



## Алгоритм выделения следа искусственного сигнала вистлера в спектрограмме с помощью интегрированной среды разработки приложений PyCharm

*Л. С. Марченко<sup>1,2\*</sup>*

<sup>1</sup> Институт космофизических исследований и распространения радиоволн ДВО РАН, 684034, с. Паратунка, ул. Мирная, д. 7, Россия

<sup>2</sup> Научно-исследовательский геотехнологический центр ДВО РАН, 683002, Россия, г. Петропавловск-Камчатский, ул. Северо-Восточное шоссе, 30, Россия

**Аннотация.** В работе предложен алгоритм выделения следа искусственного сигнала свистящего атмосферика (вистлера) в спектрограмме, реализованный на языке Python в интегрированной среде разработки PyCharm 2024.1. Алгоритм позволяет с помощью установки некоторого порогового значения (фильтра) выделить след вистлера. Фильтр учитывает интенсивность сигнала в спектре, стандартное отклонение значений от среднего, а также некоторый множитель, который позволяет исключить шум и выделить только более значимые пики в сигнале. В алгоритме с помощью маски на основе фильтра удается получить массив частот для следа искусственного вистлера. Компьютерная программа позволяет сохранять полученный массив в текстовый файл, который можно использовать для дальнейшего анализа в различных табличных процессорах, а также строить графики следа вистлера для визуального исследования. В статье была произведена проверка адекватности алгоритма на примере расчета коэффициента дисперсии. Показано, что алгоритм дает хорошие результаты.

*Ключевые слова:* искусственный вистлер, спектрограмма, след, фильтр, маска, Python, PyCharm

Получение: 25.10.2024; Исправление: 13.11.2024; Принятие: 21.11.2024; Публикация онлайн: 28.11.2024

**Для цитирования.** Марченко Л. С. Алгоритм выделения следа искусственного сигнала вистлера в спектрограмме с помощью интегрированной среды разработки приложений PyCharm // *Вестник КРАУНЦ. Физ.-мат. науки.* 2024. Т. 49. № 4. С. 99-111. EDN: YLXDQQ. <https://doi.org/10.26117/2079-6641-2024-49-4-99-111>.

**Финансирование.** Работа выполнена за счет Государственного задания ИКИР ДВО РАН (рег. № НИОКТР 124012300245-2).

**Конкурирующие интересы.** Конфликтов интересов в отношении авторства и публикации нет.

**Авторский вклад и ответственность.** Автор несет ответственность за предоставление окончательной версии статьи в печать.

\*Корреспонденция:  E-mail: [marchenko@ikir.ru](mailto:marchenko@ikir.ru)

Контент публикуется на условиях Creative Commons Attribution 4.0 International License

© Марченко Л. С., 2024

© ИКИР ДВО РАН, 2024 (оригинал-макет, дизайн, составление)





## Algorithm for Extracting an Artificial Whistler Signal in a Spectrogram Using the PyCharm Integrated Application Development Environment

*L. S. Marchenko*<sup>1,2\*</sup>

<sup>1</sup> Institute of Cosmophysical Research and Radio Wave Propagation FEB RAS, 684034, Kamchatka, Elizovsky District, Paratunka, 7 Mirnaya street, Russia

<sup>2</sup> Scientific Research Geotechnological Centre FEB RAS, 683002, Petropavlovsk-Kamchatsky, 30 Severo-Vostochnoye highway, Russia

**Abstract.** The paper proposes an algorithm for identifying the trace of an artificial whistling atmospheric signal (whistle) in a spectrogram, implemented in Python in the PyCharm 2024.1 integrated development environment. The algorithm allows you to identify the whistler trace by setting a certain threshold value (filter). The filter takes into account the signal intensity in the spectrum, the standard deviation of values from the mean, and a certain multiplier that allows you to exclude noise and identify only the most significant peaks in the signal. In the algorithm, using a mask based on the filter, it is possible to obtain an array of frequencies for the trace of an artificial whistler. The computer program allows you to save the resulting array in a text file, which can be used for further analysis in various spreadsheet processors, as well as build whistler trace graphs for visual research. The article tested the adequacy of the algorithm using the example of calculating the dispersion coefficient. It was shown that the algorithm gives good results.

