

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інженерії
програмного забезпечення

_____ Євген ДАВИДЕНКО

«__» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ОЦІНКИ ПОКАЗНИКІВ
НАВІГАЦІЙНОГО МАНЕВРУВАННЯ МОРСЬКОГО ТРАНСПОРТУ

Спеціальність 121 Інженерія програмного забезпечення
Освітня програма «Інженерія програмного забезпечення»

Здобувач

Микола СКИБА

«__» _____ 2025 р.

Керівник роботи

PhD, _____ доцентка

кафедри ПЗ

Катерина АНТІПОВА

«__» _____ 2025 р.

Миколаїв – 2025

Завдання на виконання кваліфікаційної роботи

Чорноморський національний університет імені Петра Могили

Факультет	Комп'ютерних наук
Кафедра	Інженерії програмного забезпечення
Рівень вищої освіти	Другий (магістерський)
Освітній ступінь	Магістр
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри інженерії
програмного забезпечення

_____ Євген ДАВИДЕНКО

«___» _____ 2025 р.

ЗАВДАННЯ

на кваліфікаційну магістерську роботу здобувача вищої освіти

Скиби Миколи

1. Тема кваліфікаційної роботи Програмне забезпечення оцінки показників навігаційного маневрування морського транспорту затверджена наказом ректора ЧНУ ім. Петра Могили № 182 від «2» липня 2025 р.

2. Строк представлення кваліфікаційної роботи «___» _____ 2025 р.

3. Очікуваний результат роботи та початкові дані якщо такі потрібні.

Очікуваним результатом є програмне забезпечення підтримки процесу маневрування морського судна за допомогою розрахунку розташування полюса повороту і візуалізації його переміщення при маневруванні.

4. Перелік питань, що підлягають розробці:

– аналіз проблематики керування судном та обґрунтування актуальності роботи, можливостей застосування судових навігаційних систем, особливостей виконання маневрування судном, обґрунтування переваг використаних засобів програмної розробки;

– формалізація методологічних положень та алгоритмів з логіки виконання розрахунків;

– розробка проекту та структури програмного забезпечення, опис ключових методів програмної логіки;

– розробка ключових функціональних можливостей та опис процесу використання програмного застосування;

– дослідження можливостей використання програмного забезпечення.

5. Перелік графічних матеріалів:

– презентація.

6. Консультанти:

Консультант	Кафедра (організація)	Частина роботи

Дата видачі завдання « ____ » _____ 20__ р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: Програмне забезпечення оцінки показників навігаційного маневрування морського транспорту

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КМР	01.09.2025	02.09.2025	виконано
2.	Огляд літератури за темою роботи	03.09.2025	04.09.2025	виконано
3.	Складання календарного плану КМР	05.09.2025	05.09.2025	виконано
4.	Аналіз предметної області	08.09.2025	12.09.2025	виконано
5.	Розробка проєктних рішень	15.09.2025	19.09.2025	виконано
6.	Моделювання та конструювання ПЗ	22.09.2025	26.09.2025	виконано
7.	Кодування та тестування розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	29.09.2025	05.10.2025	виконано
8.	Відгук керівника КМР	14.12.2025	15.12.2025	виконано
9.	Оформлення КМР та презентації	16.11.2025	23.11.2025	виконано
10.	Попередній захист	24.11.2025	24.11.2025	виконано
11.	Рецензування	13.12.2025	15.12.2025	виконано
12.	Завершення оформлення КМР та презентації	14.12.2025	16.12.2025	виконано
13.	Захист кваліфікаційної роботи			

Здобувач _____

Микола СКИБА

«__» _____ 2025 р.

Керівник роботи

PhD, доцентка

кафедри ІПЗ _____

Катерина АНТІПОВА

«__» _____ 2025 р.

АНОТАЦІЯ

до кваліфікаційної магістерської роботи

Програмне забезпечення оцінки показників навігаційного маневрування морського транспорту

Здобувач 608м гр.: Скиба Микола

Керівник: PhD, доцентка Антіпова Катерина

Актуальність обраної теми полягає в тому, що широке впровадження інформаційних технологій забезпечення безпеки судноплавства для цілей маневрування морського судна докорінно змінило умови роботи судноводія. Це дозволило меншою кількістю команди містка великотоннажного судна ефективно та безпечно керувати його маневруванням. При цьому використовуються сучасні навігаційні пристрої, які дозволяють забезпечувати інформаційну підтримку прийняття рішення при керуванні процесом руху судна по маршруту переходу.

Найефективнішою судновою навігаційною системою, яка змінила характер роботи штурманського складу, є навігаційно-інформаційна система з електронною картою (ECDIS). Окрім звичайних функцій паперової карти, вона дозволяє забезпечити значні переваги щодо виконання окремих елементів штурманської роботи з планування безпечного маршруту переходу і функцій підтримки прийняття рішення при керуванні його рухом і маневруванням.

Об'єктом кваліфікаційної роботи є процес навігаційного маневрування морського транспорту.

Предметом кваліфікаційної роботи є методи оцінки показників навігаційного маневрування судном.

Метою кваліфікаційної роботи є підвищення ефективності інформаційної підтримки процесів маневрування судном, шляхом розробки програмного забезпечення оцінки показників навігаційного маневрування морського транспорту.

Для досягнення мети необхідно виконати наступні завдання:

– аналіз проблематики керування судном та обґрунтування актуальності роботи, можливостей застосування судових навігаційних систем, особливостей

виконання маневрування судном, обґрунтування переваг використаних засобів програмної розробки;

- формалізація методологічних положень та алгоритмів з логіки виконання розрахунків;

- розробка проєкту та структури програмного забезпечення, опис ключових методів програмної логіки;

- розробка ключових функціональних можливостей та опис процесу використання програмного застосування;

- дослідження можливостей використання програмного забезпечення.

Кваліфікаційна робота складається із вступу, 4 розділів, висновків та переліку джерел посилання.

У вступі описано тему, мету, предмет, об'єкт та завдання кваліфікаційної роботи, наведено додаткову інформацію щодо використаного матеріалу.

У першому розділі проаналізовано особливості технічного керування судном, ризики та фактори, що впливають на безпеку плавання. Визначено ключову роль сучасних навігаційних систем (GNSS, AIS, INS) у підвищенні точності та ефективності керування. Показано важливість врахування полюса повороту судна як критичного параметра маневрування та обґрунтовано необхідність його автоматизованого визначення. На основі отриманих результатів сформовано передумови для подальшого дослідження математично-алгоритмічної частини розробки.

Другий розділ присвячено формалізації процесів оцінки показників маневрування судна та розробці алгоритмічної основи для їх практичної реалізації. Сформульовано методологічні положення щодо опису руху судна, визначення параметрів його положення і маневреності. Обґрунтовано точне визначення абсциси полюса повороту та точки прикладання рівнодіючої сил. Розроблено блок-схеми алгоритмів автоматизованого та напівавтоматичного визначення положення полюса повороту з урахуванням зовнішніх впливів і похибок.

У третьому розділі виконана розробка програмного забезпечення оцінки показників навігаційного маневрування морського транспорту. Обґрунтовано

застосування засобів програмної розробки, описано структуру ПЗ, ключові методи програмної логіки, основні функціональні можливості та процес використання програмного застосування.

Четвертий розділ присвячено дослідженню можливостей моделювання сценаріїв маневрування судном на базі системи Matlab. Використано фільтр Калмана для оцінки стабільності та точності маневрування, проаналізовано результати симуляції руху судна за заданою траєкторією. Застосування фільтра Калмана дозволяє підвищити безпеку на борту та знизити ймовірність аварійних ситуацій, пов'язаних із помилками при маневруванні.

У висновках описано досягнення мети роботи: підвищення ефективності інформаційної підтримки процесів маневрування судном шляхом розробки програмного забезпечення для оцінки показників маневрування. Проведено аналіз результатів, виконано розробку алгоритмів, структури ПЗ, функціональних можливостей та оцінку ефективності використання програмного забезпечення при моделюванні різних сценаріїв маневрування.

Кваліфікаційна робота викладена на 94 сторінках машинописного тексту, складається із вступу, 4 розділів, загальних висновків, переліку джерел посилення з 30 найменувань та 1 додатків. Праця містить 1 таблицю та 43 рисунки.

Ключові слова: маневрування судном, оцінка траєкторії руху, фільтр Калмана, візуальне моделювання, Python.

ABSTRACT

to the qualifying master's thesis

Software for evaluating navigation maneuvering performance of marine transport

Student of 608m group: Skyba Mykola

Supervisor: PhD, Associate Professor Antipova Kateryna

The relevance of the chosen topic lies in the fact that the widespread introduction of information technologies to ensure shipping safety for the purposes of maneuvering a seagoing vessel has radically changed the working conditions of a ship's pilot. This has allowed a smaller number of bridge crews of a large-tonnage vessel to effectively and safely manage its maneuvering. At the same time, modern navigation devices are used that allow providing information support in accordance with the decision when managing the process of vessel movement along the transition route.

The most effective ship navigation system that has changed the nature of the work of the navigation staff is the electronic chart navigation information system (ECDIS). free of charge, the usual functions of a paper map, it allows providing significant advantages in the implementation of individual elements of navigation work for planning a safe transition route and supports decision-making when managing its movement and maneuvering.

The object of the qualification work is the process of navigational maneuvering of marine transport.

The subject of the qualification work is the methodology for assessing the indicators of navigational maneuvering by a ship.

The purpose of the qualification work is to increase the efficiency of information support for ship maneuvering processes by developing software for evaluating navigation maneuvering indicators of marine transport.

To achieve the goal, the following tasks need to be completed:

- analysis of ship control issues and justification of the relevance of the work, the possibility of using ship navigation systems, the features of ship maneuvering, justification of the advantages of the software development tools used;

- formalization of methodological provisions and algorithms for the logic of calculation execution;
- development of the project and structure of the software, description of key methods of program logic;
- development of key functional capabilities and description of the process of using the software application;
- research into the possibilities of using the software.

The introduction describes the topic, aim, subject, object, and tasks of the qualification work, as well as provides additional information about the materials used.

In the first chapter, the features of technical ship handling, risks, and factors affecting navigational safety are analyzed. The key role of modern navigation systems (GNSS, AIS, INS) in improving the accuracy and efficiency of ship control is highlighted. The importance of considering the ship's pivot point as a critical parameter of maneuvering is demonstrated, and the necessity of its automated determination is justified. Based on the obtained results, the prerequisites for further research of the mathematical and algorithmic part of the development are formed.

The second chapter is dedicated to the formalization of the processes for evaluating ship maneuvering indicators and the development of an algorithmic basis for their practical implementation. Methodological principles for describing ship movement, determining its position, and maneuverability parameters are formulated. The precise determination of the pivot point coordinate and the point of application of the resultant forces is justified. Flowcharts of algorithms for automated and semi-automated determination of the pivot point position, taking into account external influences and errors, are developed.

In the third chapter, software for evaluating the indicators of maritime transport maneuvering is developed. The use of software development tools is justified, the structure of the software, key programming logic methods, main functional capabilities, and the process of using the software application are described.

The fourth chapter focuses on the study of the possibilities of modeling ship maneuvering scenarios using Matlab. The Kalman filter is applied to assess the stability

and accuracy of maneuvering, and the results of ship movement simulation along a specified trajectory are analyzed. The use of the Kalman filter allows for more precise and stable ship control, increasing onboard safety and reducing the likelihood of accidents associated with incorrect turn calculations or sudden maneuvers.

The conclusions describe the achievement of the work's aim: improving the efficiency of informational support for ship maneuvering processes through the development of software for evaluating maneuvering indicators. The analysis of results is conducted, the development of algorithms, software structure, functional capabilities, and evaluation of the software's effectiveness in modeling various maneuvering scenarios are completed.

The qualification work is presented on 94 pages of typewritten text, consisting of an introduction, 4 sections, general conclusions, a list of source references from 30 items and 1 appendices. The work contains 1 tables and 43 figures.

