

## ЧИСЛЕННЫЙ АНАЛИЗ СИГНАЛОВ ПЕРЕХОДНОГО ПРОЦЕССА В ЗАДАЧЕ ОПРЕДЕЛЕНИЯ МЕСТА ПОВРЕЖДЕНИЯ В ЛИНИЯХ ЭЛЕКТРОПЕРЕДАЧИ

*Иркагалиева Илюза Ильнуровна*, студент, КГЭУ, 420066, Россия, г. Казань, ул. Красносельская 51, e-mail: [irkagalieva2001@mail.ru](mailto:irkagalieva2001@mail.ru)

*Кузьмин Игорь Леонидович*, к.т.н., доцент, КГЭУ, 420066, Россия, г. Казань, ул. Красносельская 51, email: [kigor@list.ru](mailto:kigor@list.ru)

*Хузяшев Рустэм Газизович*, к.ф.-м. н., доцент, КГЭУ, 420066, Россия, г. Казань, ул. Красносельская 51, e-mail: [142892@mail.ru](mailto:142892@mail.ru)

**Аннотация.** В статье подробно описывается численный анализ сигналов переходного процесса в задаче волнового определения места повреждения на примере модального анализа. Прокомментирована программа на языке Python с использованием блок-схемы. Приведены результаты обработки осциллограммы сигнала переходного процесса.

**Ключевые слова:** сигнал переходного процесса, модальный анализ, волновой метод определения мест повреждения, интеллектуальный анализ данных.

## NUMERICAL ANALYSIS OF TRANSITION SIGNALS IN THE PROBLEM OF DETERMINING THE LOCATION OF DAMAGE IN POWER LINES

*Irkalgalieva Ilyuza Ilnurovna*, student, KSPEU, 420066, Russia, Kazan, st. Krasnoselskaya 51, e-mail: [irkagalieva2001@mail.ru](mailto:irkagalieva2001@mail.ru)

*Kuzmin Igor Leonidovich*, Ph.D., Associate Professor, KSPEU, 420066, Russia, Kazan, Krasnoselskaya st. 51, email: [kigor@list.ru](mailto:kigor@list.ru)

*Khuziyashev Rustem Gazizovich*, Ph.D. D., associate professor, KSPEU, 420066, Russia, Kazan, st. Krasnoselskaya 51, e-mail: [142892@mail.ru](mailto:142892@mail.ru)

**Annotation.** The article describes in detail the numerical analysis of transient signals in the wave problem of determining the location of damage using the example of modal analysis. A Python program commented using a flowchart. The results of processing the waveform of the transient signal are presented.

**Key words:** transient signal, modal analysis, wave method for determining the location of damage, data mining.

Задача определения места повреждения (ОМП) в линиях электропередач является актуальной со времени начала передачи электроэнергии в широких масштабах [1-2]. Практической целью решения этих задач является достижение наименьшей погрешности

ОМП наиболее бюджетными методами. Развитие электроники, средств радиосвязи и вычислительной техники позволили вплотную подойти к точностям измеряемым метрами [3] с использованием волновых алгоритмов. Бюджетность получаемых решений [4] позволяет внедрять такие системы в сетях среднего класса напряжений с древовидной структурой при ОМП с малыми токами при однофазных замыканиях на землю. Регистрация сигналов переходного процесса (СПП) с высокой, до 10 МГц, частотой дискретизации поставляет большой объем цифровой информации анализ которой требует применения интеллектуальных алгоритмов. Комплексы с волновым алгоритмом ОМП регистрируют широкий класс явлений электроразрядной активности в линии начиная от частичных разрядов и кончая плановыми коммутациями и короткими замыканиями как внутри контролируемой зоны, так и на соседних участках сети. Определение существенных признаков СПП, к которым относится время начала СПП, форма амплитуды, период свободных колебаний и других параметров невозможно без использования классических численных алгоритмов. Далее в работе рассматривается применение одного из видов модального анализа, основанного на построении огибающей СПП [5,6], примененного к осциллограммам, полученным в [4].

Метод эмпирической модовой декомпозиции предполагает последовательное разложение непрерывных или дискретных сигналов исходных данных на модовые составляющие. Алгоритм подразумевает поиск локальных экстремумов и построение огибающей сигнала сверху и снизу [5].