*Key words:* artificial whistler, spectrogram, trace, filter, mask, Python, PyCharm


Received: 25.10.2024; Revised: 13.11.2024; Accepted: 21.11.2024; First online: 28.11.2024

**For citation.** Marchenko L. S. Algorithm for extracting an artificial whistler signal in a spectrogram using the PyCharm integrated application development environment. *Vestnik KRAUNC. Fiz.-mat. nauki.* 2024, 49: 4, 99-111. EDN: YLXDQQ. <https://doi.org/10.26117/2079-6641-2024-49-4-99-111>.

**Funding.** The work was supported by IKIR FEB RAS State Task (Reg. No. NIOKTR 124012300245-2).

**Competing interests.** The authors declare that there are no conflicts of interest regarding authorship and publication.

**Contribution and Responsibility.** Author is solely responsible for providing the final version of the article in print.

\*Correspondence:  E-mail: [marchenko@ikir.ru](mailto:marchenko@ikir.ru)

The content is published under the terms of the Creative Commons Attribution 4.0 International License

© Marchenko L. S., 2024

© Institute of Cosmophysical Research and Radio Wave Propagation, 2024 (original layout, design, compilation)



## Введение

В последние десятилетия наблюдение и анализ электромагнитных волн, связанных с природными явлениями, стали важной областью исследований в физике плазмы и геофизике [1, 2]. Одним из наиболее интересных аспектов таких исследований является изучение вистлеров - низкочастотных электромагнитных волн, возникающих в результате грозových разрядов с последующим взаимодействием с магнитосферой Земли [3–11, 11]. Эти волны играют ключевую роль в динамике плазмы и могут оказывать значительное влияние на радиосигналы и системы связи. Свое название вистлеры (свисты или свистящие атмосферерики) получили из-за характерного звука, который они производят при преобразовании в звуковые волны.

Теоретические основы исследования вистлеров более подробно и системно были изложены в фундаментальных работах [12–14]. Дальнейшее исследование вистлеров получило широкое развитие в работах различных ученых, как зарубежных [3–8], так и российских [9–11, 11].

Отметим, что наиболее важными являются задачи исследования источников вистлеров – грозových разрядов [9–11], а также распознавания вистлеров в реальном времени [11]. К сожалению задача распознавания вистлеров в реальном времени до конца не была решена методами машинного обучения в силу сложности идентификации вистлеров по спектрограммам.

В работе Шагимуратова И.А. [15] подробно изложена теория исследования динамических спектров вистлеров, в которой указывается на важность определения следа вистлера, что в свою очередь дает возможность применять многоточечные методы обработки.

В настоящей работе для искусственно смоделированного вистлера [16] был разработан на языке Python в среде PyCharm 2024.1 алгоритм выделения его следа из спектрограммы. Данный алгоритм далее планируется применить для исследования характеристик искусственного вистлера согласно работе [15], а потом его адаптировать для исследования реальных сигналов вистлера, полученных с помощью ОНЧ-пеленгатора ИКИР ДВО РАН

## Функциональная модель алгоритма

Интегрированная среда разработки (IDE) PyCharm 2024.1 для языка программирования Python имеет ряд преимуществ по сравнению с другими версиями PyCharm [17]: улучшенная поддержка AI: встроенные инструменты на базе искусственного интеллекта помогают в написании кода, предоставляя автозаполнение, рекомендации и исправления; оптимизированный интерфейс: новый интерфейс стал более интуитивно понятным и удобным, что упрощает навигацию и работу с проектами; расширенные возможности рефакторинга: новые инструменты для рефакторинга позволяют быстро и безопасно изменять структуру кода без риска появления ошибок: поддержка новых

технологий: улучшена поддержка последних версий Python и популярных библиотек, таких как Django, Flask и других; инструменты для тестирования: улучшенные функции для написания и запуска тестов, включая поддержку новых фреймворков и расширенные возможности анализа результатов; оптимизация производительности: повышенная скорость работы IDE, улучшенное использование ресурсов системы и уменьшение времени загрузки проектов; поддержка многопоточности: улучшенная работа с асинхронным кодом и многопоточностью, что делает разработку более эффективной.

На рис.1 приведена блок-схема алгоритма выделения следа искусственного вистлера в среде PyCharm 2024.1.



Рис. 1. Блок-схема алгоритма в среде PyCharm.

Figure 1. Block diagram of the algorithm in the PyCharm environment.

