error() | Геттер для считывания признака наличия ошибки решателя в численных методах. К примеру, не достигнута сходимость в цикле. Данная ошибка не является критичной, но не гарантируется точность решения.  *Входные параметры*:  Нет.  *Выходные параметры:*  Значение **boolean** признака ошибки решателя. Если true ошибка есть, false – ошибки нет. |
| get\_messeg\_data\_error () | Геттер для считывания сообщения с текстом ошибки во входных данных. Сообщение возникает если get\_data\_error() возвращает true.  *Входные параметры*:  Нет.  *Выходные параметры:*  Значение String c с текстом ошибки во входных данных. |
| get\_messeg\_solver\_error () | Геттер для считывания сообщения с текстом ошибки решателя в численных методах. Сообщение возникает если get\_solver\_error() возвращает true.  *Входные параметры*:  Нет.  *Выходные параметры:*  Значение String c с текстом ошибки решателя в численных методах. |

Концепция работы с классом

1. Создать экземпляр класса, указав количество путей и сеток.
2. Через обращение к элементу массива tracks[] задать необходимые свойства рельсов каждого пути и обязательно задать массивы ЭПС и ФОТ, при количестве путей более одного также МПС. А если есть подходящие элементы для ЗАЗ и сосредоточенные точечные продольные сопротивления, то их тоже задать. Значения токов ФОТ и ЭПС, а также координаты ЭПС задаются для каждой мгновенной схемы на основании стандартного электрического расчёта. Для каждого пути следует указать соответствующую ему сетку из массива meshes[] по ее индексу. Если путь имеет явное начало и окончание (отходящая ветка, тупик), то соответствующему волновому сопротивлению в начале/конце линии присвоить большое значение.
3. В подклассе mesh задать параметры расчётной сетки: начальную и конечную координату и шаг (шаг по умолчанию 0.1 км, в большинстве случаев достаточно).
4. Инициализировать ОТС через метод init\_OTS(). При этом происходит проверка корректности ввода исходных данных. Создаётся расчётная сетка, заполняются матрицы трехдиагональные ленточные для каждого пути, матрица влияющих коэффициентов напряжений в поисковых точках от токов в них.
5. Запустить на расчёт ОТС через процедуру calc\_ots() при этом контролируя ее корректное завершение через булевую переменную, которую она возвращает. Если она вернула false, значит, есть ошибка. В этом случае необходимо обратиться к дочернему классу err\_and\_mes. Ошибка исходных данных критична, ошибка решателя нет, но точность не гарантируется. В случае, если имеется несходимость решателя, желательно уменьшить начальное значение коэффициента демпфирования, увеличить число итераций или увеличить невязку (прибегать в последнюю очередь) и повторить расчёт. В крайнем случаи ошибку решателя можно проигнорировать, если цифры токов и напряжений в «пределах разумного».
6. После того как расчёт пройдет без ошибок в первую очередь по свойству err\_and\_mes.data\_error необходимо воспользоваться сеттерами get\_U\_rail(), get\_I\_rail(), get\_I\_grnd(), get\_P\_ots(), get\_I\_poisk() для возврата величины тока и напряжения в рельсах, тока в земле и МПС. Данные величины необходимо последовательно обработать по всему графику движения: нахождения зон с повышенным потенциалам рельсов для коррозии и электробезопасности, выявления элементов и участков с большим эффективным током для проверки на нагрев и т.п.
7. Повторить расчёт для следующей мгновенной схемы, начиная с пункта 5, изменив только ЭПС, и токи ФОТ. МПС, ЗАЗ, параметры рельсовой цепи менять не надо. Если при другой схеме все-таки нужно изменить что-то из этого, то после изменения нужно заново инициализировать ОТС (пункт 4).

Всё состояние системы ОТС для каждой мгновенной схемы удобно сохранять в виде массива величин токов МПС через get\_I\_poisk(), а после при необходимости воспроизвести состояние без непосредственного расчёта через метод set\_I\_poisk\_no\_call(), задав в качестве аргумента данный массив. А также перед вызовом этого метода для каждого пути следует ввести массивы ЭПС, ФОТ и прочее соответствующие данной мгновенной схеме.