Keywords: vessel maneuvering, trajectory estimation, Kalman filter, visual modeling, Python.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	4
ВСТУП.....	5
1 АНАЛІЗ КЛЮЧОВИХ АСПЕКТІВ ТА ПОЛОЖЕНЬ МАНЕВРУВАННЯ СУДНОМ.....	7
1.1 Аналіз проблематики керування судном та обґрунтування актуальності роботи	7
1.2 Аналіз аналогів програмного забезпечення для оцінки показників навігаційного маневрування суден.....	11
1.2.1 DynaSim (Dynaflow, Inc.).....	11
1.2.2 SHIPMA (MARIN / Conceptia).....	12
1.3 Аналіз можливостей застосування суднових навігаційних систем	13
1.4 Аналіз особливостей виконання маневрування судном.....	17
Висновки до розділу 1	20
2 ФОРМАЛІЗАЦІЯ ПРОЦЕСІВ ОЦІНКИ ПОКАЗНИКІВ МАНЕВРУВАННЯ СУДНА	21
2.1 Методологія визначення параметрів маневрування судна	21
2.2 Алгоритми автоматизованого визначення полюса повороту	26
2.3 Специфікація вимог до програмного забезпечення.....	36
Висновки до розділу 2.....	39
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ОЦІНКИ ПОКАЗНИКІВ НАВІГАЦІЙНОГО МАНЕВРУВАННЯ МОРСЬКОГО ТРАНСПОРТУ	40
3.1 Обґрунтування використаних засобів програмної розробки	40
3.1.1 Мова Python.....	40
3.1.2 IDE PyCharm	42

	3
3.1.3 Мова розмітки QML	42
3.2 Розробка проєкту програмного застосування	43
3.3 Розробка ключових методів програмної логіки.....	52
3.4 Розробка ключових функціональних можливостей та опис процесу використання програмного застосування	55
Висновки до розділу 3	73
4 ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ МОДЕЛЮВАННЯ СЦЕНАРІЇВ МАНЕВРУВАННЯ СУДНОМ	74
4.1 Формалізація процесу використання фільтра Калмана для дослідження сценарії моделювання маневрування	74
4.2 Дослідження роботи моделі симуляції руху судна за заданою траєкторією руху	76
Висновки до розділу 4.....	83
ВИСНОВКИ	84
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	85
ДОДАТОК А (Main.pyw).....	88

ПЕРЕЛІК СКОРОЧЕНЬ

ДП – діаметральна площа

ОС – операційна система

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

ПП – полюс повороту

СНС – суднові навігаційні системи

МГО – міжнародна гідрографічна організація

МЕК – міжнародна електротехнічна комісія

AIS – Automatic Identification System

ECDIS – Electronic Chart Display and Information System

GNSS – Global Navigation Satellite Systems

GPS – Global Positioning Systems

IMO – International Maritime Organization

INS – Inertial Navigation Systems

SOLAS – Safety of Life at Sea

ВСТУП

Актуальність обраної теми полягає в тому, що широке впровадження інформаційних технологій забезпечення безпеки судноплавства для цілей маневрування морського судна докорінно змінило умови роботи судноводія. Це дозволило меншою кількістю команди містка великотоннажного судна ефективно та безпечно керувати його маневруванням. При цьому використовуються сучасні навігаційні пристрої, які дозволяють забезпечувати інформаційну підтримку прийняття рішення при керуванні процесом руху судна по маршруту переходу.

Найефективнішою судновою навігаційною системою, яка змінила характер роботи штурманського складу, є навігаційно-інформаційна система з електронною картою (ECDIS). Окрім звичайних функцій паперової карти, вона дозволяє забезпечити значні переваги щодо виконання окремих елементів штурманської роботи з планування безпечного маршруту переходу і функцій підтримки прийняття рішення при керуванні його рухом і маневруванням.

Об'єктом кваліфікаційної роботи є процес навігаційного маневрування морського транспорту.

Предметом кваліфікаційної роботи є методи оцінки показників навігаційного маневрування судном.

Метою кваліфікаційної роботи є підвищення ефективності інформаційної підтримки процесів маневрування судном, шляхом розробки програмного забезпечення оцінки показників навігаційного маневрування морського транспорту.

Для досягнення мети необхідно виконати наступні завдання:

- аналіз проблематики керування судном та обґрунтування актуальності роботи, можливостей застосування судових навігаційних систем, особливостей виконання маневрування судном, обґрунтування переваг використаних засобів програмної розробки;

- формалізація методологічних положень та алгоритмів з логіки виконання розрахунків;

- розробка проекту та структури програмного забезпечення, опис ключових методів програмної логіки;
- розробка ключових функціональних можливостей та опис процесу використання програмного застосування;
- дослідження можливостей використання програмного забезпечення.

1 АНАЛІЗ КЛЮЧОВИХ АСПЕКТІВ ТА ПОЛОЖЕНЬ МАНЕВРУВАННЯ СУДНОМ

1.1 Аналіз проблематики керування судном та обґрунтування актуальності роботи

Технічне керування судном є важливою складовою судноплавства, яка охоплює низку аспектів, пов'язаних із забезпеченням безпечної та ефективної експлуатації судна. Основні аспекти, ризики та проблеми технічного керування судном включають наступне:

1) Технічне обслуговування і ремонт. Це передбачає проведення регулярних перевірок судових систем, заміну обладнання, проведення планових ремонтних робіт та модернізації судна. Недостатнє або невчасне обслуговування може призвести до серйозних аварій або поломок. Відсутність кваліфікованого персоналу для обслуговування, обмежений бюджет на ремонт або використання низькоякісних запчастин.

2) Керування паливом і енергоефективність. Вибір правильного типу палива, контроль за витратами палива, дотримання норм з енергоефективності (наприклад, MARPOL). Неефективне використання палива призводить до високих витрат і забруднення довкілля. Вибір неправильного палива може спричинити поломки двигуна або інші технічні проблеми. Існує також потреба в постійній оптимізації витрат палива та використання нових технологій (наприклад, альтернативних видів палива), які можуть бути дорогими.

3) Суднові системи і обладнання, що включає навігаційне, електричне, машинне та інші ключові системи на борту. Несправність будь-якої з цих систем може поставити під загрозу безпеку судна, зокрема старіння обладнання, недостатній моніторинг або контроль роботи систем.

4) Забезпечення відповідності міжнародним стандартам. Дотримання вимог Міжнародної морської організації (IMO), Конвенції SOLAS (Safety of Life at Sea), MARPOL (забруднення з суден), а також національних і міжнародних правил [17].

Порушення стандартів може призвести до штрафів, затримання суден або втрати ліцензії на експлуатацію, також постійні зміни в міжнародному законодавстві вимагають регулярного оновлення знань і адаптації до нових вимог.

5) Моніторинг і контроль стану судна. Використання сучасних систем для дистанційного моніторингу стану судна, систем діагностики для запобігання поломкам. Можливими є поломки без попередження через недостатній моніторинг або некоректну інтерпретацію даних. Головною проблемою в даному випадку є висока вартість впровадження сучасних систем і необхідність регулярного оновлення програмного забезпечення.

Автоматизація процесу визначення координат судна під час його маневрування є важливою складовою сучасного судноводіння, оскільки вона підвищує точність навігації, безпеку і ефективність операцій. Основними шляхами автоматизації цього процесу є використання різних технологій, які забезпечують точний і постійний моніторинг позиції судна. Ключовими підходами у сучасному контексті проблеми є наступні:

1) Глобальні навігаційні супутникові системи (GNSS). Основним методом автоматизації визначення координат судна є використання GNSS [18]. Сучасні суднові навігаційні системи оснащені приймачами GNSS, що дозволяє отримувати дані про точне місце розташування судна в режимі реального часу.

Переваги: висока точність (до кількох метрів), доступність в будь-якій точці світу, можливість автоматичної інтеграції з іншими системами судна.

Недоліки: залежність від супутникового сигналу, який може бути знижений або порушений через погодні умови, географічне розташування (наприклад, у фіордах чи між високими будівлями) або глушіння сигналу.

2) Автоматична ідентифікаційна система (AIS), що є системою автоматичної передачі даних про місцезнаходження суден, що дозволяє відстежувати рухи суден у режимі реального часу. AIS використовує GNSS для визначення координат і передачі цієї інформації через радіоканали.

Переваги: забезпечує постійний моніторинг інших суден у зоні видимості, запобігає зіткненням, є міжнародним стандартом для комерційних суден.

Недоліки: обмежена зона дії (на відстань до 40 морських миль від узбережжя або станцій прийому), залежність від інших суден і берегових станцій, які також повинні бути оснащені AIS.

3) Інерційні навігаційні системи (INS), використовують гіроскопи, акселерометри і спеціальні датчики для визначення зміщень та прискорення судна, що дозволяє обчислювати його положення в просторі навіть за відсутності GNSS-сигналу.

Переваги: INS працює незалежно від зовнішніх сигналів, надає точні дані навіть у місцях без покриття GNSS (тунелі, порти).

Недоліки: з часом INS може втрачати точність через накопичення похибок, що вимагає періодичного калібрування за допомогою GNSS або інших систем.

4) Радарні системи, використовуються для визначення відстаней до навколишніх об'єктів, таких як береги, інші судна, буї. Дані з радара можуть бути автоматично інтегровані з навігаційними системами для уточнення місцезнаходження судна.

Переваги: радары працюють незалежно від супутникових систем, особливо корисні в умовах обмеженої видимості (туман, ніч).

Недоліки: менша точність у порівнянні з GNSS, обмежена зона дії (особливо у відкритому океані).

5) Лідари і оптичні системи. Використовують світлові хвилі для вимірювання відстаней до об'єктів. Ці технології можуть допомагати в точній навігації під час маневрування в портах чи вузьких місцях.

Переваги: висока точність при близькому наближенні до об'єктів, корисні для причалювання.

Недоліки: висока вартість, залежність від умов освітлення і видимості.

Окремо слід відмітити аспекти маневрування корпусом судна. При поступальному прямолінійному русі судна всі його точки переміщуються по

паралельних траєкторіях і мають однакову лінійну швидкість. При маневруванні корпус здійснює обертальний рух, коли точки тіла мають різну швидкість і існує така точка, лінійна швидкість якої дорівнює нулю. Стосовно до судна таку точку називають полюсом повороту (ПП). Знання того, де знаходиться ПП, надзвичайно важливо для судноводія при маневруванні в обмежених умовах, так як геометричні розміри судна порівнянні з акваторією для маневрування, відстанями до небезпек і знаків навігаційного огороження.

З цієї причини судноводій використовує оглядово-порівняльний метод керування маневруванням. Особливості зорового сприйняття напрямків і відстаней до орієнтирів при обертальному русі призводить до того, що судноводій здається, що обертання відбувається навколо нього, в той час коли фактично відбувається навколо ПП. Це призводить до невірної оцінки напрямків і відстаней до орієнтирів, і виникають передумови для появи аварійної ситуації.

Крім того стратегія використання буксирів вимагає знання становища ПП для вибору точки прикладання сили від них, що дозволяє оптимізувати їх розстановку і дає можливість меншою потужністю отримати максимальний момент від сили упору буксира, шляхом вибору точки прикладання, чим далі від ПП, тим він більше.

Якщо інформація про положення ПП відсутня, судноводій формує керуючі впливи в застосунку на підставі власного досвіду керування конкретним судном та інтуїтивних оцінок. Оптимізувати процес керування йому дозволить допоміжна інформація про процес маневрування, яку йому буде поставляти ЕОМ, обробляючи сигнали, що надходять від датчиків, на підставі закладеної в неї математичної моделі руху судна.

Застосовуючи такий підхід, судноводій зможе керувати процесом маневрування більш точно і безаварійно.

Історично склалося так, що формування розуміння закономірностей маневрування морського судна купувалося при практичній роботі методом проб і помилок. Хронологія формування способу використання інформації про параметри

маневрування взагалі і вплив положення ПП на процес керування судном викладені в роботі [1]. У ній показано, що значення абсциси ПП визначає ширину маневреного зміщення, тому він може бути використаний для нормування параметрів поворотності. Обмеження максимального значення абсциси ПП на циркуляції фактично дозволить унормувати значення ширини смуги руху судна в залежності від його головних розмірів.