Из блок-схемы на рис.1. мы можем выделить следующие этапы алгоритма:

1. Построение искусственного вистлера. Этот этап был подробно рассмотрен в работе автора [16].

2. Фильтрация искусственного сигнала вистлера. Установка порога для выделения следа искусственного вистлера.

3. Проверка работы алгоритма на примере вычисления коэффициента дисперсии.

## Построение искусственного вистлера

Мы здесь приведем лишь фрагмент кода на языке Python, более полный код смотрите в работе [16]. Значения параметров искусственного сигнала возьмем также из работы [16].

```
# Подключение необходимых библиотек Python
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
from tkinter import Tk
from tkinter.filedialog import asksaveasfilename
import pandas as pd
# Параметры модели, [17]
Dis = 100 # Коэффициент дисперсии
f_min = 1000 # Минимальная частота, Гц
f_max = 8000 # Максимальная частота, Гц
t_max = 3 # Максимальное время, с
fs = 50000 # Частота дискретизации, Гц
```

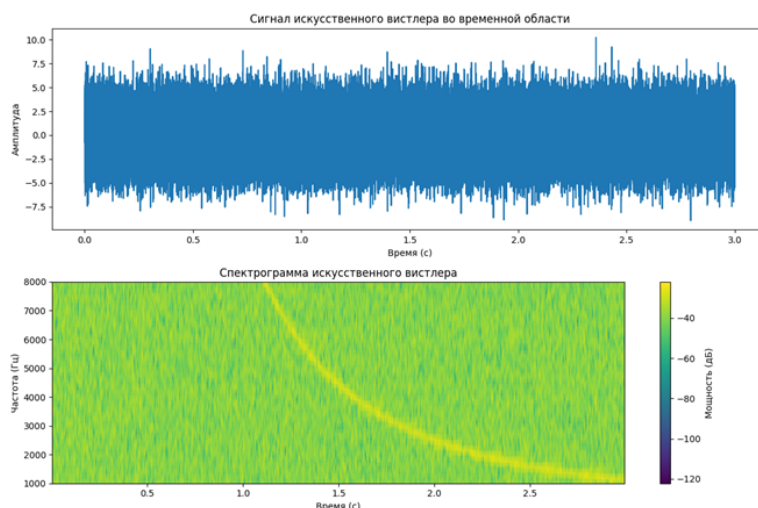


Рис. 2. Искусственный вистлер при  $D = 100 \text{ с}^{1/2}$  [16].

Figure 2. Artificial whistler at  $D = 100 \text{ с}^{1/2}$  [16].

## Описание фильтра для выделения следа искусственного вистлера

В этом пункте мы опишем фильтрацию для выделения следа искусственного сигнала вистлера на рис. 2. Для этой цели установка порога для выделения следа на языке Python имеет вид:

$$\text{threshold} = \text{np.mean}(S_{xx}) + k * \text{np.std}(S_{xx}).$$

Здесь мы устанавливаем пороговое значение (`threshold`), которое будет использоваться для выделения значимых участков в спектрограмме; `np.mean(Sxx)` вычисляет среднее значение всех элементов в массиве `Sxx`, который представляет собой спектрограмму и показывает, какова типичная интенсивность сигнала в спектре; `np.std(Sxx)` вычисляет стандартное отклонение элементов в `Sxx`, оно указывает, насколько значения в массиве разбросаны относительно среднего; `k * np.std(Sxx)` - умножение стандартного отклонения на `k` позволяет установить порог выше среднего значения, учитывая разброс данных, что помогает исключить шум и выделить только более значимые пики.

$$\text{Создание маски: } \text{mask} = S_{xx} > \text{threshold}.$$

Здесь мы создаем логическую маску (`mask`), которая будет содержать `True` для всех точек в `Sxx`, где интенсивность превышает установленный порог (`threshold`), и `False` в противном случае.

Это позволяет нам выделить области в спектрограмме, где интенсивность сигнала значительно выше среднего уровня, что может указывать на наличие интересующих нас вистлеров.

Заметим, что мы выбрали простой пороговый метод. Однако можно использовать более сложный метод пороговой обработки, например, известны такие методы: адаптивный пороговый метод, который зависит от локальных характеристик сигнала; метод Оцу [18] — метод используется для нахождения оптимального порога, который минимизирует внутриклассовую дисперсию; краевые методы — алгоритмы для нахождения краев сигнала, такие как метод Собеля или Кэнни [19, 20].