Это позволяет экономить объём памяти при сохранении результатов расчёта ОТС. Не нужно сохранять в памяти или на диске все массивы токов и напряжений вдоль пути для каждой мгновенной схемы. Эти массивы просто необходимо вызвать и последовательно обработать для каждой расчётной схемы для определения интегральных (средних, СКО, RMS) параметров и экстремумов для каждой точки пути.

Теория расчёта кратко

Сам расчёт напряжений в отличие от однородной бесконечно длинной линии для неоднородной линии (возможно конечной длинны) выполняется за счёт дискретизации электрической схемы рельсов каждого пути вдоль длины методом конечных разностей. Это фактически численный метод решения уравнений. Создается сетка, между узлами которой расположены продольные сопротивления рельсов, а на землю – сосредоточенны переходное сопротивление рельсов в области действия узла +ЗАЗ если он есть в данном узле.

Слева от крайне левого узла и справа от крайне правого соединены с землей волновые сопротивления. Таким образом получается полная электрическая цепная линия для каждого пути. В некоторых узлах (ЭПС, ФОТ, ЗАЗ) будут втекать токи. По сути получается линейная длинная электрическая схема. Ее логично рассчитать с помощью законов Кирхгофа. Каждый узел связан с двумя соседними: напряжение на данном узле связано только с потенциалами соседних узлов, а также в уравнения входят величины продольных сопротивлений и сопротивлений на землю, а еще величина тока (для ЭПС, ФОТ, ЗАЗ), втекающего в данный узел. Особые – это крайние точки у них с внешней стороны от линии узлов нет – там волновые сопротивления на землю.

Таким образом расчёт потенциалов каждого узла сводится к расчёту СЛАУ в матричном виде:

* матрица левой части – матрица проводимости;
* вектор неизвестных это – вектор напряжений в узлах;
* а вектор правой части – вектор втекающих в узлы токов.

Количество неизвестных равно числу узлов сетки. Важно заметить, что матрица левой части имеет ненулевые элементы только по главной диагонали, а также на одну диагональ выше и ниже ее. Получается СЛАУ с трёхдиагональной матрицей. Это логично т.к. протекание тока по длинной (и не длинной тоже) линии описывается уравнением Пуассона. Подобную СЛАУ решается методом прямой и обратной проходки. Здесь эффективно использовать запись матрицы в ленточной форме с точки зрения экономии памяти. Допустим 1 тыс. узлов при полной матрице потребует хранить массив в 1 млн значений цифр с плавающей запятой, а при ленточном представлении нужен массив только 3 тыс. цифр. Метод прямой и обратной проходки также хорошо работает с ленточной матрицей.

Тут стоит отметить, что МПС, ФОТ и ЗАЗ при дискретизации по сетке полностью соотносятся с один узлом, ближайшим по координате к данному элементу. А вот ЭПС, если его координата попала между двумя узлами сетки, дискретизуется в оба близлежащих узла. Ток ЭПС в этом случае делится между этими двумя узлами по расстоянию. Это сделано умышленно: предполагается, что при расчете по графику движения координаты МПС, ФОТ и ЗАЗ будут постоянными, а ЭПС меняться. Позволит более плавно описать процесс по времени.

Как и с аналитической методикой проблема в том, что токи поисковых элементов (поисковых точках) не известны заранее. В данном случае это только МПС. ЗАЗ уже дискретизуются по стеке в сопротивление заземления для узла и естественным образом попадают в трехдиагональную ленточную матрицу. В аналитической методике такое было невозможно.

Для расчёта токов МПС в этом случае составляется система уравнений по невязке напряжений, создаваемых в этих точках непосредственно на них. Тут возникают два условия:

1. Напряжения на МПС будут равны разности напряжений между начальной и конечной точкой подключения к путям.
2. Напряжения на МПС равны токи МПС умноженные на их сопротивления.

Токи МПС подбираются таким образом, чтобы разность напряжений на МПС, полученных по этим двум условиям (невязка), была меньше заданного значения. Т.е. по сути тоже решается система алгебраических уравнений только итерационным методом.