Крім того зазначається особливість параметра $X_{пп}$, яка проявляється в тому, що при повороті судна абсциса ПП досягає значення, відповідного сталої циркуляції даної кривизни, ще на початку еволюційного періоду. Це дає можливість приймати в розрахунках геометричних параметрів повороту судна значення $X_{пп}$ для сталої циркуляції даного радіуса. Наведено формалізовані моделі визначення параметрів криволінійного руху. Однак розрахункові схеми і алгоритми створення навігаційних пристроїв, для визначення положення ПП, не розроблені.

В роботі [2] розглянуто особистий практичний досвід капітана з керування процесом маневрування морського судна у вигляді змістовних моделей виконання різних морських операцій. При цьому використовується оглядово порівняльний метод інформаційного забезпечення процесу маневрування і метод проб і помилок для набуття практичних навичок роботи. Формалізовані моделі керування процесом маневрування не наведено, а розрахункові схеми визначення абсциси $X_{пп}$ не обговорюються. Актуальною задачею є автоматизація визначення координат $X_{пп}$ за допомогою розробки прикладного програмного забезпечення з можливостями візуалізації отриманих результатів.

1.2 Аналіз аналогів програмного забезпечення для оцінки показників навігаційного маневрування суден

1.2.1 DynaSim (Dynaflow, Inc.)

Опис:

— це симулятор маневрування суден на базі ПК;

– модель включає компоненти корпусу, двигунів, гвинтів, рулів, бокових рушіїв, а також враховує навколишні умови – вітер, течії, хвилі, дно, набережні тощо;

– працює в трьох режимах: ручне керування (користувач керує рулем, рушіями), стандартні маневри за ІМО (наприклад, круг повороту, «зигзаг»), і комп'ютерне керування (автопілот, трафік);

– графічний інтерфейс, підтримка кількох суден одночасно (multi-thread), підрахунок витрат палива, симуляція буксирів [3].

Переваги:

– висока достовірність гідродинамічного моделювання через розвинену математичну модель;

– враховує зовнішні фактори: вітер, хвилі, течію, дно що дає реалістичну симуляцію маневрування;

– можливість запуску на звичайному ПК, без необхідності великих тренажерних установок;

– підтримка сертифікаційних маневрів (ІМО), що корисно для досліджень або сертифікації;

– гнучке налаштування моделей: користувач може вводити характеристики власного судна.

Недоліки:

– як комерційне рішення, вартість може бути значною, особливо якщо потрібна ліцензія або кастомізація;

– потребує вводу детальних даних про судно (корпус, рушії, рама), що може бути ресурсоємним;

– не завжди легко інтегрувати або модифікувати під специфічні дослідницькі задачі без технічної експертизи.

1.2.2 SHIPMA (MARIN / Conceptia)

Опис:

- SHIPMA – fast-time симулятор маневрування суден, розроблений MARIN (Maritime Research Institute Netherlands) [4];
- призначений для симуляції поведінки суден в портах, каналах, внутрішніх водах;
- містить бібліотеку моделей суден (близько 100 різних моделей), які можна вибирати або налаштовувати;
- інтерфейс з графічним деревом (GUI), через яке задаються параметри моделювання (оточення, модель судна, траєкторії).

Переваги:

- дуже висока точність моделей – MARIN має великий досвід у гідродинаміці та маневруванні;
- сильна бібліотека типових кораблів, що дозволяє використовувати готові моделі без створення з нуля;
- підходить для досліджень транспортної інфраструктури: проектування каналів, портів, аналіз безпеки трафіку;
- ефективний для «fast-time» моделювання, тобто симуляції великої кількості сценаріїв за короткий час (не повністю в реальному часі).

Недоліки:

- «fast-time» означає, що симуляція не дає детальної «візуалізації дії»: вона більше підходить для аналітики, а не для повного immersive досвіду;
- вартість і ліцензування: як спеціалізований інструмент для досліджень або інститутів, може бути дорогим;
- може бути складним у налаштуванні для незвичайних або дуже нових конструкцій суден, якщо моделі відсутні в бібліотеці.

1.3 Аналіз можливостей застосування суднових навігаційних систем

Навігаційне обладнання суден – це багатокomпонентна електронна система водного транспортного засобу, встановлена з метою обчислення оптимального маршруту руху.

За точність обчислення і виконання навігаційних розрахунків відповідальний судновий навігаційний комплекс, що складається з: суднових технічних засобів навігації та суднового обладнання для задач керування маневрами судна.

Обсяг даних, якими оперує суднові навігаційні системи (СНС) неухильно збільшується. При розгляді певних питань всю інформацію, з якою оперують СНС, умовно ділять на два основних види: що належить до зображення земної поверхні (картографічну) і відноситься до навігації. В останні роки інформаційні ресурси СНС поповнюються головним чином відносяться до навігації даними. Тут можна назвати електронні варіанти настанов для плавання, посібників з радіотехнічних засобів, бази даних про порти, кліматичні бази, тощо [8], [19].

Сучасні СНС представляють вахтовому помічнику:

- дані про власне судно (поточне місце, кінематичні параметри, минулий шлях, запланований маршрут і ряд інших елементів);
- радіолокаційне зображення і кінематичні параметри цілей;
- дані транспондера про інші судна [20];
- відомості про навігаційні огорожі, оптичні та радіотехнічні навігаційні засоби;
- настанови для плавання;
- елементи сервісу берегових систем керування рухом;
- гідрометеорологічні відомості (багаторічні дані, відомості про поточну погоду, дані про льодову обстановку, прогностичну інформацію).

Використання в СНС для прокладання шляху судна векторних карт з автоматичною сигналізацією про небезпеки створило цим системам репутацію коштів, покликаних унеможливити посадку судів на мілину (antigrounding tool).

В даний час проводиться велика робота для створення СНС можливості в реальному часі отримувати гідрометеорологічну інформацію і оперувати нею для вирішення завдань судноводіння. В результаті, СНС стає ефективним засобом для забезпечення конструктивної безпеки судна і збереження вантажу в складних погодних умовах.

Простежуючи зміни, що відбуваються і з огляду на їх тенденцію, можна помітити, що СНС надають штурманам все більш інформаційну допомогу в керуванні судном. В результаті, в навігаційно-інформаційних системах все чіткіше проявляються риси засобів інформаційної підтримки рішень вахтового помічника. Для такої СНС картографічна інформація – лише невелика (хоча і дуже важлива) частину даних, що використовуються системою при підготовці рішень, а також фон, на якому представляються вихідні дані СНС судноводій.

СНС як повноцінна система інформаційної підтримки прийняття рішень повинна [9]:

- забезпечувати можливість хорошого знання обстановки;
- попереджати про елементи, явища, які потребують уваги вахтового помічника, а також про ситуації, що обумовлюють необхідність прийняття керуючих впливів;
- виробляти рекомендації для вирішення виникаючих проблем.

В недалекому майбутньому СНС буде здатна отримувати і оперувати практично всю необхідну для прийняття рішень з керування судном інформацію.

Це є передумовою для впровадження до складу СНС (з можливістю ефективного використання) експертної системи, що забезпечує: вироблення рішень виникаючих проблем судноводіння, рекомендацію їх вахтовому помічнику, автоматичну активацію в критичних ситуаціях процедур для забезпечення комплексної безпеки. ECDIS – це СНС, задовольняє спеціальним вимогам ІМО, МГО, МЕК і що є відповідно до SOLAS74 альтернативою паперових навігаційних карт [21]. Ці системи являють основний напрямок розвитку судових навігаційно-інформаційних систем [10].

Схема роботи СНС ECDIS наведена на рис. 1.1.

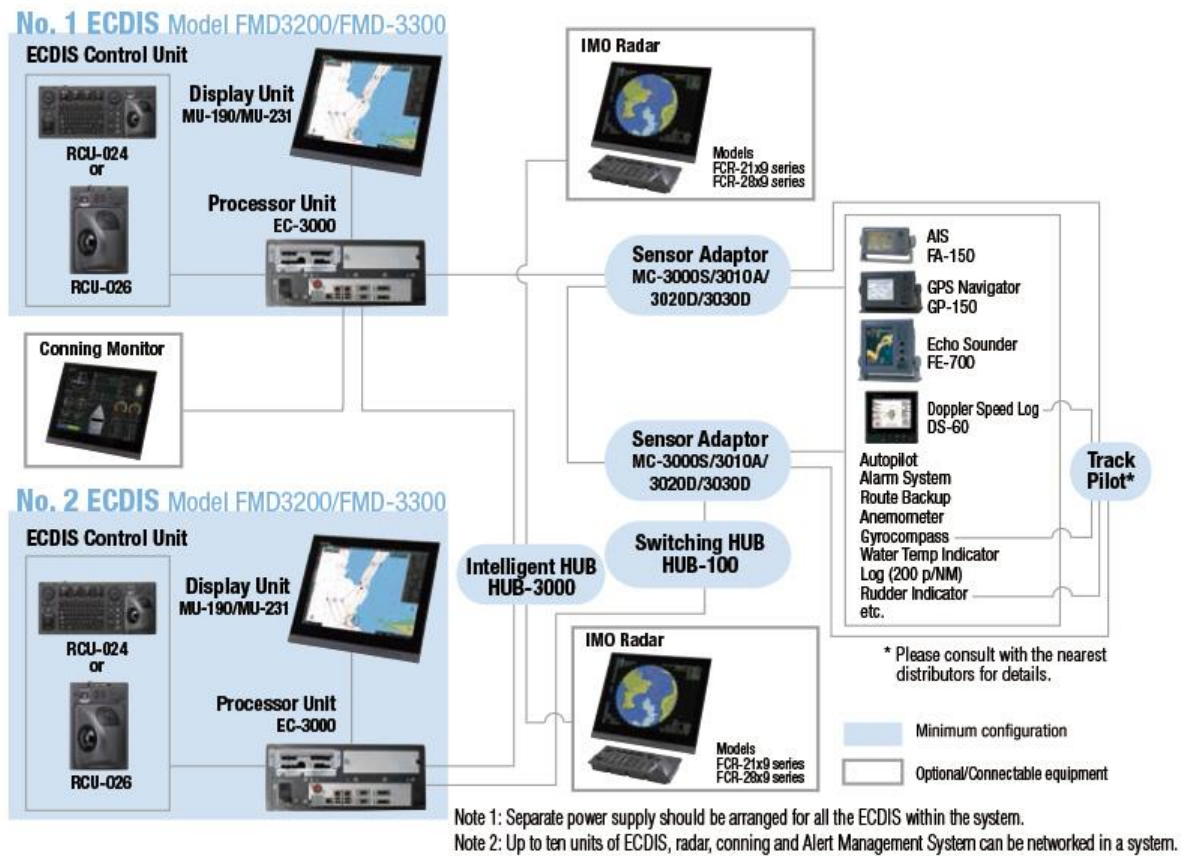


Рисунок 1.1 – Схема роботи СНС ECDIS

Застосовуючи СНС, вахтовий помічник звільняється від виконання багатьох рутинних операцій.

Його основними функціями стає спостереження за навколишнім оточенням, контроль СНС та інших засобів судноводіння, керування їх роботою для отримання необхідних обстановкою відомостей, оцінка наданої інформації та прийняття рішень з керування судном.

СНС підвищує результативність діяльності судноводія, забезпечує використання більшого обсягу і номенклатури даних, збільшує швидкість їх обробки, поліпшує точність і достовірність результатів, спрощує оцінку ситуацій, підвищує безпеку мореплавання і призводить до зростання фінансових показників роботи судна [12].

1.4 Аналіз особливостей виконання маневрування судном

Формалізовані моделі сталого руху судна по осі Y навколо вертикальної осі Z можуть бути записані [5], [22]:

$$R_y = P_y; M_R = P_y \cdot X_p, \quad (1.1)$$

де R_y – рівнодіюча поперечних гідродинамічних сил на корпусі судна; P_y – рівнодіюча поперечних сил від засобів керування; M_R – момент рівнодіючої поперечних гідродинамічних сил щодо вертикальної осі Z ; X_p – плече рівнодіючої поперечної сили від засобів керування.

Зазвичай плече висловлюють в відносних безрозмірних одиницях за формулою $\bar{X}_p = X_p / L_\perp$, де $L_\perp$ – довжина судна між перпендикулярами. При початкових умовах, коли судно на рівному кілі, і вважаючи його симетричним щодо мідель-шпангоута, диференціальне рівняння для елементарної гідродинамічної сили dR_y запишемо у вигляді:

$$dR_y = C_r \cdot \frac{\rho}{2} dV^2 \cdot dx \quad (1.2)$$

де C_r – безрозмірний гідродинамічний коефіцієнт опору; V – лінійна швидкість елемента; ρ – щільність води.