Код на языке Python, характеризующий работу фильтра имеет вид:

```
# 1. Установка порога для выделения следа
threshold = np.mean(Sxx) + 10 * np.std(Sxx) # Простой порог
# 2. Создаем маску
mask = Sxx > threshold
# 3. Получаем индексы, где маска равна True
indices = np.where(mask)
# 4. Сохранение значений частоты и времени в текстовый файл
Tk().withdraw() # Скрыть основное окно
file_path = asksaveasfilename(defaulttextextension=".txt",
filetypes=[("Text files", "*.txt"), ("All files", "*.*")],
```

```
title="Сохранить как")
if file_path: # Проверка, что путь был выбран
# Создаем DataFrame с частотой и временем
data = {
    'Частота (Гц)': f[indices[0]],
    'Время (с)': t_spec[indices[1]]
}
results_df = pd.DataFrame(data)

# Сохраняем в текстовый файл
results_df.to_csv(file_path, sep='\t', index=False, header=True)
print(f"Файл сохранен: {file_path}")
else:
print("Сохранение отменено.")
# 5. Построение графика частот свистящего атмосферика
plt.figure(figsize=(12, 6))
plt.scatter(t_spec[indices[1]], f[indices[0]], s=10, color='blue',
alpha=0.5) # Используем scatter для отображения точек
plt.title('Выделенные следа свистящего атмосферика')
plt.xlabel('Время (с)')
plt.ylabel('Частота (Гц)')
plt.ylim(f_min, f_max)
plt.grid()
plt.tight_layout()
plt.show()
```

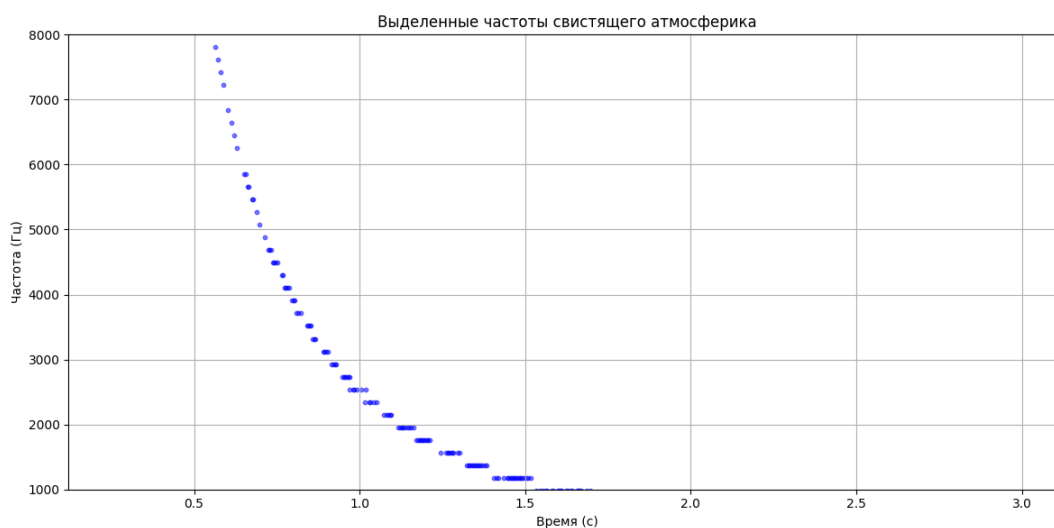


Рис. 3. График следа искусственного вистлера из рис.2.  
Figure 3. Graph of the trace of the artificial whistler from Fig. 2.

## Вычисление коэффициента дисперсии

След искусственного вистлера на рис. 3 позволяет исследовать его характеристики. Покажем, что след искусственного вистлера найден корректно. Для этого мы найдем коэффициент дисперсии по углу наклона прямой построенной по точкам  $(1/\sqrt{f}, t)$ . Методика расчета коэффициента дисперсии подробно описана в книге [12].

