

РОЗРОБКА СИСТЕМИ АВТОМАТИЗОВАНОГО ФУНКЦІОНАЛЬНОГО АНАЛІЗУ ТРАЄКТОРІЇ РУХУ СУДЕН

Павло Носов, Вікторія Пономарьова, Андрій Бень, Павло Маменко

*Херсонська державна морська академія,
(Україна)*

Вступ. Морська галузь постійно стикається з викликами, пов'язаними з безпекою навігації, точністю керування суднами та ефективністю їхнього руху у складних умовах. Важливою складовою під час підготовки курсантів є тренажерна навігаційна практика, досвід використання сучасних симуляторів для відпрацювання навичок управління рухом судна у складних умовах плавання [1].

Для забезпечення безпечного руху суден через важливі водні шляхи, такі як протока Босфор, судноводіям необхідно мати відповідні навички та використовувати передові навігаційні технології для підтримки точного курсу і швидкості [2]. У той же час, інструктори зазвичай оцінюють результативність виконання певних завдань та маневрів керуючись власним досвідом, що у деяких випадках може носити суб'єктивний характер.

Натомість важливо враховувати дрібні відхилення від курсу та швидкості судна щоб завчасно оцінити рівень імовірної небезпеки, вжити відповідних дій, тощо. Саме тому слід розширювати спектр засобів які б давали більше інформації про результативність виконання вправ курсантами задля підвищення рівня навігаційної безпеки у реальних умовах плавання.

Актуальність дослідження зумовлена сучасними вимогами до безпеки морської навігації. В умовах збільшення кількості суден, що проходять через вузькі протоки та підходять до портів, важливість автоматизованих систем аналізу та підтримки прийняття рішень зростає [3]. Можливість точно аналізувати та прогнозувати рух суден дозволить запобігти аваріям та небезпечним ситуаціям на морі, особливо у складних навігаційних умовах, таких як вузькі протоки або райони з активним судноплаванням.

Отже актуальним завданням у контексті зазначеної проблематики є створення програмного засобу для аналізу траєкторії руху судна та оцінки спроможності судноводія тримати заданий курс. Засіб має застосовувати математичне моделювання для дослідження стабільності руху судна, що є критично важливим під час маневрування у складних умовах. За допомогою різних моделей (лінійних, поліноміальних, експоненціальних, логарифмічних) засіб дозволить об'єктивно оцінювати майстерність судноводіїв, виявляти можливі ризики та розробляти рекомендації для підвищення ефективності навігації.

Постановка задачі. Морська галузь потребує інструментів, які дозволять підвищити безпеку та ефективність навігації, особливо в зонах з високою інтенсивністю руху, наприкладі протоки Босфор що забезпечує пряме сполучення морських шляхів України із акваторіями всього світу. А отже постає необхідність у розробці програмного засобу, який матиме функціонал щодо аналізу траєкторію руху судна та об'єктивної оцінки здатності судноводія тримати заданий курс.

Передбачається що такий інструмент повинен використовувати різні математичні моделі та бібліотеки використовуючи мову Python для моделювання та аналізу даних, що дозволить виявляти проблемні зони в навігації та надавати рекомендації для покращення маневрування.

Завдання полягає у створенні універсальної програми, яка може застосовуватися в різних сферах: від навчання та тренування молодих судноводіїв до оцінки майстерності персоналу судновласниками та операторами компаній. Програма повинна мати можливість інтеграції в системи підтримки прийняття рішень для автоматичного моніторингу та

управління рухом суден, сприяючи тим самим підвищенню рівня безпеки та розвитку навчальних технологій у морській галузі.

Результати дослідження. Отже, виходячи із постановки задачі дослідження, створення програмного засобу передбачає наступні етапи, що описує як хід розробки так і послідовність алгоритмізації роботи програми:

1. Завантаження та обробка даних (`pandas`, `datetime`).

Функція `load_and_clean_data(file_path)` відповідає за завантаження даних з Excel-файлу що отримано із серверу ECDIS:

Використовується бібліотека `pandas` для зчитування даних з файлу Excel:

```
data = pd.read_excel(file_path, engine='xlrd')
```

Метод `read_excel` із зазначенням рушія `xlrd` дозволяє зчитувати Excel-файл. Це є основним способом завантаження табличних даних у Python.

Обробка колонок з координатами (широта, довгота):

Координати можуть мати різні формати (наприклад, коми замість крапок або негативні значення), тому програма конвертує їх у числовий формат:

```
data['LAT'] = pd.to_numeric(data['LAT'].astype(str).str.replace(',', '.', ' ').replace('-', ''), errors='coerce')
data['LON'] = pd.to_numeric(data['LON'].astype(str).str.replace(',', '.', ' ').replace('-', ''), errors='coerce')
```

Метод `astype(str)` перетворює значення у текстовий формат.