Лінійна швидкість будь-якої елементарної ділянки корпусу визначаються кутовою швидкістю і відстанню від полюса повороту, і розраховують за формулою:

$$V = (\bar{X}_{ПП} - \bar{X}) \cdot \omega \cdot L_\perp, \quad (1.3)$$

де ω – кутова швидкість обертання.

Для випадку, коли ПП поза корпусом $\bar{X}_{ПП} > 1/2$ (1.1) може бути записано:

$$R_y = C_r \cdot \frac{\rho}{2} dL^3 \cdot \omega^2 \int_{-1/2}^{1/2} (\bar{X}_{ПП} - \bar{X})^2 d\bar{x}. \quad (1.4)$$

Для випадку, коли ПП в межах корпусу рівняння (1.1) може бути записано:

$$R_y = C_r \cdot \frac{\rho}{2} dL^3 \cdot \omega^2 \left[\int_{-1/2}^{\bar{X}_{ПП}} (\bar{X}_{ПП} - \bar{X})^2 d\bar{x} - \int_{\bar{X}_{ПП}}^{1/2} (\bar{X}_{ПП} - \bar{X})^2 d\bar{x} \right]. \quad (1.5)$$

Після рішення диференціальних рівнянь (1.4) і (1.5) між точкою докладання бічної сили X_p і положенням ПП на лінії ДП $X_{П}$ отримана залежність:

– при розташуванні полюса повороту за межами корпусу, $\bar{X}_{\Pi} >$

$$\bar{X}_P = (-X_{\Pi\Pi}/6)/(\bar{X}_{\Pi\Pi}^2 + 1/12). \quad (1.6)$$

– при розташуванні полюса повороту в межах корпусу, $\bar{X}_{\Pi} < 0/5$

$$\bar{X}_P = (\bar{X}_{\Pi\Pi}^4/6 - \bar{X}_{\Pi\Pi}^2/4 - 1/32)/(2 \cdot \bar{X}_{\Pi\Pi}^3/3 + \bar{X}_{\Pi\Pi}/2). \quad (1.7)$$

Для переходу від положення точки прикладання сили до абсцис ПП і назад автори пропонують будувати графік по залежностям (1.6) і (1.7) і за ним визначати параметри. Недоліком цього способу є його трудомісткість і неможливість виконувати розрахунки в процесі маневрування при дії декількох сил.

Існує алгоритм і розрахункова схема формалізованої моделі визначення рівнодіючої декількох внутрішніх сил і зовнішніх збурень [6].

Для визначення рівнодіючої поперечних сил виконаємо розрахунок суми за такими залежностями:

$$\sum_{i=1}^n P_{\text{рез}} = P_1 \cdot \sin \alpha_1 + P_2 \cdot \sin \alpha_2 + \dots + P_i \cdot \sin \alpha_i + \dots P_n \cdot \sin \alpha_n, \quad (1.8)$$

де $P_{\text{рез}}$ – рівнодіюча поперечних сил; $P_1, P_2, \dots, P_i, \dots, P_n$ – сили, прикладені від буксирів, поперечна сила гвинта, сила від керма та підрулює пристрою з відповідним знаком, «+» в сторону правого борта, «-» в сторону лівого; $\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_n$ – кут між діаметральної площиною і напрямком дії сили.

Суму моментів сил відносно центра ваги G визначимо:

$$\sum_{i=1}^n M = P_1 \cdot \sin \alpha_1 \cdot \bar{\ell}_1 + P_2 \cdot \sin \alpha_2 \cdot \bar{\ell}_2 + \dots + P_i \cdot \sin \alpha_i \cdot \bar{\ell}_i + \dots P_n \cdot \sin \alpha_n \cdot \bar{\ell}_n, \quad (1.9)$$

де $\bar{\ell}_1, \bar{\ell}_2, \dots, \bar{\ell}_i, \dots, \bar{\ell}_n$ – безрозмірне плече кожної сили, $\bar{\ell} = \ell/L_{\perp}$;

Крапку докладання рівнодіюча поперечної сили визначимо за такою залежністю:

$$\bar{X}_P = \sum_{i=1}^n M / P_{\text{рез}}. \quad (1.10)$$

На основі залежностей (1.6), (1.7) і (1.10) будуємо функцію виду $\bar{X}_{\Pi\Pi} = f(\bar{X}_P)$, табулюємо залежності (1.6) і (1.7) і апроксимуємо їх поліномом 3-го порядку. В результаті отримаємо функцію з достовірністю $R^2 = 0,999$:

$$\bar{X}_{III} = \begin{cases} 18,11\bar{X}_P^3 + 27,44\bar{X}_P^2 + 14,05\bar{X}_P + 2,576; & \text{для } -0,5 \leq \bar{X}_P \leq -0,15. \\ -0,149/(\bar{X}_P + 0,005); & \text{для } -0,15 \leq \bar{X}_P \leq 0. \\ -0,149/(\bar{X}_P - 0,005); & \text{для } 0 \leq \bar{X}_P < 0,15. \\ 18,11\bar{X}_P^3 - 27,44\bar{X}_P^2 + 14,05\bar{X}_P - 2,576; & \text{для } 0,15 \leq \bar{X}_P \leq 0,5. \end{cases} \quad (1.11)$$

Наведена залежність (1.11) піддається операціями диференціювання і інтегрування і дозволяє отримати значення абсциси ПП.

Недоліком даного способу є необхідність вводити нові дані при зміні зовнішніх і внутрішніх сил. Він може бути використаний тільки в напівавтоматичному режимі.

В роботі [7] розглянуто пристрій, в якому для усунення зазначеного недоліку, використовується спосіб визначення абсциси ПП за значенням тангенціальних швидкостей точок на носовому і кормовому перпендикулярах ДП.

Принцип дії приладу заснований на розрахунку положення ПП за такою залежністю:

$$X_{III} = \frac{-V_H}{V_K - V_H} (X_B - X_A) + X_A, \quad (1.12)$$

де X_{III} – абсциса ПП з відповідним знаком «+» в сторону носа і «-» в бік корми щодо центра ваги; V_H – тангенціальна швидкість точки ДП на носовому перпендикулярі V_K – тангенціальна швидкість точки ДП на кормовому перпендикулярі; $X_B = -0,5 \cdot L_{\perp}$ – координата точки ДП на кормовому перпендикулярі; $X_A = 0,5 \cdot L_{\perp}$ – координата точки ДП на носовому перпендикулярі; $L_{\perp}$ – довжина судна між перпендикулярами.

Застосування даного способу і пристрою дозволить судноводію оперативно керувати рухом судна, шляхом контролю точки ПП, координати якої визначаються без необхідності вводити дані про режими роботи суднових і зовнішніх керуючих сил, що значно прискорює прийняття рішення по керуванню процесом маневрування і зменшує вірогідність виникнення непередбачених ситуацій.

Висновки до розділу 1

У першому розділі проаналізовано особливості технічного керування судном, ризики та фактори, що впливають на безпеку плавання. Визначено ключову роль сучасних навігаційних систем (GNSS, AIS, INS) у підвищенні точності та ефективності керування. Показано важливість врахування полюса повороту судна як критичного параметра маневрування та обґрунтовано необхідність його автоматизованого визначення. Зроблено висновок про доцільність подальшої автоматизації й визначення специфікацій вимог до ПЗ, що дозволить оптимізувати керування та знизити ризик аварійних ситуацій.

На базі отриманих результатів стає можливим подальше дослідження математично-алгоритмічної частини розробки, що буде розглянуто у наступному розділі роботи.

2 ФОРМАЛІЗАЦІЯ ПРОЦЕСІВ ОЦІНКИ ПОКАЗНИКІВ МАНЕВРУВАННЯ СУДНА

2.1 Методологія визначення параметрів маневрування судна

Морські операції з швартування та відшвартування суден у порту й на відкритих рейдах супроводжуються впливом параметричних змін. Для збереження високих експлуатаційних характеристик системи в таких складних умовах необхідно використовувати підхід, заснований на навігаційних пристроїв підтримки прийняття рішення при маневруванні [23]. На відміну від використовуваного оглядово-порівняльного способу керування рухом, який заснований на зоровому сприйнятті оператором пеленг і відстаней до орієнтирів і прийняття по їх значенням рішень з керування судном. Створення систем підтримки прийняття рішення вимагає подальшого дослідження процесу маневрування і розвитку сучасних навігаційних пристроїв. Серед таких параметрів є інформація про абсцису полюса повороту (ПП). Методом вирішення такого завдання є синтез структури навігаційної системи визначення положення ПП

Для опису поведінки судна, як об'єкта керування, зазвичай використовують дві системи координат (рис. 2.1):

O_1 – нерухома система координат;

OXY – система координат жорстко пов'язана з судном так, що її початок збігається з центром його тяжкості, а вісь OX направлена в сторону носа, а вісь OY – в сторону правого борта.

Площини O_1 та OXY збігаються з поверхнею води (площиною ватерлінії). Положення судна визначається координатами його центра ваги і кутом курсу, а кінематичними параметрами руху є лінійна швидкість V , кутова швидкість обертання ω і кут дрейфу від повороту.

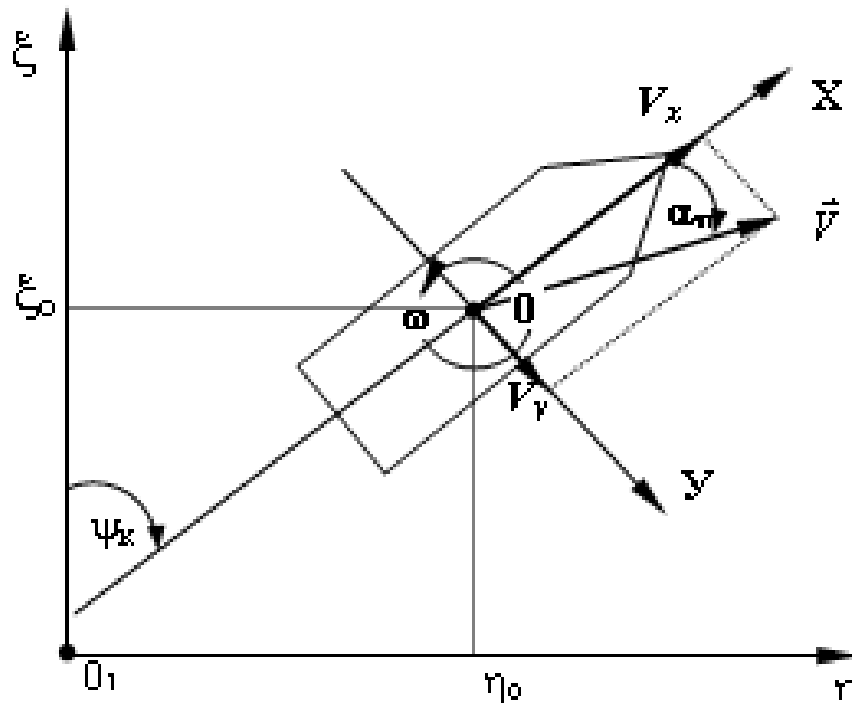


Рисунок 2.1 – Системи координат і параметри положення судна

Вважаючи, що центр тяжіння знаходиться в площині мідель-шпангоута, можна отримати наступні диференціальні рівняння при початкових умовах $t=0$, :

$$\xi = \xi_0 + \int_0^t V \cdot \cos(\psi_k - \alpha_{\pi}) dt; \quad (2.1)$$

$$\eta = \eta_0 + \int_0^t V \cdot \sin(\psi_k - \alpha_{\pi}) dt; \quad (2.2)$$

$$\psi_k = \psi_{k_0} + \int_0^t \omega \cdot dt., \quad (2.3)$$

де ξ_0, η_0 – координати центра ваги в початковий момент часу $t=0$;

ψ_{k_0} – початковий курс судна.