Код на языке Python для расчета коэффициента дисперсии.

```
# Создаем дополнительную таблицу с временем t и 1/f^(1/2)
additional_data = {
    'Время (с)': t_spec[indices[1]][valid_indices], # t
    '1/f^(1/2)': 1 / np.sqrt(f[indices[0]][valid_indices]) # 1/f^(1/2)
}
additional_df = pd.DataFrame(additional_data)
# Сохранение дополнительной таблицы в отдельный файл
additional_file_path = asksaveasfilename(defaultextension=".txt",
filetypes=[("Text files", "*.txt"), ("All files", "*.*")],
title="Сохранить дополнительную таблицу как")
if additional_file_path: # Проверка, что путь был выбран
additional_df.to_csv(additional_file_path, sep='\t', index=False, header=True)
print(f"Дополнительная таблица сохранена: {additional_file_path}")
# Построение графика для дополнительной таблицы
X = additional_df['Время (с)'].values.reshape(-1, 1)
y = additional_df['1/f^(1/2)'].values
# Линейная регрессия
model = LinearRegression()
model.fit(X, y)
# Получаем тангенс угла наклона и свободный член
tangent = model.coef_[0]
intercept = model.intercept_
# Уравнение линейной регрессии
equation = f'y = {tangent:.4f} * t + {intercept:.4f}'
# Вычисление котангенса угла наклона
cotangent = 1 / tangent
# Построение графика
plt.figure(figsize=(10, 6))
plt.scatter(X, y, color='blue', label='Данные')
plt.plot(X, model.predict(X), color='red', label='Линейная регрессия')
plt.title('График зависимости времени от 1/f^(1/2)')
plt.xlabel('Время (с)')
plt.ylabel('1/f^(1/2)')
plt.legend()
plt.grid()
```

```
# Вывод уравнения на график
plt.text(0.05, 0.95, equation, transform=plt.gca().transAxes,
        fontsize=12, verticalalignment='top', color='black')
plt.tight_layout()
plt.show()
```

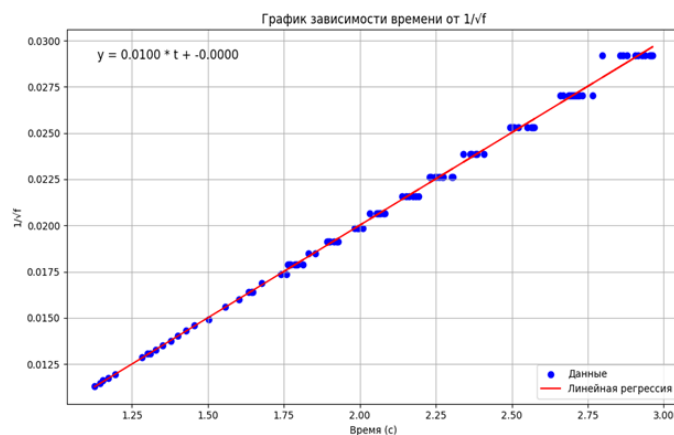


Рис. 4. Данные, полученные по следу искусственного вистлера (точки) при  $D = 100 \text{ c}^{1/2}$ , прямая линия – уравнение линейной регрессии.

Figure 4. Data obtained from the trail of an artificial whistler (points) at  $D = 100 \text{ c}^{1/2}$ , the straight line is the linear regression equation.

Из рис.4 видно, что угловой коэффициент или тангенс угла наклона равен 0.01, тогда расчетный коэффициент дисперсии  $D_R = 1/0.01 = 100 \text{ c}^{1/2}$ . Отсюда следует, что расчетный коэффициент  $D_R$  совпадает с теоретическим коэффициентом  $D$ .

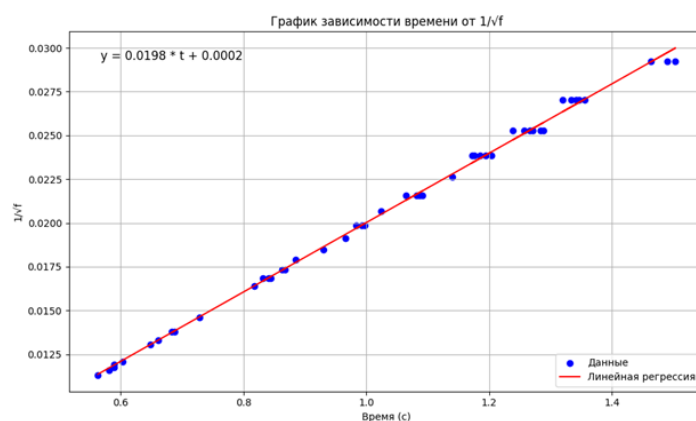


Рис. 5. Данные полученные по следу искусственного вистлера (точки) при  $D = 50 \text{ c}^{1/2}$ , прямая линия – уравнение линейной регрессии.