`replace(',', '.', ' ')` замінює коми на крапки для коректного зчитування чисел.

`pd.to_numeric(..., errors='coerce')` перетворює текст у числовий формат, а проблемні значення змінює на `NaN`.

Перетворення часу:

Дані часу конвертуються з формату Unix (секунди з 1970 року) у формат `datetime`:

```
data['Time'] = pd.to_datetime(data[time_column], unit='s', origin='unix')
```

Метод `pd.to_datetime` перетворює числовий формат часу в стандартний `datetime`, що спрощує роботу з часом.

Очищення даних від некоректних рядків:

Після конвертації, програма видаляє всі рядки, які містять `NaN` в ключових колонках (LAT, LON, Time):

```
return data.dropna(subset=['LAT', 'LON', 'Time'])
```

Використовується метод `dropna`, що дозволяє усунути неповні дані.

2. Моделювання даних (`numpy`, `scipy.optimize.curve_fit`):

Для аналізу використовуються математичні моделі для апроксимації даних [4]:

Лінійна модель:

```
def linear_function(x, a, b): return a * x + b
```

Поліноміальна модель: `def polynomial_function(x, a, b, c): return a * x**2 + b * x + c`

Експоненціальна модель: `def exponential_function(x, a, b): return a * np.exp(b * x)`

Логарифмічна модель: `def logarithmic_function(x, a, b): return a * np.log(x + 0.01) + b`

Для підбору параметрів кожної з моделей використовується метод `curve_fit` з бібліотеки `scipy.optimize`:

```
popt_speed_lin, _ = curve_fit(linear_function, time_seconds, y_speed)
```

Метод `curve_fit` дозволяє підбирати параметри функції, що найкраще описує дані. Він використовує метод найменших квадратів для мінімізації різниці між експериментальними і апроксимованими даними.

3. Аналіз сегментів:

Функція `analyze_segment(data, start_min, end_min)` виконує:

Визначення часових меж для сегменту:

Використовується `timedelta` для роботи з проміжками часу:

```
start_time = data['Time'].iloc[0] + timedelta(minutes=start_min)
end_time = data['Time'].iloc[0] + timedelta(minutes=end_min)
```

Це дозволяє вибрати дані тільки за вказаний проміжок часу для аналізу.

Для кожної моделі (лінійної, поліноміальної, експоненційної, логарифмічної)

виконується підбір параметрів, а у разі помилки програма виводить повідомлення:

```
try:    popt_speed_lin, _ = curve_fit(linear_function, time_seconds, y_speed)
except RuntimeError as e:
print(f"Linear fit failed for speed: {e}")
```

Кожна модель намагається підібрати свої параметри. Якщо це не вдається через недостатню кількість даних або інші причини, програма обробляє помилку.

4. Візуалізація (matplotlib):

Програма будує графіки, що показують реальні дані і апроксимовані функції:

```
fig, axes = plt.subplots(2, 4, figsize=(20, 10))
axes[0, 0].plot(time_seconds, y_speed, 'b-', label='Speed [NM]')
axes[0, 1].plot(time_seconds, linear_function(time_seconds,
*results['speed_linear']), 'r--', label='Linear Fit')
axes[0, 0].set_title('Linear Fit (Speed)')
```

Використовується бібліотека `matplotlib` для побудови графіків:

`plt.subplots(2, 4)` створює сітку з 2 рядків і 4 стовпців для кількох графіків.

`axes[0, 0].plot()` будує лінії для реальних даних та апроксимованої функції.

`set_title()` додає заголовок до кожного графіка.

На рисунку 1, представлено візуалізацію траєкторій на двох контрольних проміжках руху судна у протоці Босфор.

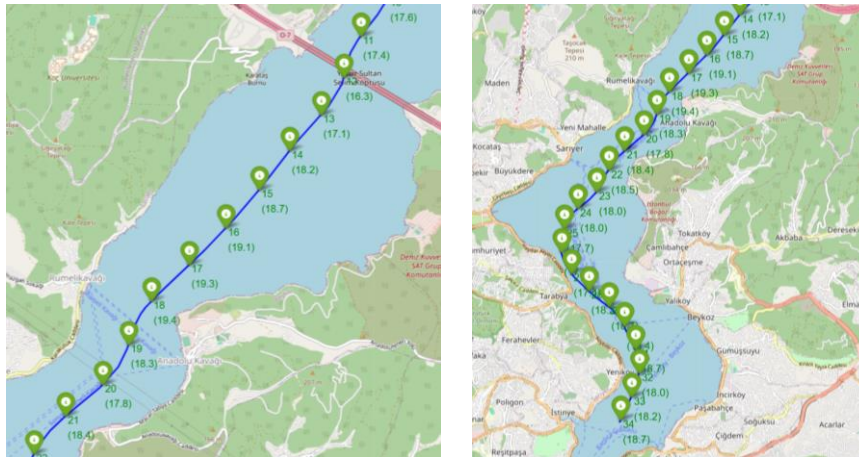


Рисунок 1 – Візуалізація траєкторій на проміжках 13-18 хв та 18-24 хв.