СПП формируется на входе датчика из сигналов тока и напряжения, которые содержат как сигнал промышленной частоты, так и СПП [4]. Входные высокочастотные аналоговые цепи частично ослабляют сигналы промышленной частоты, имеющие значительную амплитуду. СПП является значительно более высокочастотным и проходит с меньшим ослаблением на вход аналого-цифрового преобразователя. Способ обнаружения СПП основан на дифференциальном алгоритме [7] и он хорошо функционирует даже в присутствии сигналов промышленной частоты. Зафиксированная осциллограмма СПП, с интервалом дискретизации 1,085 мкс, передается на выделенный сервер, где происходит уточнение времени начала СПП и совместно обрабатываются синхронно зафиксированные СПП в разных точках сети с целью ОМП.

Рассмотрим основные этапы реализации алгоритма модального анализа и их иллюстрацией блок схемой (рис.1) и программой на языке Python (табл.1). Алгоритм основан на построении ступенчатой огибающей СПП сверху и снизу, проходящей через локальные максимумы и минимумы. Ширина и высота ступенек огибающей содержат информацию о начале, максимальной амплитуде, количестве глобальных максимумов, длительности, периоде свободных колебаний СПП.

Алгоритм состоит из следующих этапов.

1. Определяем среднее значение от мгновенных амплитуд исходной осциллограммы на доаварийном интервале 0-1500мкс. Вычитаем полученное среднее значение из всех мгновенных амплитуд исходной осциллограммы.

2. Локальные максимумы делятся на два типа – с максимумом в одной точке и с плоской вершиной максимума, состоящей из нескольких точек. Для построения верхней огибающей задаем пять условий определения локальных максимумов, для построения нижней огибающей задаем условия определения локальных минимумов. Огибающие находятся лишь в своей полуплоскости.

3. Определение величины порога происходит на доаварийном интервале первых 1500 отсчетов осциллограммы. Он необходим для определения начала СПП. необходимо воспользуемся величиной размаха. Размах равен разности положительной огибающей и отрицательной огибающей. Определяем порог как минимальное значение размаха на интервале первых 1500 отсчетов осциллограммы.

4. Находим индекс максимального размаха. Для нахождения начала СПП двигаемся от

максимума размаха справа-налево и находим первую точку СПП, превысившую порог. Проверяем высоту и ширину ступеньки размаха, на которой находится анализируемая точка. Если высота больше 1, а ширина ступеньки размаха больше 12(примерно), то определяем текущую точку как время начала СПП. Если ширина ступеньки размаха меньше 12, то двигаемся влево до пересечения со следующим локальным максимумом СПП.