Для того чтобы решить данную систему для каждой схемы нужно рассчитать матрицу коэффициентов влияний напряжения в поисковых точках от величины тока в них (отдельно для каждого пути). Количество столбцов и строк такой матрицы равно числу МПС. Для расчёта каждого элемента матрицы коэффициентов влияния необходимо рассчитать схему с одним втекающим током только в данный МПС. Расчёт описанным выше методом прямой и обратной проходки с трёхдиагональной ленточной матрицей. Количество таких расчётов равно числу поисковых точек, т.е. МПС. Данная операция происходит при инициализации ОТС и на ее выполнение затрачивается большая часть времени процедуры инициализации. Но зато сокращается время расчёта каждой мгновенной схемы. Сейчас процедура инициализации введена и в новую версию класса для аналитического расчёта, с этой же целью.

Можно также отметить, что если токи МПС известны каким-то образом, то это позволяет быстро рассчитать все напряжения и токи узлов. Самой затратной процедурой при расчёте каждой мгновенной схемы (если инициализацию уже проведена) является, именно расчёт поисковых токов. Поэтому целесообразно для каждой мгновенной схемы сохранять результат расчёта токов в поисковых точках, для быстрого воспроизведения всего расчёта данной схемы в случае необходимости. Это также позволяет объём хранимой информации, т.е. для каждой мгновенной схемы достаточно запоминать только токи МПС.

После того как выполнена процедура расчёта мгновенной схемы с помощью геттеров выводим результаты в узлах и элементах (токи и напряжения в рельсах, ток в земле, токи в МПС, мгновенная мощность потерь в ОТС). Ток в узле равен среднему от токов продольных сопротивлений слева и справа от него. Ток в земле считается как накапливаемая сумма всех токов, стекающих в узлах через сопротивление на землю. Ток заземлителя при необходимости можно рассчитать «вручную» как напряжение в узле деленное на сопротивление заземлителя. Мгновенная мощность потерь в ОТС определяется как сумма произведений токов ФОТ и ЭПС (это внешние по отношению к ОТС токи в отличие от токов ЗАЗ и МПС) на напряжения в рельсах, в которых они расположены.

Пример работы с классом



На представленной схеме имеется главная трехпутная ветка (пути 1гл, 2гл и 3гл), отходящий от 1гл пути электрифицированный тупик длиной 7 км, а также отходящая двухпутная ветка от 2гл и 3гл. На участке расположены ЭПС, ФОТ имеется множество МПС. Главные пути 1гл, 2гл и 3гл имеют общую систему координат Xгл, однопутный тупик имеет систему координат Xотх1, а двухпутный отходящий участок – Xотх2.

Рассчитаем ОТС данной схемы с помощью класса OTS\_DC\_HeterogRail\_call и сравним его с расчётом в комсоле МКЭ.

/\*

Каждому реальному пути на схеме, включая пути отходящих веток, нужно присвоить номер.

Этот номер по сути является индексом в массиве путей. Поэтому нумерацию путей следует начинать с 0.

Номера путей давать в любом произвольном порядке от 0 до N-1. N - это всего путей. На схеме N=6

Условно примем, что:

-главные пути 1, 2 и 3 это пути 0, 1 и 2;

-однопутная отходящий электриф тупик это путь номер 3;

-двухпутная отходящая ветка отх2 будет иметь номера путей 4 и 5.

Сетки также как и пути нумеруются с 0 и по сути являются индексами для массива сеток.

Для данной схемы будет три сетки, что соответствует системам координат вдоль путей.

сетка 0 - пути 0, 1 и 2;

сетка 1 - путь 3

сетка 2 - пути 4 и 5.

\*/

// создадим класс ОТС с указанием конструктору количества путей и сеток

OTS\_DC\_HeterogRail\_call ots1 = **new** OTS\_DC\_HeterogRail\_call(6, 3); //создаём экземпляр класса

// в классе обращение к конкретным путям и сеткам идет в массивах для них по индексу. Индексация с 0.