Розраховані відхилення за моделями апроксимації (Рис. 2) даних є наступні:

```
Speed linear function parameters: [7.09535695e-03 1.76741063e+01]
Speed polynomial function parameters: [-2.89322565e-05 1.57750339e-02
1.72473555e+01]
Speed exponential function parameters: [1.77039390e+01 3.74992362e-04]
Speed logarithmic function parameters: [ 0.36108055 17.0774474 ]
Course linear function parameters: [2.53093601e-02 2.18269170e+02]
Course polynomial function parameters: [ 1.92842437e-04 -3.25433716e-02
2.21113596e+02]
Course exponential function parameters: [2.18270251e+02 1.14583954e-04]
Course logarithmic function parameters: [ 0.7076977 218.81018019]
```

Дані та графіки вказують на достатньо високий рівень керування рухом судна утримуючі заданий курс в протоці.

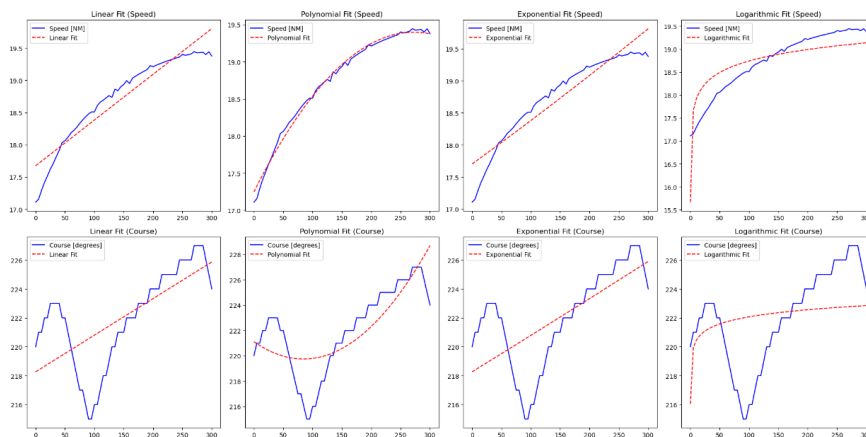


Рисунок 2 – Графіки для аналізу руху судна на проміжку 13-18 хв

Наступним проміжком для аналізу руху судна є в межах 18-24 хв. (Рис. 3):

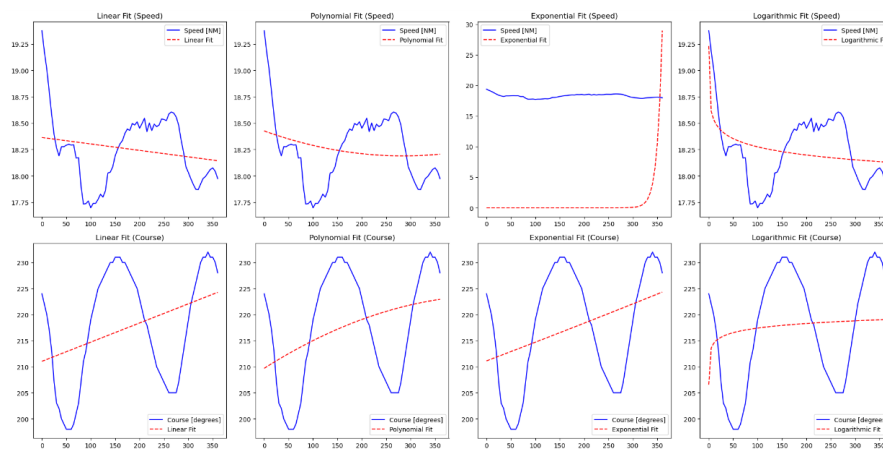


Рисунок 3 – Приклад підпису (назви) рисунка

Speed linear function parameters: $[-6.11298287e-04 \ 1.83634857e+01]$
 Speed polynomial function parameters: $[\ 2.92123548e-06 \ -1.66294384e-03 \ 1.84257081e+01]$
 Speed exponential function parameters: $[6.71292034e-15 \ 9.99999921e-02]$
 Speed logarithmic function parameters: $[-0.11218481 \ 18.79172778]$
 Course linear function parameters: $[3.67703320e-02 \ 2.11038874e+02]$
 Course polynomial function parameters: $[-6.29674651e-05 \ 5.94386186e-02 \ 2.09697668e+02]$
 Course exponential function parameters: $[2.11121489e+02 \ 1.68510822e-04]$
 Course logarithmic function parameters: $[\ 1.27248945 \ 211.55198228]$

Аналіз траєкторії руху судна зроблений на даних що було отримано під час тренажерної підготовки. Даний варіант даних вказує на значне перевищення швидкості від рекомендованої, у той самий час ці дані наявно вказують на деякі порушення, що дозволяє програмному засобу дійти певних висновків.