Figure 5. Data obtained from the trail of an artificial whistler (points) at  $D = 50 \text{ c}^{1/2}$ , the straight line is the linear regression equation.

Из рис.5 видно, что тангенс угла наклона равен 0.0198, тогда расчетный коэффициент дисперсии  $D_R = 1/0.0198 = 50.45 \text{ с}^{1/2}$ . Отсюда следует, что расчетный коэффициент  $D_R$  практически совпадает с теоретическим коэффициентом  $D = 50 \text{ с}^{1/2}$ .

Отметим, что коэффициент дисперсии можно рассчитать согласно математической модели, предложенной в статье [16]:  $D_M = T\sqrt{f_T}$ , где  $T$  — время существования вистлера, в нашем случае это последнее время в отчете текстового файла, которое генерируется нашей программой,  $f_T$  — частота соответствующая времени  $T$ . Тогда мы получаем следующие значения:

$$D_M = 0.70144\sqrt{5078.124999999999} = 49,985 \text{ с}^{1/2} \text{ при теоретической } D = 50 \text{ с}^{1/2},$$

$$D_M = 2.17536\sqrt{2148.4374999999995} = 100.83 \text{ с}^{1/2} \text{ при теоретической } D = 100 \text{ с}^{1/2}.$$

Мы видим, что оценки, полученные по математической модели, также дают приемлемый результат

## Заключение

В работе предложен алгоритм выделения следа искусственного вистлера из спектрограммы на языке Python в среде PyCharm 2024.1. С помощью фильтрации определяются необходимые частоты и времена, которые им соответствуют и характеризуют след искусственного вистлера. Далее, чтобы проверить адекватность алгоритма, был произведен расчет коэффициента дисперсии по уравнению линейной регрессии. Показано, что алгоритм является адекватным и дает приемлемые результаты.

Отметим, что следующим этапом развития работы является исследование реальных сигналов вистлеров, полученные с помощью ОНЧ-пеленгатора в ИКИР ДВО РАН. Следует подчеркнуть, что природные сигналы являются более сложными, так как помимо вистлеров регистрируются атмосферники. Поэтому необходимо будет корректировать алгоритм на этапе фильтрации с возможным применением машинного обучения и других методов обработки изображений.

Автор выражает благодарность своему научному руководителю д.ф.-м.н. Паровику Р.И. за ценные советы, которые послужили улучшению статьи.


## Список литературы

1. Artekha S. N., Belyan A. V. On the role of electromagnetic phenomena in some atmospheric processes // *Nonlinear Processes in Geophysics*, 2013. vol. 20, no. 3, pp. 293-304 DOI:10.5194/npg-20-293-2013.
2. Mahmood N., Edminister J.A. *Schaum's Outline of Electromagnetics*. New York: McGraw Hill, 2019.
3. Lichtenberger J., Ferencz C., Bodnár L. et al. Automatic whistler detector and analyzer system: Automatic whistler detector // *Geophys. Res.*, 2008. vol. 113.
4. Koroncay D., Lichtenberger J., Clilverd M. A. et al. The source regions of whistlers // *Journal of Geophysical Research: Space-Physics*, 2019. vol. 124, pp. 5082–5096.
5. Li W., Shen X.-C., Menietti J. D. et al. Global distribution of whistler mode waves in Jovian inner magnetosphere // *Geophysical Research Letters*, 2020. vol. 47, no. 15. DOI: 10.1029/ 2020GL088198.