// пути

// заполним массивы ФОТ

ots1.tracks[0].fot = **new** **double**[][]{{140.5, 2300}, {160.2, 2400}, {176.7, 3000}}; // по гл1 - в классе путь 0

ots1.tracks[4].fot = **new** **double**[][]{{10.2, 1400}}; // по отх ветке 2 путь1 - в классе путь 4

// заполним массивы ЭПС

ots1.tracks[0].eps = **new** **double**[][]{{149.0, 800}, {171.2, 3400}}; // по гл1 - в классе путь 0

ots1.tracks[1].eps = **new** **double**[][]{{156.7, 1900}}; // по гл2 - в классе путь 1

ots1.tracks[2].eps = **new** **double**[][]{{145.3, 1600}}; // по гл3 - в классе путь 2

ots1.tracks[5].eps = **new** **double**[][]{{12.1, 1400}}; // по отх ветке 2 путь2 - в классе путь 5

// заполним массивы МПС: точка 1 - начальняа точка подключения, точка2 - конечная точка подключения

// элемент массива МПС {номер пути точки1, номер пути точки2, координ точки1, координ точки2, сопротивлен}

ots1.mps = **new** **double**[][]{{0, 1, 140.5, 140.5, 0.9e-3}, {1, 2, 140.5, 140.5, 0.9e-3}, {0, 1, 151.5, 151.5, 1.5e-3}, // МПС по главным путям 1-3

{0, 2, 155.5, 155.5, 1.6e-3}, {0, 1, 160.2, 160.2, 1.4e-3}, {1, 2, 160.2, 160.2, 1.1e-3},

{1, 2, 167.2, 167.2, 1.4e-3}, {0, 1, 176.7, 176.7, 0.7e-3}, {1, 2, 177.1, 177.1, 1.8e-3},

{4, 5, 10.2, 10.2, 1.0e-3}, // МПС по отход2

{0, 3, 152.5, 0.0, 1.0e-5}, // соединение путь гл1 (путь 0 в классе) и однопутн отход тупик (путь 3 в классе)

{1, 4, 170.5, 0.0, 1.0e-5}, // соединение путь гл2 (путь 1 в классе) и отход2 путь1 (путь 4 в классе)

{2, 5, 170.5, 0.0, 1.0e-5}}; // соединение путь гл3 (путь 2 в классе) и отход2 путь2 (путь 5 в классе)

//заземлители и сосредоточенные продольные сопротивления учитывать не будем

// продольные сопротивления и переходные сопротивления рельс-земля каждого пути

**for** (**int** i = 0; i < 3; ++i) {

ots1.tracks[i].r = **new** **double**[][]{{189, 0.0254}}; // пути гл 1-3 продольное сопротивлен мало, переходное большое - путь бесстыковой в хорошем состоянии с высокой изоляцией от земли

ots1.tracks[i].rp = **new** **double**[][]{{179, 20}};

}

ots1.tracks[3].r = **new** **double**[][]{{8, 0.028}}; // электриф тупик путь звеневой, изоляция плохая

ots1.tracks[3].rp = **new** **double**[][]{{8, 1.1}};

**for** (**int** i = 4; i < 6; ++i) {

ots1.tracks[i].r = **new** **double**[][]{{15, 0.026}}; // двухпутная отходящая ветка отх2 -путь бестыковой изоляция норм

ots1.tracks[i].rp = **new** **double**[][]{{15, 10}};

}

// для однопутного тупика зададим большое волновое сопротивление в начале и в конце линии

ots1.tracks[3].Rvn = (1e6);

ots1.tracks[3].Rv0 = (1e6);

// для путей одходящей ветки2 зададим большое волновое сопротивление в начале линии

ots1.tracks[4].Rv0 = (1e6);

ots1.tracks[5].Rv0 = (1e6);

// сетки

// задание границ сетки

ots1.meshes[0].X\_beg = 138.0; // сетка 0 соответствует Хгл

ots1.meshes[0].X\_end = 180.0;

ots1.meshes[1].X\_beg = 0.0; // сетка 1 соответствует Хотх1

ots1.meshes[1].X\_end = 7.0;

ots1.meshes[2].X\_beg = 0.0; // сетка 2 соответствует Хотх2

ots1.meshes[2].X\_end = 14.0;