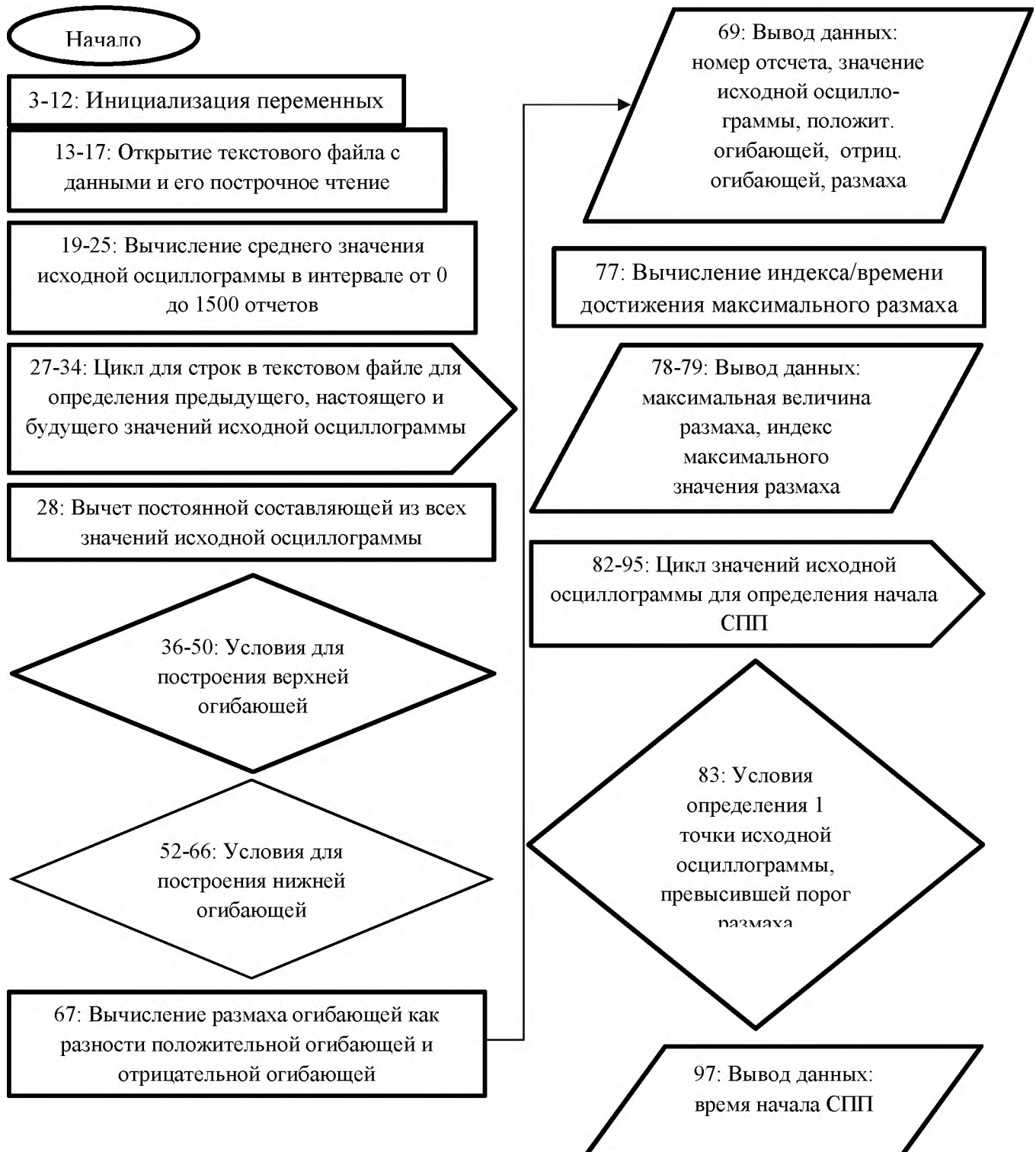


Рис.1 Блок схема программы

Запишем отдельно условия для построения верхней огибающей.

1. Если настоящая точка больше предыдущей и больше будущей точки, значение огибающей равно значению исходной осциллограммы
2. Если настоящая точка больше предыдущей и меньше будущей точки, то значение огибающей не меняется
3. Если настоящая точка равна предыдущей и больше будущей точки, то значение огибающей равно значению исходной осциллограммы
4. Если настоящая точка равна предыдущей и меньше будущей точки, а градиент больше 0, значение огибающей не меняется

На рис.2 изображена огибающая СПП, которая является исходной информацией к вычислению значений существенных признаков.