Для визначення положення ПП при маневруванні судна необхідно використовувати високоточні методи оперативного визначення тангенціальних швидкостей країв судна щодо води, з урахуванням зовнішніх збурень, для визначення абсциси положення ПП. При цьому необхідно враховувати наступні закономірності, які встановлені теоретично і вимагають підтвердження експериментально [25]. Точка прикладання поперечної сили, що викликає обертальний рух, лежить на діаметральній площині по різні боки від початку координат щодо ПП.

Протягом не впливає на положення ПП, оскільки воно викликає переміщення щодо ґрунту, впливаючи на стан судна в просторі. При наявності течії і розрахунку абсциси ПП по тангенціальним швидкостям країв вектор течії необхідно виключати з розрахунків.

Якщо в процесі розрахунку напівавтоматичним способом є вітер, то силу від нього необхідно враховувати.

Надводна частина корпусу судна виступає як крило гранично малого подовження, з двома властивостями (рис. 2.2): точка докладання аеродинамічної сили зміщується до передньої кромці крила; напрямок зазначеної сили розташовується по нормалі до поверхні.

Під дією удаваного вітру u_k з курсовим кутом q_w відбувається бічне переміщення зі швидкістю $V_\square$ за рахунок аеродинамічної сили R_{ay} з плечем ℓ_{ay} [26]. При цьому виникає кут вітрового дрейфу α і судно переміщається зі швидкістю $\vec{V} = \vec{V}_n + \vec{V}_\alpha$. При русі з кутом дрейфу на корпусі виникає гідродинамічна сила R_{gy} з плечем ℓ_g .

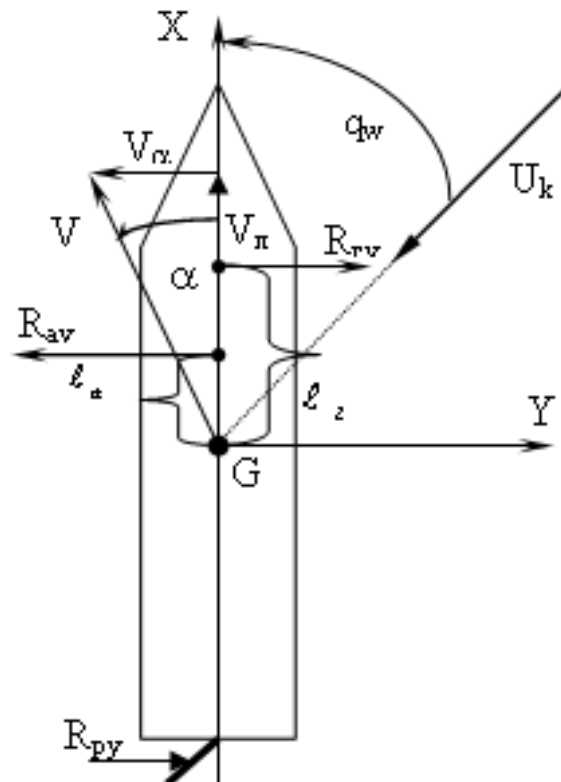


Рисунок 2.2 – Дія удаваного вітру на судно

Відносне зміщення аеродинамічної сили від мідель-шпангоута (плече аеродинамічної сили в довжинах корпусу) можна визначити за формулою Федяєвського:

$$\bar{\ell}_{ay} = 0,5 + \frac{\ell_{\text{цп}}}{L} - \frac{q_w^0}{360^0} \quad (2.4)$$

де L – довжина судна між перпендикулярами, м; $\ell_{\text{цп}}$ – відстань центру парусності (ЦП) від мідель-шпангоута, м; q_w^0 – курсовий кут уявного вітру.

У формулі (2.4) плече аеродинамічної сили має знак «+», якщо ЦП зміщений в ніс від міделя і знак «-» при його зміщенні в бік корми. Слід зауважити, що розташування ЦП по довжині судна залежить від розмірів і розташування надбудов, диференту судна і його опади. Ці дані зазвичай розраховують при будівництві судна, і вони є в судових документах.

Таким чином, на судно діє аеродинамічна сила, момент якої $M_{ay} = R_{ay} \cdot \ell_{ay}$, прагне розвернути судно щодо вертикальної осі лагом до вітру.

Для дослідження запропонованих способів розрахунку абсциси ПП і обліку течії при визначенні положення ПП експерименти проводилися на високо-енергетичному буксирі – постачальника «RAWABI 15» в лютому 2014 року в Перській затоці при відсутності вітру. Воно обладнане сучасним високоточним навігаційним пристроєм, включаючи чотирьох променевої доплерівський лаг. Лаг FURUNO DS-60 має таку точність: швидкість відносно ґрунту (при швидкості менш 1 вузла) ± 0.01 м / с; швидкість відносно ґрунту (при швидкості понад 1 вузла) $\pm 1\%$ або $\pm 0,1$ вузла (в залежності, що більше); швидкість щодо води $\pm 1\%$ або $\pm 0,1$ вузла (в залежності, що більше); швидкість бічного переміщення $\pm 1\%$ або $\pm 0,04$ вузла (в залежності, що більше). Діапазон глибин застосування – від 0 до 200 м.

Доплерівський лаг за своїм фізичним змістом може показувати швидкість центра ваги судна відносно ґрунту. Крім того він може обчислювати і показувати швидкість і напрямок течії щодо ґрунту, а також тангенціальні швидкості країв. Положення ж ПП на судні визначається щодо води, і щоб визначити його

положення слід визначити відносні тангенціальні швидкості країв і по ним розрахувати абсцису ПП.

Для обліку течії використовуємо схему руху судна для експерименту, дані про який представлені на рис. 2.3.

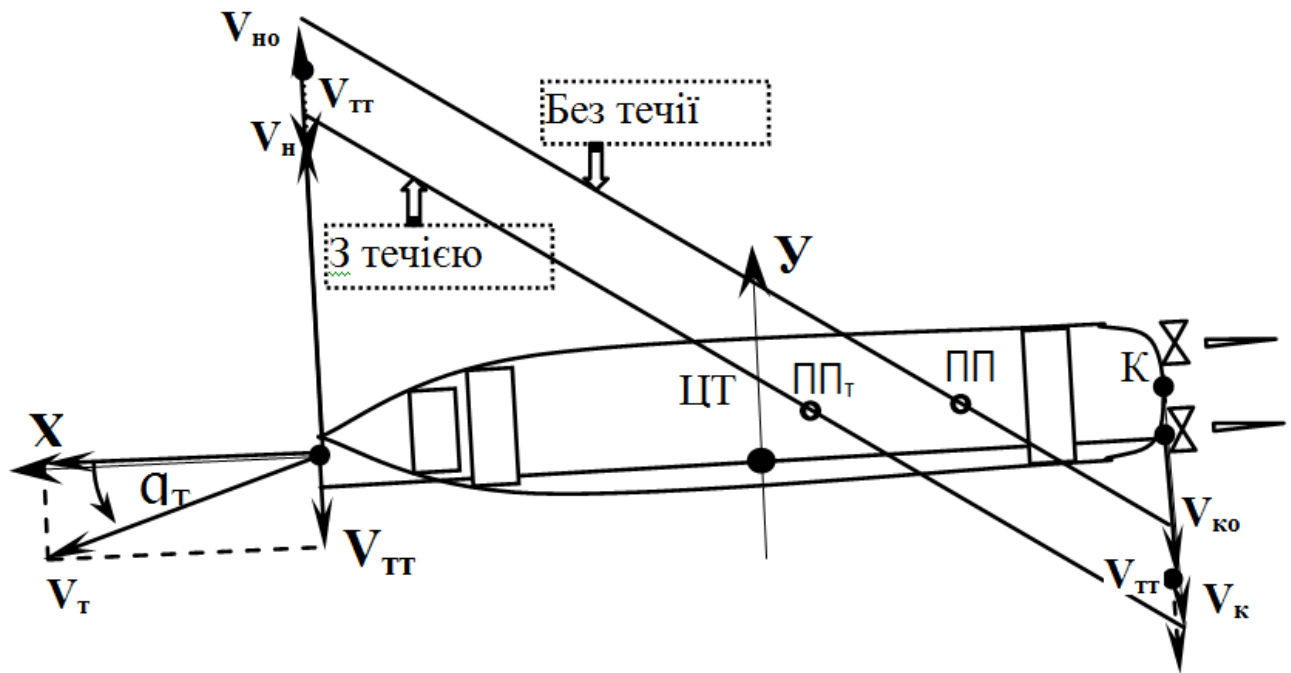


Рисунок 2.3 – Розрахункова схема визначення ПП на течії

Судно має курс 267° , бічну швидкість носа $V_n = 0,66$ вузла, бічну швидкість корми $V_k = 0,48$ вузла, протягом напрямком 243° швидкістю $V_t = 0,5$ вузла.

Для розрахунку тангенціальних швидкостей щодо води зробимо визначення курсового кута течії q_t за формулою:

$$q_t = K_t - \text{ИК}. \quad (2.5)$$

Тангенціальна складова течії буде дорівнює:

$$V_{tt} = V_t \cdot \sin q_t \quad (2.6)$$

Виробляючи геометричне складання векторів тангенціальних швидкостей носа і корми, з урахуванням їх знаків і знаку вектора течії отримаємо їх відносне значення $\vec{V}_{но}, \vec{V}_{ко}$:

$$\vec{V}_{но} = \vec{V}_n - \vec{V}_{tt} \quad (2.7)$$

$$\vec{V}_{ко} = \vec{V}_k - \vec{V}_{tt} \quad (2.8)$$

З'єднуємо отримані вектори $V_{\text{но}}$ та $V_{\text{ко}}$ відрізком прямої лінії. У точці перетину цього відрізка з діаметральної площиною отримаємо положення полюса повороту ПП без урахування течії.

Таким чином, після врахування швидкості течії залишається 2 варіанти: коли швидкості носа і корми спрямовані в різні боки і коли вони спрямовані в одну сторону. Для спрощення розрахунків модифікуємо формулу для розрахунку абсциси ПП по тангенціальним швидкостям носа і корми з урахуванням перебігу $X_{\text{ППТ}}$

$$X_{\text{ППТ}} = \frac{V_{\text{н}} \cdot L_{\perp}}{V_{\text{к}} - V_{\text{н}}} + X_{\text{А}}, \quad (2.9)$$

де $X_{\text{ППТ}}$ – абсциса полюса повороту з відповідним знаком «+» в сторону носа і «-» в бік корми щодо центра ваги; $L_{\perp}$ – довжина судна між перпендикулярами (м).

Для розрахунку абсциси полюса повороту по тангенціальним швидкостям носа $V_{\text{но}}$ і корми $V_{\text{ко}}$ без урахування течії $X_{\text{ППо}}$ відносно води

$$X_{\text{ПП}} = \frac{V_{\text{но}} \cdot L_{\perp}}{V_{\text{ко}} - V_{\text{но}}} + X_{\text{А}}. \quad (2.10)$$

Розрахувавши абсцису полюса повороту, ми можемо розрахувати і абсцису точки прикладання рівнодіючої бічних сил $\bar{X}_{\text{Р}}$.

2.2 Алгоритми автоматизованого визначення полюса повороту

Вихідними даними для виконання розрахунків будуть наступні дані, що надходять в пристрій розрахунку від доплерівського лага через інтерфейс: $V_{\text{н}}$ – тангенціальна швидкість носа щодо ґрунту, узл.; $V_{\text{к}}$ – тангенціальна швидкість корми щодо ґрунту, узл.; $V_{\text{т}}$ – швидкість течії, узл.; $q^0_{\text{т}}$ – курсовий кут течії, градуси; $K^0_{\text{т}}$ – напрямок течії, градуси; ИК – справжній курс, градуси.

Блок-схема навігаційного пристрою автоматичного розрахунку абсциси ПП $\bar{X}_{\text{ПП}}$ і точки докладання рівнодіюча всіх сил $\bar{X}_{\text{Р}}$, прикладених до судна, приведена на рис. 2.4.