//присваиваем каждому пути номер сетки

ots1.tracks[0].num\_mesh = 0; // сетка 0 для пути 0, 1 и 2 (гл1, гл2, гл3)

ots1.tracks[1].num\_mesh = 0;

ots1.tracks[2].num\_mesh = 0;

ots1.tracks[3].num\_mesh = 1; // сетка 1 для пути 3 (однопутн тупик 7 км)

ots1.tracks[4].num\_mesh = 2; // сетка 2 для пути 4 и 5 (отход двухпутн ветка)

ots1.tracks[5].num\_mesh = 2;

ots1.init\_OTS(); // инициализируем ОТС

**if** (ots1.err\_and\_mes.get\_data\_error()) { // проверим если ошибка в исходных данных

*debugLog*(ots1.err\_and\_mes.get\_messeg\_data\_error()); // выведем в консоль

}

// рассчитаем ОТС

ots1.calc\_ots();

// для примера выведем ток и напряжение в рельсах вдоль всего пути номер 0 (1 гл) по координатам узлов сетки

**int** mesh\_track0 = ots1.tracks[0].num\_mesh; // номер сетки для пути 0

**double** X\_track0[] = ots1.meshes[mesh\_track0].get\_X(), // координаты узлов сетки для пути 0

U\_track0[] = ots1.get\_U\_rail(0), // напряжение в пути 0

I\_track0[] = ots1.get\_I\_rail(0); // ток в пути 0

// По аналогии можно сделать для других путей,

// можно вторым аргументам в геттерах указывать пользовательский массив координат, но в пределах сетки.

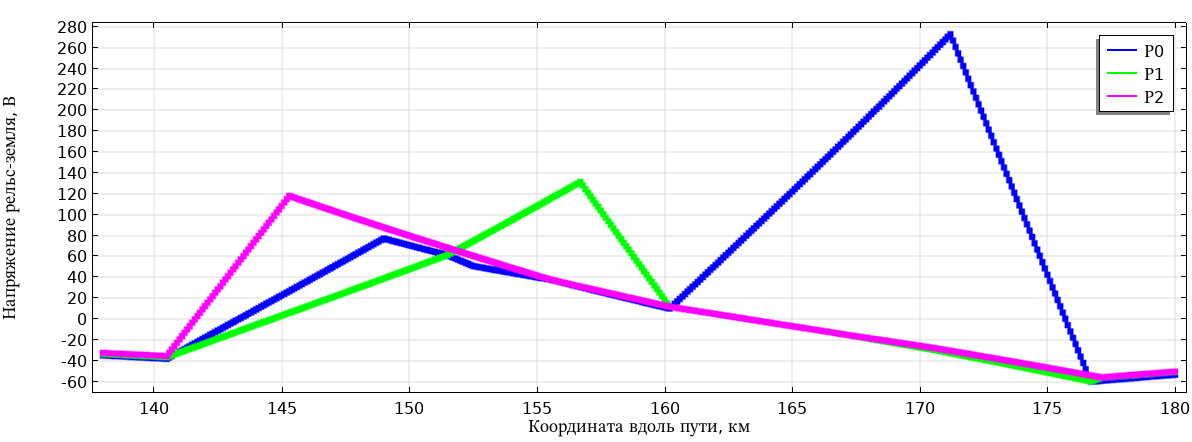
//Также при необходимости можно вывести токи в МПС

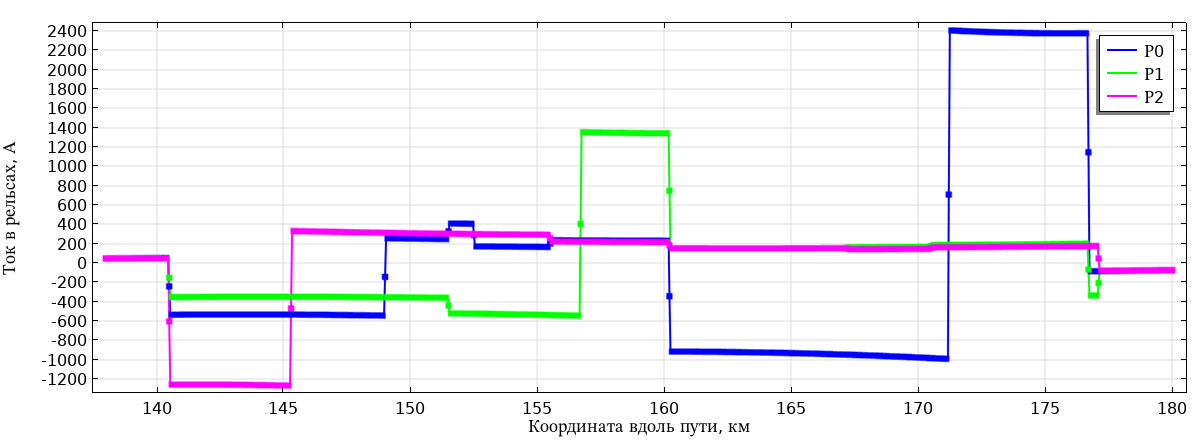
**double** I\_mps[] = ots1.get\_I\_poisk();