6. Morris P.J., Bohdan A., Weidl M. S. et al. Pre-acceleration in the electron foreshock. II. oblique whistler waves // *The Astrophysical Journal*, 2023. vol. 944, no. 1 DOI: 10.3847/1538-4357/acaec8.
7. Sonwalkar V. S., Reddy A. Specularly reflected whistler: A low-latitude channel to couple lightning energy to the magnetosphere // *Science Advances*, 2024. vol. 10., no. 33 DOI: 10.1126/sciadv.ado2657.
8. Xiang T., Liu M., He S., Wang X., Zhou C. Automatic segmentation model and parameter extraction algorithm for lightning whistlers // *Radio Science*, 2024. vol. 59, no. 11 DOI: 10.1029/2024RS007984.
9. Чернева Н. В., Водинчар Г. М., Сивоконь В. П. и др. Корреляционный анализ потоков свистящих атмосфериков и грозовых разрядов // *Вестник КРАУНЦ. Физико-математические науки*, 2013. Т. 7, № 2, С. 59–67 DOI: 10.18454/2079-6641-2013-7-2-59-67.
10. Сивоконь В. П., Богданов В. В., Дружин Г. И. и др. Модуляция вистлеров // *Геомагнетизм и аэронаука*, 2014. Т. 54, № 6, С. 851–851 DOI: 10.7868/S0016794014060182.
11. Малыш Е. А. Алгоритм для автоматического распознавания свистящих атмосфериков в режиме реального времени // *Вестник КРАУНЦ. Физико-математические науки*, 2015. № 2(11), С. 82-87 DOI: 10.18454/2079-6641-2015-11-2-82-87.
12. Малкин Е. И., Казаков Е. А., Санников Д. В. и др. Статистическая связь между вистлерами и спрайтами по данным AWDANET и WWLLN // *Вестник КРАУНЦ. Физико-математические науки*, 2022. Т. 41, № 4, С. 178-190 DOI: 10.26117/2079-6641-2022-41-4-178-190.
13. Storey L. R. O. An investigation of whistling atmospherics // *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 1953. vol. 246, no. 908, pp. 113-141.
14. Гершман Б. Н., Коробков Ю. С. К теории распространения свистящих атмосфериков // *Известия вузов. Серия Радиофизика*, 1958. Т. 1, № 2, С. 51-58.
15. Гершман Б. Н., Угаров В. А. Распространение и генерация низкочастотных электромагнитных волн в верхней атмосфере // *Успехи физических наук*, 1960. Т. 72, № 2, С. 235-271.
16. Шагимуратов И. И. *Вариации электронной концентрации в плазмосфере по данным свистящих атмосфериков*, специальность 01.04.12. М., 1985. 189 с.
17. Марченко Л. С. Паровик Р. И. Моделирование искусственных вистлеров в среде PyCharm // *Известия Кабардино-Балкарского научного центра РАН*, 2024. Т. 26, № 5, С. 53-63 DOI: 10.35330/1991-6639-2024-26-5-53-63.
18. Van Horn, B. M. II, Nguyen Q. *Hands-On Application Development with PyCharm: Build Applications like a Pro with the Ultimate Python Development Tool*. Birmingham, UK: Packt Publishing Ltd., 2023.
19. Talab A. M. A. et al. Detection crack in image using Otsu method and multiple filtering in image processing techniques // *Optik*, 2016. vol. 127, no. 3, pp. 1030-1033.
20. Mathur N., Mathur S., Mathur D. A novel approach to improve sobel edge detector // *Procedia Computer Science*, 2016. vol. 93, pp. 431-438.
21. Yan X., Li Y. A method of lane edge detection based on Canny algorithm // *IEEE*, 2017. 2120-2124 pp.

### Информация об авторе



Марченко Людмила Сергеевна ✉ – ведущий специалист учёного секретариат института космофизических исследований и распространения радиоволн ДВО РАН, Паратунка, аспирант, Научно-исследовательский геотехнологический центр ДВО РАН, г. Петропавловск-Камчатский, Россия,  ORCID 0000-0003-3634-2443.