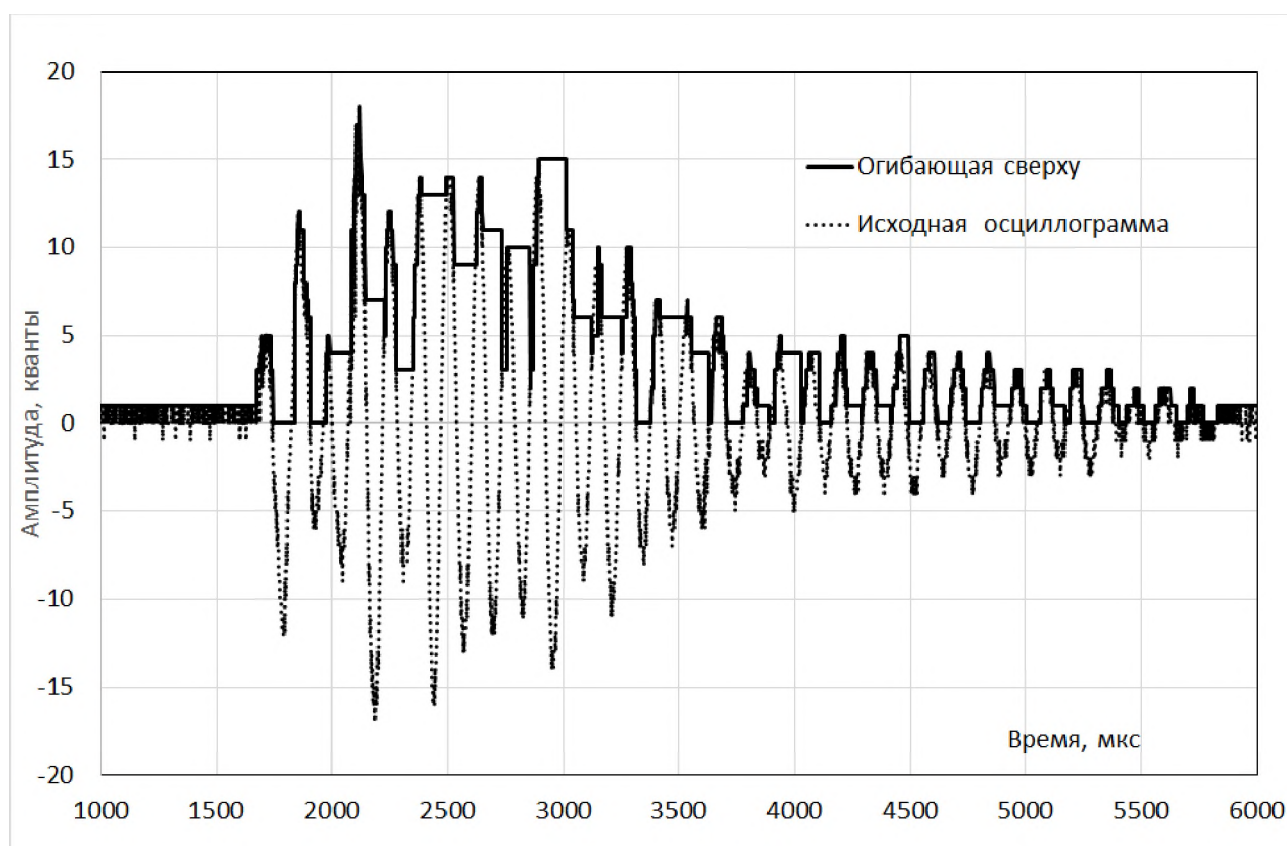


Рис.2 Пример построения огибающей СПП события №29288

Табл.1 Код программы в Python

	<b>import</b> itertools
	<b>from</b> itertools <b>import</b> groupby
	ypred,y,ybud = 0,0,0
	z1,z2 = 0,0
	d1,d2 = 0,0
	i = 0
	p = [] #список значений осциллограммы
	b = [] #список значений осциллограмма для вычета постоянной составляющей
	v = [] #список значений огибающей сверху

n = []	#список значений огибающей снизу
r = 0	#последовательные значения размаха
s = []	#список размахов
m = []	
with open("1.txt") as file:	
lines = file.read().split()	
for line in lines:	
w = line	
w = int(w)	
#вычисление среднего значения y	
b.append(w)	
b = b [0:1500]	
def Average(b):	
return sum(b)/len(b)	
average = Average(b)	
q = average	
q = int(q)	
#вычет постоянной составляющей	
for line in lines:	
line = int(line)-q	
ypred = y	
y = ybud	
ybud = line	
ypred = int(ypred)	
y = int(y)	
ybud = int(ybud)	
#верхняя огибающая	
if ((y > ypred) and (y > ybud) and (y >= 0)):	
z1 = y	
d1 = ybud-y	
elif ((y > ypred) and (y < ybud)):	
z1 = z1	
d1 = ybud-y	
elif ((y == ypred) and (y < ybud)):	
z1 = z1	
d1 = ybud-y	
elif ((y == ypred) and (y > ybud) and (y >= 0) and (d1 > 0)):	
z1 = y	
d1 = ybud-y	
elif ((y == ypred) and (y < ybud) and (y >= 0) and (d1 > 0)):	
z1 = z1	
d1 = yb-y	
#нижняя огибающая	
if ((y < ypred) and (y < ybud) and (y <= 0)):	
z2 = y	
d2 = ybud-y	
elif ((y < ypred) and (y > ybud)):	
z2 = z2	
d2 = ybud-y	

