УДК 004

РЕАЛИЗАЦИЯ ВИЗУАЛИЗАЦИИ ГРАФИКОВ В НАСТОЛЬНОМ ПРИЛОЖЕНИИ

Черепанов П.В. – студент группы Пим-241,
Буйная Е.В. – научный руководитель, доцент кафедры
прикладных информационных технологий, к.э.н.
Кузбасский государственный технический университет имени Т.Ф. Горбачева
Россия, г. Кемерово

Визуализация является ключевым компонентом анализа данных. Традиционно для выполнения вычислений и построения графиков применяются электронные таблицы. Однако данный метод не всегда является оптимальным. Рассмотрим следующий пример.

В распоряжении имеется набор данных, включающий результаты натурных измерений, выполненных с использованием прибора учета на различные даты. Требуется построить график, основанный на результатах расчетов, произведенных с применением этих данных. Представленные данные оформлены в соответствии со стандартом CSV. Ввиду применения специфических методологических подходов к расчету, использование существующих программных продуктов, разработанных для данной отрасли, не представляется возможным.

Выполним визуализацию данных в виде графика с использованием программного обеспечения для электронных таблиц, основываясь на результатах проведенных вычислений.

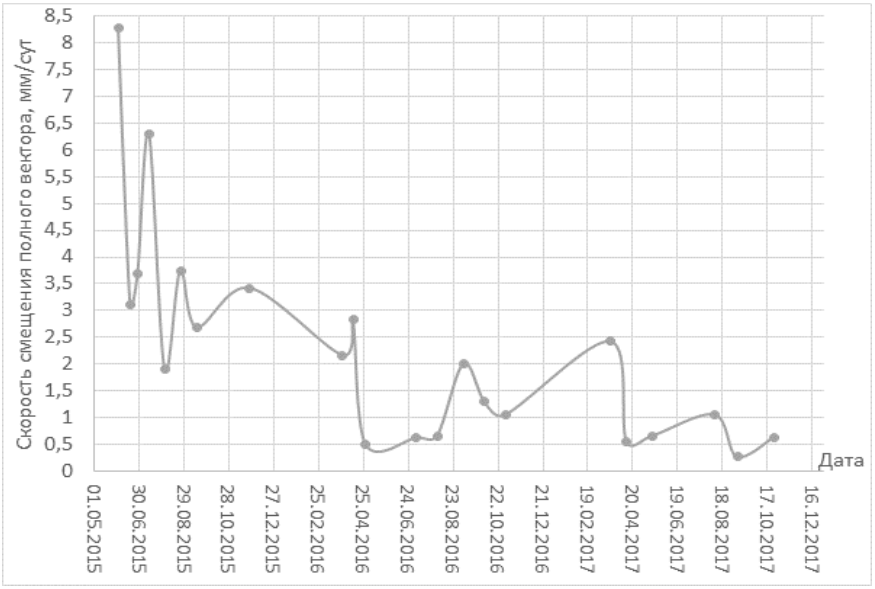

Рисунок 1 – График, построенный с помощью электронных таблиц

Класс Graph реализует визуализацию данных мониторинга смещений горных пород в графическом интерфейсе на основе PySide6 и Matplotlib. При инициализации создается холст для построения графиков размером 10×5

дюймов с разрешением 90 DPI, интегрируются панель навигации и кнопка вызова окна настроек. Основной метод drawMultiple выполняет построение графиков, принимая словарь данных, где ключи представляют метки кривых, а значения - кортежи с датами и значениями скорости смещения.

Даты в строковом формате "дд.мм.гггг" преобразуются в объекты datetime, после чего выполняется построение графиков с использованием циклической палитры из семи цветов и маркеров точек размером 7 пикселей. Особенностью визуализации является система цветового кодирования зон безопасности, отображаемая в виде полупрозрачных горизонтальных областей: зеленой ("Внимание"), желтой ("Опасно") и красной ("Стоп"). Параметры границ зон вручную вводит пользователь.