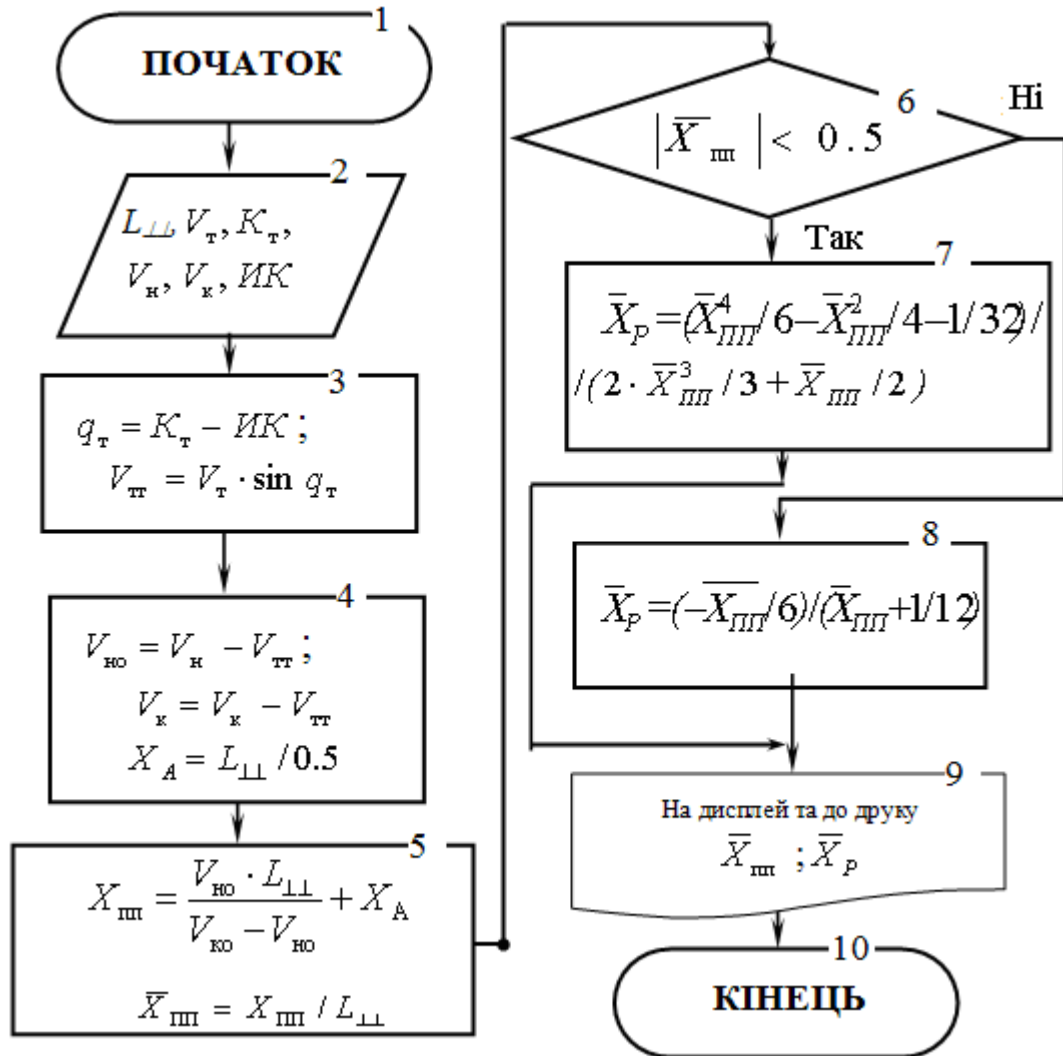


Рисунок 2.4 – Блок-схема алгоритму автоматичного розрахунку абсциси ПП

На відміну від визначення положення полюса повороту по тангенціальним швидкостям пропонується спосіб розрахунку напівавтоматичним пристроєм по рівнодіючої всіх сил, що діють на судно. Однак вказаний спосіб дає менш точні результати, через неточного визначення точки прикладання зовнішніх сил, їх напрямки та величини і вимагають більше часу для обробки.

Крім того використовуючи доплерівський лаг, можна також визначати швидкість і напрямок течії, що дозволяє підвищити точність письмового числення [24].

Вихідні дані для розрахунку діляться на дві групи даних – суднові з постійною точкою докладання і змінною величиною і зовнішні, зі змінними точкою докладання і величиною зовнішніх сил.

Якщо на судні встановлено два підрулюючих пристрої то число випадків, для яких необхідно розраховувати абсцису ПП буде сім: P_{py} , $P_{py} + P_{np}$, $P_{py} + P_{kp}$, $P_{py} + P_{np} + P_{kp}$, P_{np} , P_{kp} , $P_{np} + P_{kp}$. При наявності тільки носового підрулює пристрою число використовуваних комбінацій буде тільки три. Блок-схема розрахунку результуючих сил і моментів приведена на рис. 2.5.

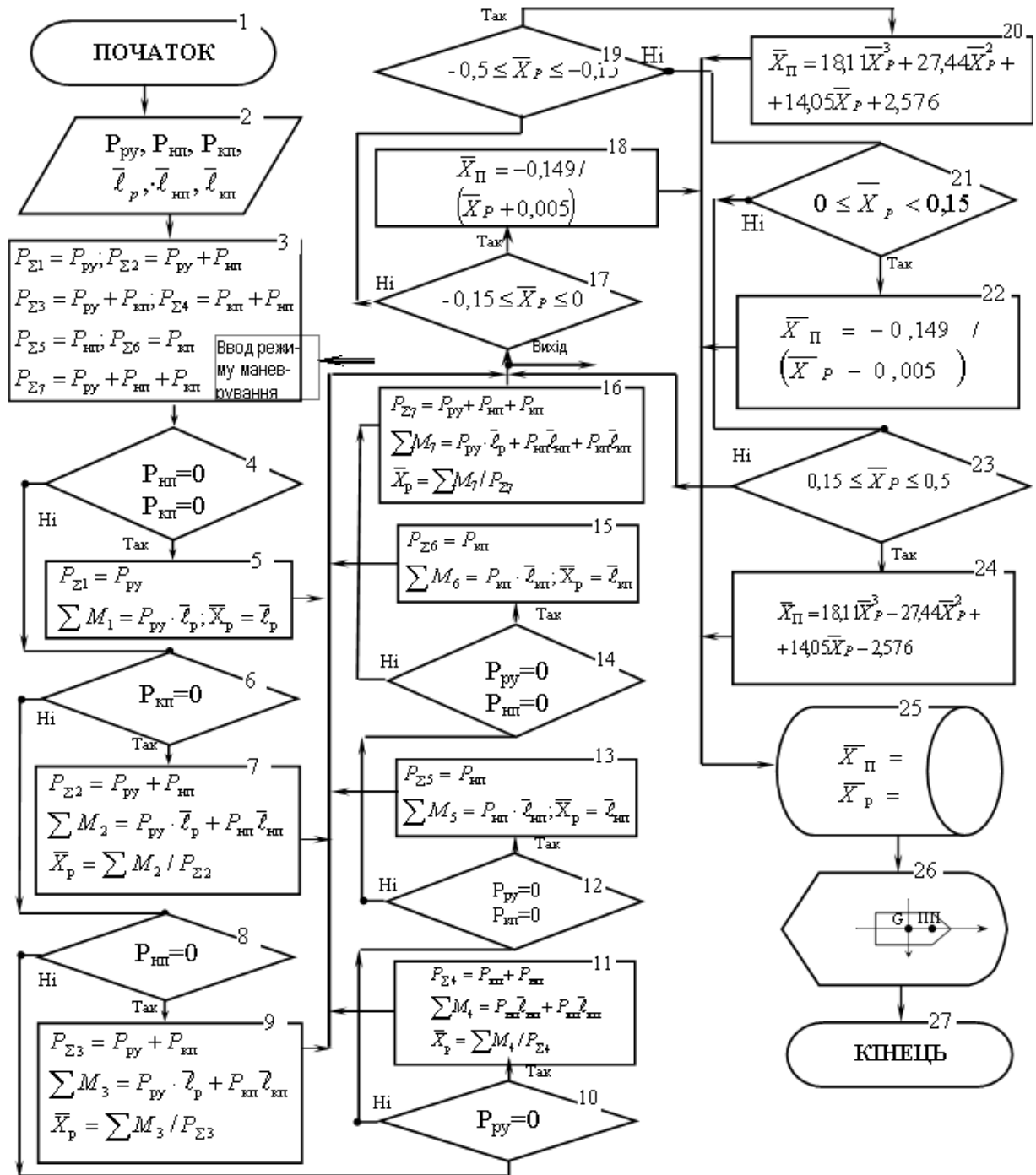


Рисунок 2.5 – Блок-схема розрахунку положення ПП без зовнішніх сил

Введення режиму маневрування проводиться в блоці 3, відповідно до переліку використовуваних поєднань, наведених в табл. 2.1.

Таблиця 2.1 – Перелік режимів маневрування

№ з/п	Використані поєднання режимів	Умовне позначення
1	Бічна сила від керма	$P_{\Sigma 1}$
2	Бічна сила від керма і носового підрулювального пристрою	$P_{\Sigma 2}$
3	Бічна сила від керма і кормового підрулювального пристрою	$P_{\Sigma 3}$
4	Бічна сила від носового і кормового підрулювального пристрою	$P_{\Sigma 4}$
5	Бічна сила від носового підрулювального пристрою	$P_{\Sigma 5}$
6	Бічна сила від кормового підрулювального пристрою	$P_{\Sigma 6}$
7	Сумарна бічна сила від керма, носового і кормового підрулювального пристрою	$P_{\Sigma 7}$

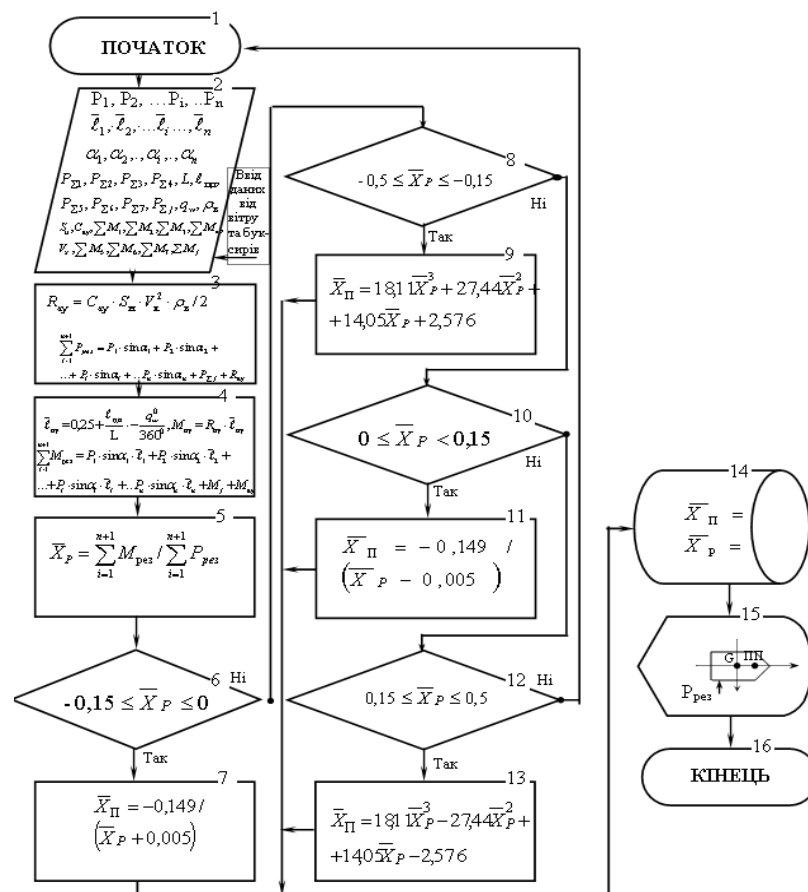
Для розрахунку положення ПП при впливі зовнішніх сил і результуючої всіх керуючих впливів на корпус судна в блок схемі передбачено один вхід в блоці 3 і вихід в навігаційний пристрій з входу блоку 17 даних про сумарну бічну силу $P_{\Sigma 1}$ і її моменту ΣM_1 .

При цьому значення сум використовуваних поєднань $P_{\Sigma 1}, P_{\Sigma 2}, P_{\Sigma 3}, P_{\Sigma 4}, P_{\Sigma 5}, P_{\Sigma 6}, P_{\Sigma 7}$ сил і моментів $\Sigma M_1, \Sigma M_2, \Sigma M_3, \Sigma M_4, \Sigma M_5, \Sigma M_6, \Sigma M_7$, будуть вихідними даними для розрахунку плеча рівнодіючої внутрішніх і зовнішніх сил і по ньому абсциси ПП і точки прикладання результуючої сили.