// и мгновенную мощность потерь в ОТС в Вт

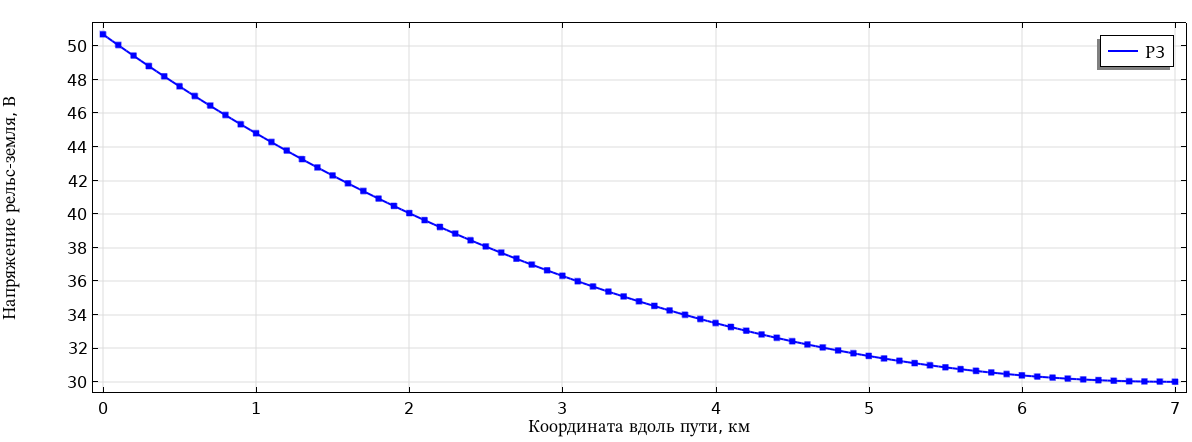
**double** P\_ots = ots1.get\_P\_ots();

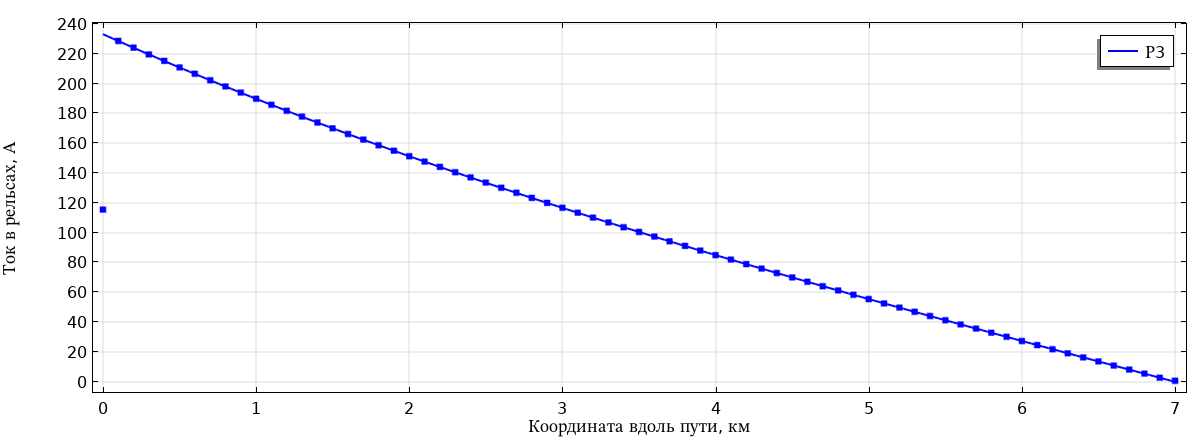
Ниже на рисунках показаны данные результаты в виде графиков. Точками это расчёт в классе, сплошная линия это расчёт в комсоле. Цвет означает конкретный номер пути.