Настройки визуального представления включают форматирование оси времени с поворотом меток на 45 градусов, отображение дат в заданном формате, добавление координатной сетки, легенды и заголовков осей. Интегрированная панель навигации обеспечивает возможности масштабирования и панорамирования графика.

```
try:
    # Настройка зон безопасности
    settingsGraph = SettingsGraph()
    coalpit_id = self.ui.coalpitsFilterComboBoxForGraph.currentData()
    x_min, x_max = self.ax.get_xlim()
    text_x = x_max - (x_max - x_min) * 0.1
    # Зеленая зона - "Внимание"
    gy1, gy2 = settingsGraph.selectGY1(coalpit_id), settingsGraph.selectGY2(coalpit_id)
    self.ax.axhspan(gy1, gy2, facecolor='green', alpha=0.3)
    self.ax.text(text_x, (gy1 + gy2) / 2, "Внимание",ha='left', va='center',bbox=dict(facecolor='white', alpha=0.7, edgecolor='white', boxstyle='round,pad=0.5'))
    # Желтая зона - "Опасно"
    yy1, yy2 = settingsGraph.selectYY1(coalpit_id), settingsGraph.selectYY2(coalpit_id)
    self.ax.axhspan(yy1, yy2, facecolor='yellow', alpha=0.3)
    self.ax.text(text_x, (yy1 + yy2) / 2, "Опасно", ha='left', va='center',bbox=dict(facecolor='white', alpha=0.7, edgecolor='white', boxstyle='round,pad=0.5'))
    # Красная зона - "Стоп"
    ry1, ry2 = settingsGraph.selectRY1(coalpit_id), settingsGraph.selectRY2(coalpit_id)
    self.ax.axhspan(ry1, ry2, facecolor='red', alpha=0.3)
    self.ax.text(text_x, (ry1 + ry2) / 2, "Стоп",ha='left', va='center',bbox=dict(facecolor='white', alpha=0.7, edgecolor='white', boxstyle='round,pad=0.5'))
except Exception:
    pass
```

Рисунок 2 – Фрагмент программного кода

График включает панель навигации, сетку, легенду и настроенное форматирование временной оси. Интерфейс позволяет изменять параметры визуализации через диалоговое окно настроек.

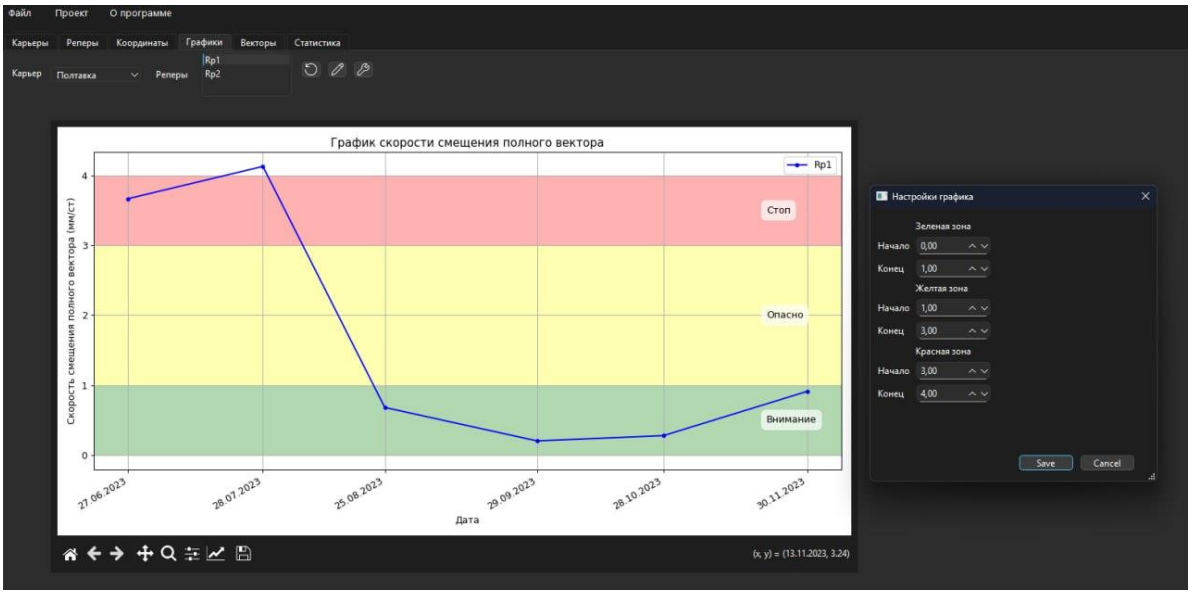

Рисунок 3 – Результат построения графика в программе

На основе проведенного исследования установлено, что разработанное программное решение превосходит традиционные методы визуализации в электронных таблицах. Реализованный подход обеспечивает автоматизацию построения графиков и расширенную функциональность, включая систему цветового кодирования зон безопасности для оперативной оценки критических уровней смещений горных пород.

Интеграция Matplotlib с PySide6 обеспечивает интерактивность визуализации, позволяя пользователям масштабировать графики, панорамировать данные и настраивать параметры отображения. Особое преимущество решения - гибкость настройки параметров зон безопасности через специализированный интерфейс.

Список литературы:

1. Маккинни, У. Python и анализ данных / У. Маккинни; перевод с английского А. А. Слинкина. — 2-ое изд., испр. и доп. — Москва: ДМК Пресс, 2020. — 540 с. — ISBN 978-5-97060-590-5. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/131721 (дата обращения: 18.11.2025). — Режим доступа: для авториз. пользователей.

2. Титов, А. Н. Визуализация данных в Python. Работа с библиотекой Matplotlib: учебно-методическое пособие / А. Н. Титов, Р. Ф. Тазиева. — Казань: КНИТУ, 2022. — 92 с. — ISBN 978-5-7882-3176-1. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/331025 (дата обращения: 21.11.2025). — Режим доступа: для авториз. пользователей.