Якщо судно маневрує в умовах вітру, то його вплив буде однаковим при будь-яких режимах руху і використовуваному буксирному забезпеченні. Таким

чином, напівавтоматичний навігаційний пристрій буде складатися з двох блоків: визначення абсциси ПП і результуючої сили від керуючих впливів; визначення абсциси ПП і результуючої всіх сил, що діють на судно, включаючи керуючі сили, від вітру і буксирного забезпечення. Вихідними даними для розрахунків сили від вітру будуть постійні: довжина судна між перпендикулярами L ; площа проєкції надводної частини корпусу судна S_n ; абсциса центру парусності надводної частини корпусу $\ell_{цп}$; масова щільність повітря ρ_v . До змінних даних належать швидкість і курсової кут уявного вітру q_w , безрозмірний аеродинамічний коефіцієнт C_{ay} , який за даними натурних і модельних випробувань зазвичай приймають рівним $C_{ay} = 0.8 \cdot \sin q_w \div 1.3 \cdot \sin q_w$.

Дія буксира описують величиною упору гвинта P , точкою прикладання до судна ℓ і напрямом дії сили упору щодо ДП.



Для усунення зазначеного парадоксу пропонується судно розглядати у вигляді кола діаметром $L_{\max}$ з центром тяжіння на міделі. У цьому випадку виникає необхідність перерахунку координат приймальної супутникової антени на координати центру ваги (рис. 2.7) [27]. Необхідні відомості про $L_{\max}$ і положенні супутникової антени є в судових документах.

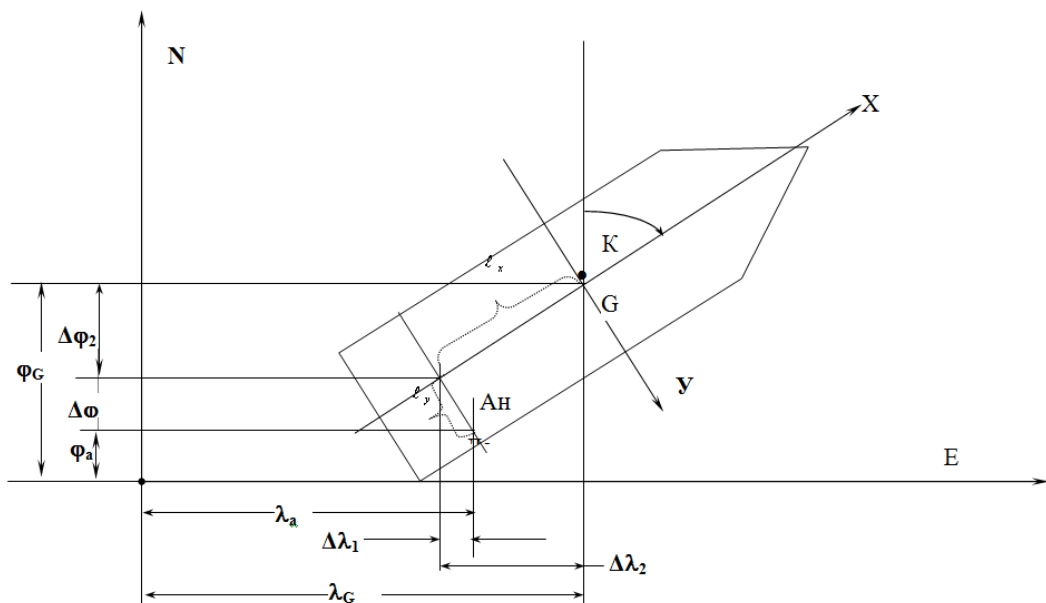


Рисунок 2.7 – Приведення обсервації до центру ваги судна

Геометричний аналіз алгоритму перерахунку і динаміки зміни курсу при маневруванні показав, що похибка перерахунку координат не залишається постійною. Вона змінюється від місця розташування супутникової антени по відношенню до центру тяжіння, географічних координат і курсу, яким слід судно. Координати центра ваги ϕ_G і λ_G розраховуються за певними супутниковою системою даними ϕ_a і λ_a при курсі K в межах $0^0 \leq K \leq 90^0$ за умови $\phi_a \geq 0, \lambda_a \geq$

0, за такими залежностями:

$$\phi_G = \phi_a + \Delta\phi_1 + \Delta\phi_2 = \phi_a + \Delta\phi_G, \lambda_G = \lambda_a - \Delta\lambda_1 + \Delta\lambda_2 = \lambda_a + \Delta\lambda_G,$$

де $\Delta\lambda_1$ – проєкція відстані точки прийому антени Ан від діаметральної площини ℓ_y на вісь λ , в милях; $\Delta\lambda_2$ – проєкція відстані точки розташування антени ℓ_x від центра ваги (ЦВ) по ДП на ось λ , в милях; $\Delta\phi_1$ – проєкція відстані точки прийому антени Ан від діаметральної площини ℓ_y на вісь ϕ , в милях; $\Delta\phi_2$ – проєкція відстані точки розташування антени ℓ_x від ЦВ по ДП на вісь ϕ , в милях; $\Delta\lambda_G$ та $\Delta\phi_G$ – поправки координат ЦВ.

У розгорнутому вигляді рівняння можна записати так

$$\Delta\lambda_1 = \ell_y \cdot \cos K; \Delta\lambda_2 = \ell_x \cdot \sin K \quad (2.11)$$

$$\Delta\phi_1 = \ell_y \cdot \sin K; \Delta\phi_2 = \ell_x \cdot \cos K \quad (2.12)$$

Тоді, після підстановки отримаємо

$$\lambda_G = \lambda_a - \ell_y \cdot \cos K + \ell_x \cdot \sin K \quad (2.13)$$

$$\phi_G = \phi_a + \ell_y \cdot \sin K + \ell_x \cdot \cos K \quad (2.14)$$

Положення супутникової антени в судовому формулярі описується проєкціями місця розташування антени ℓ_{ym} в метрах на судову вісь Е та ℓ_{xm} на вісь N. При цьому позитивне значення проєкції прийнято по осі У зі знаком плюс в сторону правого борта від центра ваги і по осі Х в бік носа. Можливі чотири випадки розташування антени по відношенню до центру тяжкості: в сторону носа з правого борту; в сторону носа з лівого борту; в бік корми з правого борту; в бік корми з лівого борту. З урахуванням географічних координат можливі 64 розрахункових схеми, детальна блок схема алгоритму яких, для випадку (рис. 2.7) розташування антени справа по кормі, за умови $\phi_a \geq 0, \lambda_a \geq 0$ наведена на рис. 2.8.

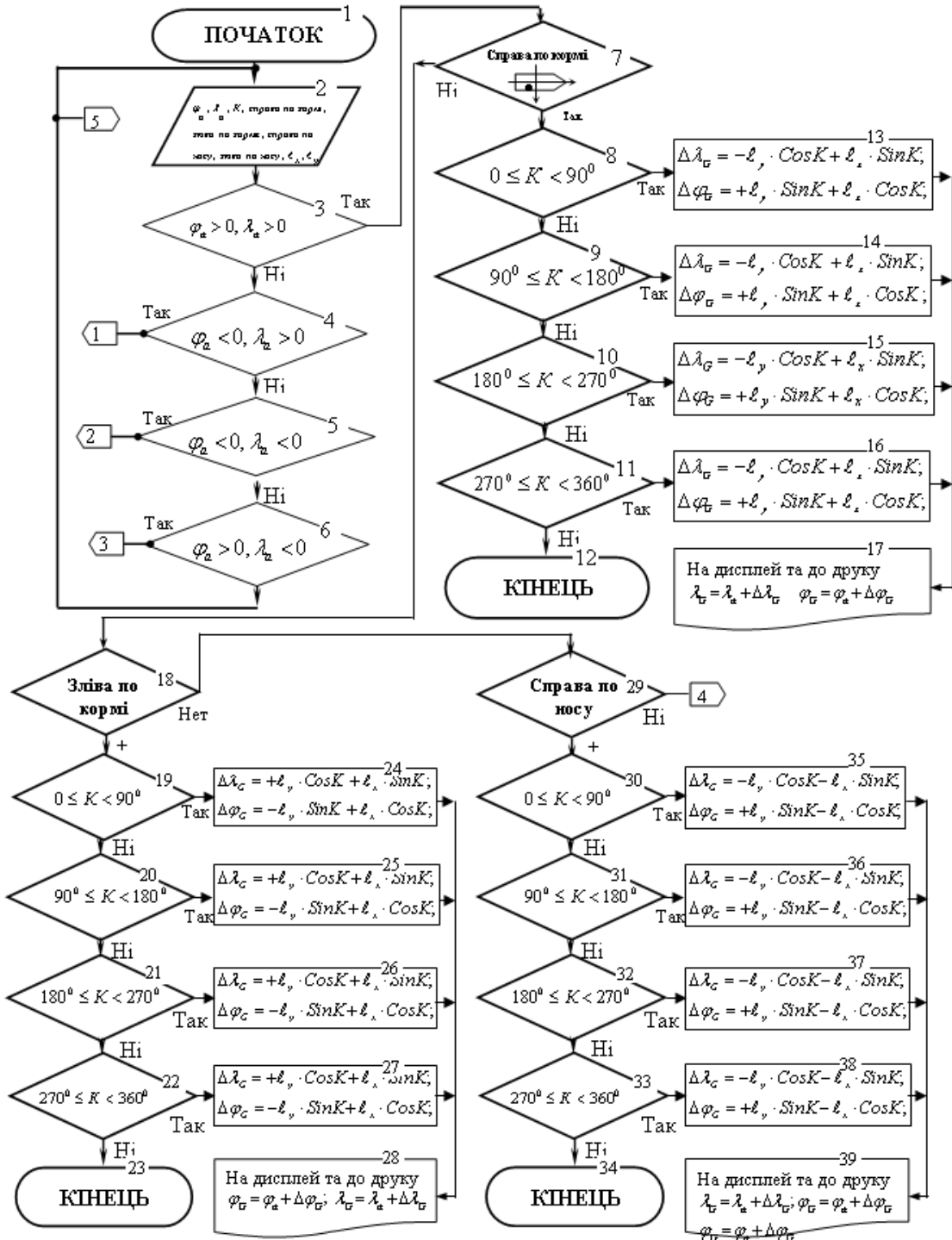
Вихідними даними для перерахунку координат будуть:

- 1) курс судна К в градусах;
- 2) географічна широта прийомної антени ϕ_a ;
- 3) географічна довгота прийомної антени λ_a ;
- 4) координата проєкції місця антени ℓ_y на судову вісь У, в милях. Якщо в

суднових документах наведені в м, то переводять в милі, $\ell_y = \ell_{ym}/1852$;

5) координата проєкції місця антени ℓ_x на суднову вісь X в милях.

Якщо в суднових документах наведені в м, то переводять в милі, $\ell_x = \ell_{xm}/1852$.