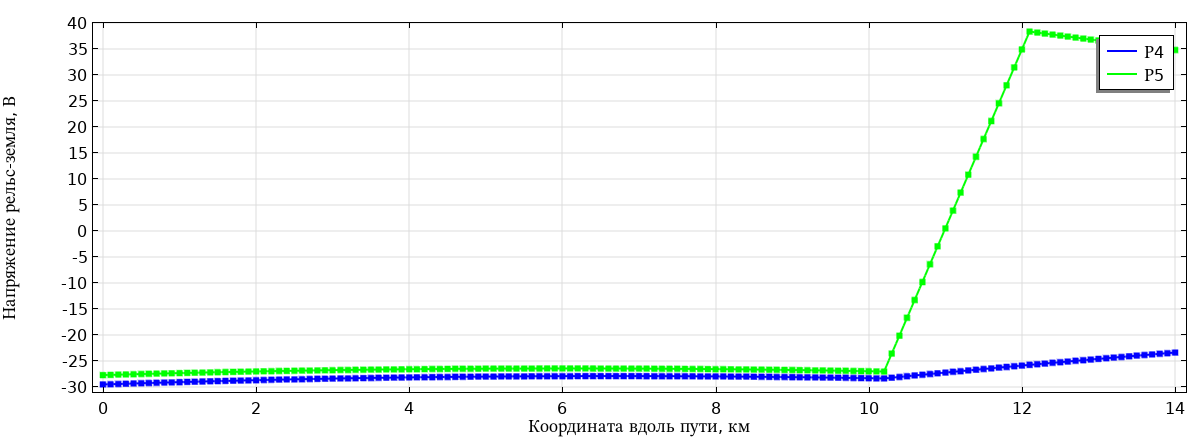


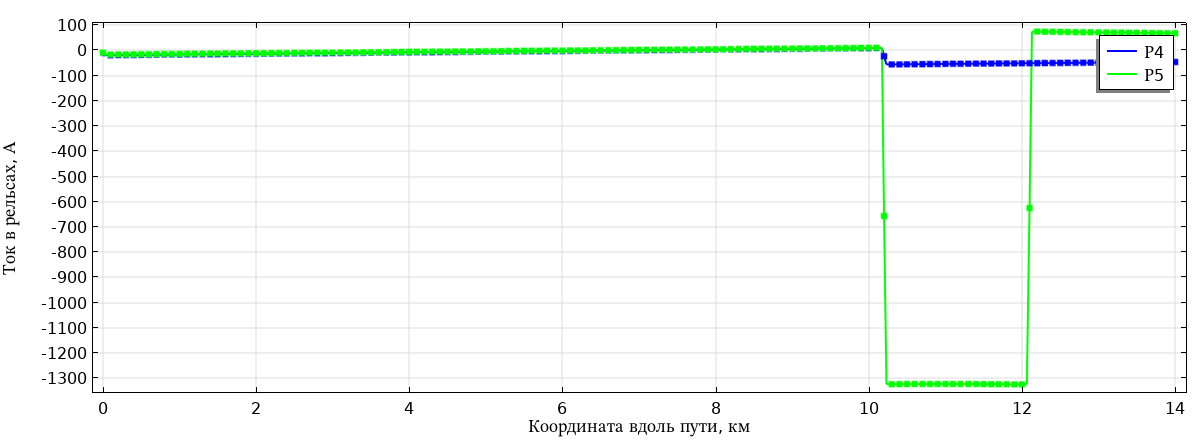
Напряжения и токи вдоль путей 0, 1 и 2 (гл1, гл2 и гл3)





Напряжения и токи вдоль пути 3 (однопутн отходящий тупик)





Напряжения и токи вдоль путей 4 и 5 (отходящая ветка 2)