	<b>elif ((y == ypred) and (y &gt; ybud)):</b>
	z2 = z2
	d2 = ybud - y
	<b>elif ((y == ypred) and (y &lt; ybud) and (y &lt;= 0) and (d2 &lt; 0)):</b>
	z2 = y
	d2 = ybud - y
	<b>elif ((y == ypred) and (y &gt; ybud) and (y &lt;= 0) and (d2 &lt; 0)):</b>
	z2 = z2
	d2 = ybud - y
	r = z1 - z2 #размах огибающей
	i = i + 1
	#print(i,y,z1,z2,r)
	p.append(y) #внесение в список значений осциллограммы
	v.append(z1) #внесение в список значений огибающей сверху
	n.append(z2) #внесение в список значений огибающей снизу
	s.append(r) #внесение в список значений размаха
	t = s[0:1500] #копирует список s и срезает его от 0 до 1500
	m = list(filter( <b>lambda</b> a: a != 0, t)) #удаление 0-х значений из списка размахов для определения порога
	o = s.index(max(s)) + 1 #индекс максимальной амплитуды размаха
	<b>print</b> ("максимальная величина размаха;", max(s))
	<b>print</b> ("время достижения максимального размаха;", o)
	q = list(reversed(p[o:])) #инвертированная во времени исходная осциллограмма
	j = list(reversed(s[o:])) #инвертированный во времени размах
	<b>for i in range(len(q)):</b> #цикл инвертированной исходной осциллограммы
	<b>if ((j[i] &lt;= 1) and (q[i] &lt;= min(m)) and (q[i] &gt;= 0)):</b> #условия для определения 1 точки исходной осциллограммы, превысившей порог размаха
	k = j[i] # значение размаха, которое соответствует точке исходной осциллограммы превысившей порог или равной ей
	f = j[i:] #срезаем список размахов от предполагаемого значения исходной осциллограммы
	<b>for b in f:</b> #счетчик считает, пока значение не изменится
	<b>if b != k:</b> #условие для подсчета ширины ступеньки размаха, выполняется пока значение в списке размахов повторяются
	g = f.index(b) + 1 #индекс размаха, значение которого отличается от найденного
	sh = f[g:] #список значений ширины ступеньки размаха, на которой находится предполагаемая точка исходной осциллограммы
	l = len(sh) #ширина ступеньки
	<b>if l &gt;= 12:</b>
	<b>break</b>
	<b>else:</b>
	<b>continue</b>
	<b>break</b>
	u = int(o) - int(i) #время начала СПП
	<b>print</b> ("время начала спп;", u)

Встроенные функции Python, использованные в алгоритме:

1. List – создание списка, обработка итерируемых объектов [<https://pythonworld.ru/tipy-dannyx-v-python/spiski-list-funkcii-i-metody-spiskov.html>]
2. Filter – фильтрация последовательностей [<https://younglinux.info/python/feature/filter>]
3. Lambda - это небольшая анонимная функция. Она может принимать любое количество аргументов, но в то же время иметь только одно выражение. [<https://pythonru.com/uroki/lambda-funcija-uroki-po-python-dlja-nachinajushhih>]
4. Max - возвращает наибольший из своих аргументов, то есть значение ближе к положительной бесконечности. [<https://andreyex.ru/yazyk-programmirovaniya-python/uchebnik-po-python-3/python-3-chislovaya-funkciya-max/>]
5. Reversed - возвращает обратный итератор по указанной последовательности. [<https://pythonz.net/references/named/reversed/>]
6. Range - позволяет генерировать ряд чисел в рамках заданного диапазона. [<https://python-scripts.com/range>]
7. Len - возвращает длину строки. [<https://andreyex.ru/yazyk-programmirovaniya-python/uchebnik-po-python-3/python-3-strokovaya-funkciya-len/>]

### Заключение

Описан и продемонстрирован на численном примере начальный этап модального анализа осциллограмм сигналов переходного процесса. Целью использования модального анализа является вычисление существенных признаков – уровень шумов, времени начала, максимальной амплитуде, количестве глобальных максимумов, длительности, периоде свободных колебаний сигнала переходного процесса. Показана процедура определения порога для вычисления времени начала сигнала переходного процесса – важнейшего признака в задаче волнового определения места повреждения.

### Литература

1. L. V. Bewley, “Traveling Waves on Transmission Systems”, General Electric Company, Pittsfield, MA, 193
2. Cole D., Diamond M., Kulshrestha A. Fault Location & System Restoration//PACworld. 2012. №9.
3. Hou M., Wang D. Travelling wave fault location principle for hybrid multi-terminal LCC-VSC-HVDC transmission line based on R-ECT // International Journal of Electrical Power & Energy Systems/ 2020. V.117, №5.
4. Хузяшев Р.Г., Кузьмин И.Л., Васильев В.Д., Тукаев С.М. Практическая реализация волнового метода определения места повреждения в разветвленных распределительных электрических сетях 6(10) кВ // Электроэнергия. Передача и распределение. 2019. Т. 2, № 53. С. 98-107.
5. Коваленко П.Ю. Методы анализа... на базе векторных измерений: дисс.к.т.н. – 2017. Екатеринбург, 161 с
6. Gong Y., Guzmán A. Synchrophasor-Based Online Modal Analysis to Mitigate Power System Interarea Oscillation // SEL Journal of Reliable Power. 2011. V.2, № 2.
7. Простейшие алгоритмы обнаружения сигналов переходного процесса напряжения в линиях электропередач / С.И.Новиков. И.Л. Кузьмин, Р.Г.Хузяшев. // Проблемы энергетики, 2017, Т. 19, № 7-8. С. 114-125.