Результати автоматизованого моделювання вказують на те, що проміжок 13-18 хвилин має наступні ознаки кваліфікації навігатора: судноводій здатен утримувати стабільний курс та швидкість (хоча і перевищену) на цьому етапі. Відповідно до параметрів лінійної, поліноміальної, експоненціальної та логарифмічної моделей для швидкості, варто зазначити, що відхилення незначні, що свідчить про високу точність маневрування. Математичні моделі для курсу також показують стабільну траєкторію без серйозних відхилень. Це свідчить про достатній рівень кваліфікації судноводія в утриманні курсу в складних умовах.

Для проміжку 18-24 хвилин: спостерігається дещо більше відхилень у швидкості та курсі судна. Параметри моделей для швидкості свідчать про певну нестабільність керування, що видно з коефіцієнтів для експоненціальної та логарифмічної моделей. Аналіз курсу також вказує на коливання, хоча лінійна та поліноміальна моделі свідчать про здатність судноводія контролювати загальний напрямок. Ці результати можуть свідчити про наявність зовнішніх факторів або зниження точності маневрування в даній зоні протоки Босфор.

Висновки. Програма є дієвим додатковим інструментом для аналізу судноводіння щодо оцінки стабільності курсу та швидкості судна за допомогою різних математичних моделей. Також засіб дозволяє виявляти можливі проблем у навігації: непослідовності у русі судна можуть бути легко виявлені за допомогою візуальних та кількісних результатів.

Особливим аспектом застосування програми є тренування судноводіїв: за допомогою детальних графіків та аналізу можна оцінити техніку судноводіння та надати рекомендації. Цей фактор дозволить зменшити час та звузити пошук проблем під час підготовки судноводіїв при застосуванні навігаційних тренажерів.

Із технічного аспекту, можна виділити, що програма використовує потужні бібліотеки Python: `pandas` для обробки даних, `scipy` для підбору функцій, `matplotlib` для візуалізації та `numpy` для числових операцій, що робить її надійною та ефективною для морського аналізу.

Загалом, розроблений програмний засіб дозволить в окремих аспектах підвищити рівень безпеки під час керування рухом судна в умовах ризику та навігаційних невизначень [5].

ЛІТЕРАТУРА

1. Mou, Fangli & Fan, Zide & Li, Xiaohe & Wang, Lei & Li, Xinming. (2023). A Method for Clustering and Analyzing Vessel Sailing Routes Efficiently from AIS Data Using Traffic Density Images. *Journal of Marine Science and Engineering*. 12. 75. <https://doi.org/10.3390/jmse12010075>.
2. Kyaw, Aung. (2024). Application of Advanced Technology on Transport Ships as a Technological Revolution in the Maritime Industry. *Maritime Park Journal of Maritime Technology and Society*. 1-7. <https://doi.org/10.62012/mp.v3i2.35384>.
3. Ponomaryova, V., Nosov, P., Ben, A., Popovych, I., Prokopchuk, Y., Mamenko, P., Dudchenko, S., Appazov, E., & Sokol, I. (2024). Devising an approach for the automated restoration of shipmaster's navigational qualification parameters under risk conditions. *Eastern-European Journal of Enterprise Technologies*, 1(3 (127), 6–26. <https://doi.org/10.15587/1729-4061.2024.296955>
4. Osuagwu, U. & Okenwe, I. (2024). Simple Regression Models: A Comparison using Criteria Measures. *African Journal of Mathematics and Statistics Studies*. 7. 131-143. <https://doi.org/10.52589/AJMSS-VKLVNUU5>.
5. Mamenko P., Zinchenko S., Kobets V., Nosov P., Popovych I. (2022) Solution of the Problem of Optimiz-ing Route with Using the Risk Criterion. In: Babichev S., Lytvynenko V. (eds) *Lecture Notes in Computational In-telligence and Decision Making. ISDMCI 2021. Lecture Notes on Data Engineering and Communications Tech-nologies*, vol 77. Springer, Cham. https://doi.org/10.1007/978-3-030-82014-5_17.

DEVELOPMENT OF A SYSTEM FOR AUTOMATED FUNCTIONAL ANALYSIS OF SHIP TRAJECTORIES

Pavlo Nosov, Victoriia Ponomariova, Andrii Ben, Pavlo Mamenko

Kherson State Maritime Academy

(Ukraine)

Introduction. The research is devoted to the creation of a universal program that should be able to be integrated into decision support systems for automatic monitoring and control of ship traffic, thus contributing to the improvement of the level of safety and the development of educational technologies in the maritime sector