## References

- [1] Artekha S. N., Belyan A. V. On the role of electromagnetic phenomena in some atmospheric processes. *Nonlinear Processes in Geophysics*. 2013. vol. 20. no. 3. P. 293-304. DOI:10.5194/npg-20-293-2013.
- [2] Mahmood N., Edminister J.A. *Schaum's Outline of Electromagnetics*. 5th ed. New York: McGraw Hill. 2019.
- [3] Lichtenberger J., Ferencz C., Bodnár L. et al. Automatic whistler detector and analyzer system: Automatic whistler detector. *Geophys. Res.* 2008. vol. 113.
- [4] Koronczay D., Lichtenberger J., Clilverd M. A. et al. The source regions of whistlers. *Journal of Geophysical Research: Space-Physics*, 2019. Vol. 124, Pp. 5082–5096.
- [5] Li W., Shen X.-C., Menietti J. D. et al. Global distribution of whistler mode waves in Jovian inner magnetosphere. *Geophysical Research Letters*. 2020. Vol. 47. No. 15. DOI: 10.1029/2020GL088198.
- [6] Morris P.J., Bohdan A., Weidl M. S. et al. Pre-acceleration in the electron foreshock. II. oblique whistler waves. *The Astrophysical Journal*. 2023. vol. 944. No. 1. Id 13. DOI: 10.3847/1538-4357/acaec8.
- [7] Sonwalkar V. S., Reddy A. Specularly reflected whistler: A low-latitude channel to couple lightning energy to the magnetosphere. *Science Advances*. 2024. Vol. 10. No. 33. eado2657. DOI: 10.1126/sciadv.ado2657.
- [8] Xiang T., Liu M., He S., Wang X., Zhou C. Automatic segmentation model and parameter extraction algorithm for lightning whistlers. *Radio Science*. 2024. vol. 59. no. 11. e2024RS007984. DOI: 10.1029/2024RS007984.
- [9] Cherneva N.V., Vodinchar G.M., Sivokon V.P. et al. Correlation analysis of fluxes of whistling atmospheric and lightning discharges *Vestnik KRAUNC. Fiziko-Matematicheskie Nauki*. 2013. Vol. 7. No. 2. Pp. 59–67. DOI: 10.18454/2079-6641-2013-7-2-59-67. (In Russian)
- [10] Sivokon V.P., Bogdanov V.V., Druzhin G.I. et al. Whistler modulation. *Geomagnetizm i Aeronomiya*. 2014. vol. 54. no. 6. P. 851–851. DOI: 10.7868/S0016794014060182. (In Russian)
- [11] Malush E.A. Algorithm for automatic recognition of whistling atmospheric in real time. *Vestnik KRAUNC. Fiziko-Matematicheskie Nauki*. 2015. No. 2(11). P. 82–87. DOI: 10.18454/2079-6641-2015-11-2-82-87. (In Russian)
- [12] Malkin E.I., Kazakov E.A., Sannikov D.V. et al. Statistical relationship between whistlers and sprites according to AWDANET and WWLLN. *Vestnik KRAUNC. Fiziko-Matematicheskie Nauki*. 2022. Vol. 41. No. 4. P. 178–190. (In Russian)
- [13] Storey L. R. O. An investigation of whistling atmospheric. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*. 1953. vol. 246. no. 908. P. 113-141.
- [14] Gershman B.N., Korobkov Yu.S. On the theory of propagation of whistling atmospheric. *Izvestiya Vuzov. Radiofizika*. 1958. Vol. 1. No. 2. Pp. 51–58. (In Russian)
- [15] Gershman B.N., Ugarov V.A. Propagation and generation of low-frequency electromagnetic waves in the upper atmosphere. *Uspekhi Fizicheskikh Nauk*. 1960. Vol. 72. No. 2. P. 235–271. (In Russian)
- [16] Shagimuratov I. I. Variations of electron concentration in the plasmasphere according to whistling atmospheric data: specialty 01.04.12. Moscow: 1985. 189 p. (In Russian)
- [17] Marchenko L.S., Parovik R.I. Modeling artificial whistlers in Pycharm. *News of the Kabardino-Balkarian Scientific Center of RAS*. 2024. Vol. 26. No. 5. P. 53–63. DOI: 10.35330/1991-6639-2024-26-5-53-63. (In Russian)

- [18] Van Horn, B. M., II; Nguyen, Q. Hands-On Application Development with PyCharm: Build Applications like a Pro with the Ultimate Python Development Tool; Packt Publishing Ltd.: Birmingham, UK, 2023.
- [19] Talab A. M. A. et al. Detection crack in image using Otsu method and multiple filtering in image processing techniques. *Optik*. 2016. vol. 127. no. 3. P. 1030-1033.
- [20] Mathur N., Mathur S., Mathur D. A novel approach to improve sobel edge detector. *Procedia Computer Science*. 2016. vol. 93. P. 431-438.
- [21] Yan X., Li Y. A method of lane edge detection based on Canny algorithm. *IEEE*. 2017. P. 2120-2124.

### Information about the author



*Marchenko Ludmila Sergeevna* ✉ – leading specialist of the scientist Secretariat of the Institute of Cosmophysical Research and Radio Wave Propagation of the Far Eastern Branch of the Russian Academy of Sciences, Paratunka, postgraduate student, Research Geotechnological Center of the Far Eastern Branch of the Russian Academy of Sciences, Petropavlovsk-Kamchatsky, Russia,  ORCID 0000-0003-3634-2443.