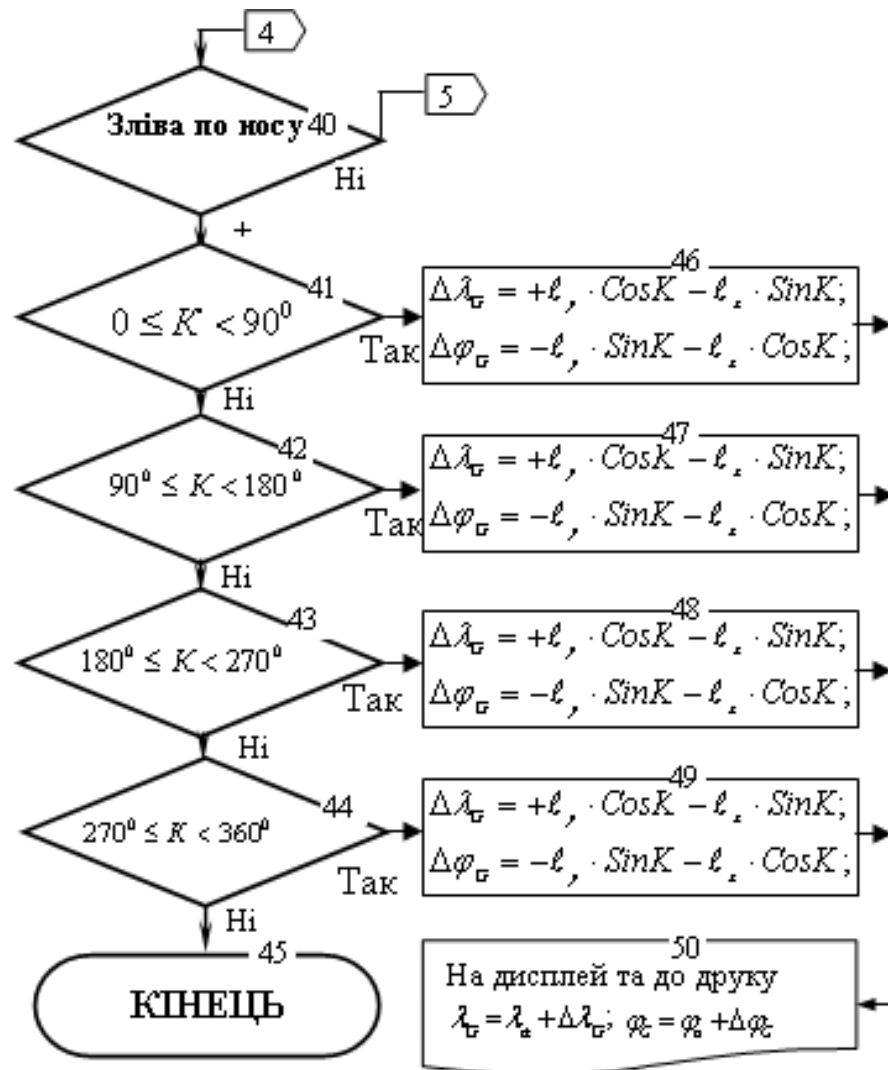


Рисунок 2.9 – Блок-схема алгоритму перерахунку координат антени на центр ваги (частина 2)

Розташування антени щодо центру G на міделі буде описуватися знаком проєкції: зліва по кормі буде $-\ell_x$ та $-\ell_y$; справа по кормі буде $-\ell_x$ та $+\ell_y$; зліва по носу буде $+\ell_x$ та $-\ell_y$; справа по носу буде $+\ell_x$ та $+\ell_y$.

З урахуванням відсутності необхідності деталізації для решти трьох чвертей Земної поверхні, розрахункова схема спроститься і набуде вигляду, як показано на рис. 2.10. Наведені розрахункові схеми на рис. 2.9 – 2.10 дозволяють провести розрахунок поправок для всіх випадків перебування судновий антени, місця перебування судна і курсу, яким воно буде. З урахуванням постійного розташування супутникової антени обсяг обчислювальної роботи зменшиться в

чотири рази до 16 варіантів для конкретного судна.

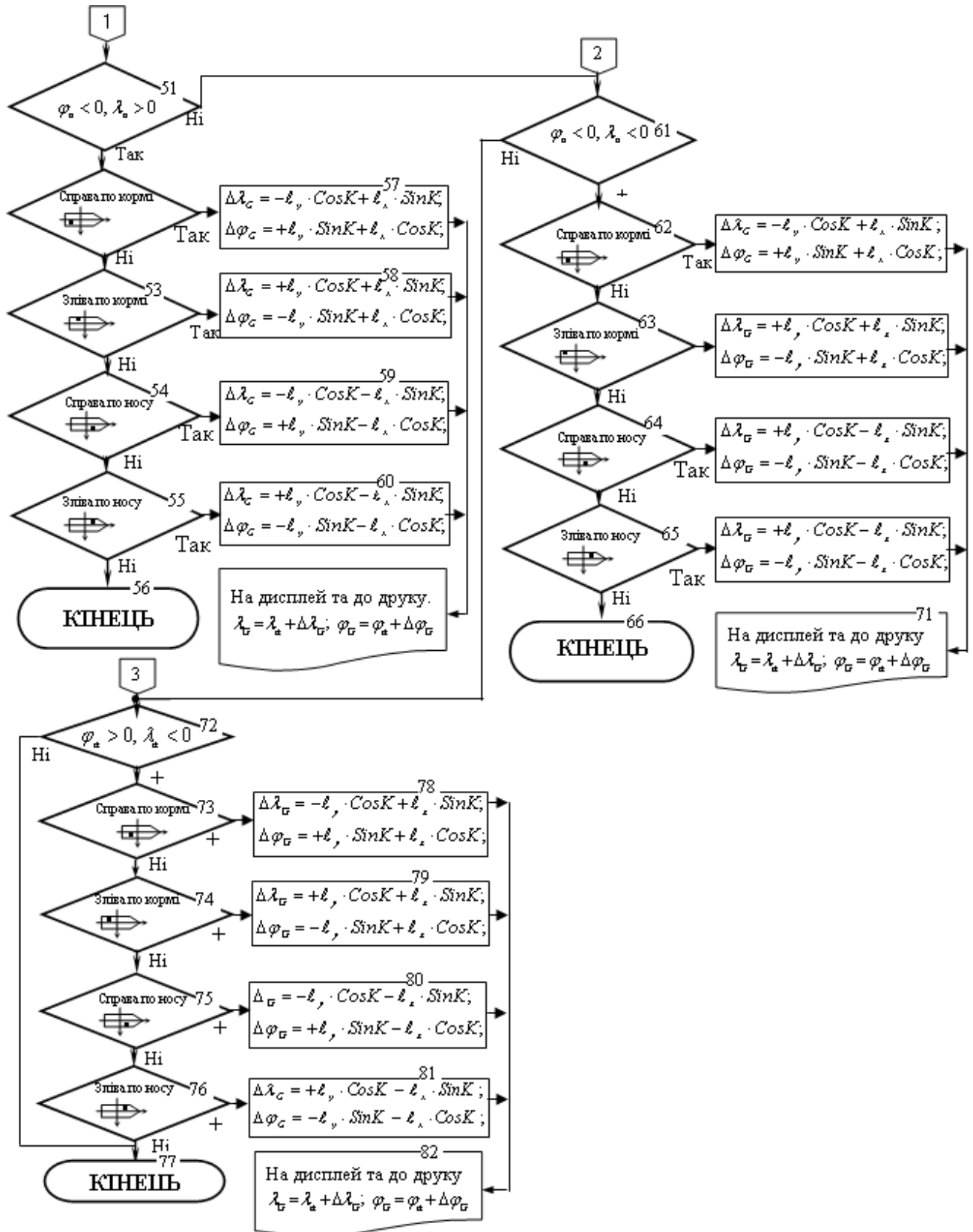


Рисунок 2.10 – Блок-схема укрупнення алгоритму перерахунку координат антени

Іншою особливістю поправок перерахунку на широту і довготу є той факт, що їх величина ніколи не буде перевищувати половини довжини судна.

З цієї причини складання поправки потрібно проводити з частиною координат в хвилинах, а розрахунок поправок виробляти до п'ятого знаку хвилини, а округлення до четвертого.

2.3 Специфікація вимог до програмного забезпечення

ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

Призначення системи, для якої розробляється програмне забезпечення

Система призначена для інформаційної підтримки судноводія під час виконання маневрування морським транспортом. Програмне забезпечення дозволяє обчислювати координати полюса повороту судна та візуалізувати його переміщення.

Погодження, що ухвалені в програмній документації

Було погоджено використання мови програмування Python, бібліотек PyQt5 та Matlab, а також технології QML для створення користувацького інтерфейсу.

Межі проєкту ПЗ

Кінцевий термін завершення розробки ПЗ – 15.10.2025 року.

ЗАГАЛЬНИЙ ОПИС

Сфера застосування

Система може бути використана у навчанні, тренажерах, практичному судноводінні, тестуванні навігаційних сценаріїв, а також для наукових досліджень.

Характеристики користувачів

Користувачами є судноводії, курсанти, оператори морських тренажерів, інженери та дослідники. Основна вимога – наявність ПК або ноутбука з ОС Windows.

Загальна структура і склад системи

Система складається з:

– модуль розрахунку полюса повороту;

- модуль перерахунку координат антени на центр ваги;
- модуль напівавтоматичного визначення ПП при впливі зовнішніх сил;
- модуль візуалізації траєкторії.

Загальні обмеження

Система функціонує у середовищі Windows.

ФУНКЦІЇ СИСТЕМИ

1) Функція автоматичного розрахунку полюса повороту

- опис: система визначає положення ПП на основі тангенціальних швидкостей носа і корми, з урахуванням зовнішніх впливів;
- вхідні дані: швидкість судна, курс, дані доплерівського лага, вітер, течія;
- вихідні дані: координати ПП, точка прикладання рівнодіючої сили.

2) Функція візуалізації маневрування

- опис: відображення руху судна та положення ПП на екрані у реальному часі;
- вхідні дані: результати розрахунку модулів;
- вихідні дані: графічне представлення траєкторії.

3) Функція перерахунку координат антени

- опис: переведення координат GNSS-антени у координати центру ваги судна.

ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

Джерела вхідної інформації

Введення інформації користувачем.

Збереження інформації

Збереження інформації у форматі XLSX.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Мінімальні технічні характеристики:

- ОС Windows 10;
- оперативна пам'ять – від 4 ГБ;
- процесор – від Intel i3 або аналог;

– дисковий простір – від 500 МБ.

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура

Клієнтський застосунок з модульною структурою.

Системне програмне забезпечення

Python 3.12, бібліотека PyQT5, Matlab.

Мова і технології розробки

Python, QML.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Інтуїтивний, з відображенням траєкторії та ПП.

Апаратний інтерфейс

Апаратним інтерфейсом є ОС Windows 10.

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

Система доступна для всіх користувачів з мінімальними знаннями ПК.

Супроводжуваність

Не потребує.

Переносимість

Сумісність з ОС Windows.

Продуктивність

Швидкість роботи залежить від обсягу вхідних даних.

Надійність

Забезпечується збереженням результатів у локальних файлах.

Безпека

Обмежується безпечним збереженням локальних даних.

ІНШІ ВИМОГИ

Усі вимоги сформовано.

Висновки до розділу 2

У другому розділі було здійснено формалізацію процесів оцінки показників маневрування судна та розроблено алгоритмічну основу для їх практичної реалізації. В результаті проведеного аналізу:

- сформульовано методологічні положення щодо опису руху судна в системах координат та визначення параметрів його положення і маневреності;
- обґрунтовано необхідність точного визначення абсциси полюса повороту (ПП) та точки прикладання рівнодіючої зовнішніх і внутрішніх сил як ключових характеристик процесу маневрування;
- наведено аналітичні залежності для розрахунку положення ПП з урахуванням течії, вітру та інших збурювальних факторів;
- розроблено блок-схеми алгоритмів автоматизованого та напівавтоматичного визначення положення ПП, що дозволяють підвищити точність оцінки маневрених можливостей судна;
- показано необхідність врахування похибок, пов'язаних із розташуванням приймальної супутникової антени відносно центра ваги судна, та запропоновано методику їх компенсації шляхом перерахунку координат.

Таким чином, у розділі обґрунтовано та формалізовано підхід до оцінки показників маневрування судна, який базується на використанні сучасних навігаційних засобів та алгоритмів обробки даних. Це створює передумови для побудови ефективних систем підтримки прийняття рішень при керуванні судном у складних умовах експлуатації.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ОЦІНКИ ПОКАЗНИКІВ НАВИГАЦІЙНОГО МАНЕВРУВАННЯ МОРСЬКОГО ТРАНСПОРТУ

3.1 Обґрунтування використаних засобів програмної розробки

3.1.1 Мова Python

Python – це сучасна об’єктно-орієнтована мова з підтримкою динамічної типізації, автоматичного процесу керування пам’яттю, високорівневих гнучких структур даних (словники, кортежі, списки). Всі об’єкти в Python підрозділяються на атомарні і посилальні. До перших відносяться `int`, `long`, `complex`.

При присвоєнні такого роду об’єктів відбувається копіювання їх значень, а в довідкових об’єктах здійснюється копіювання лише покажчика на об’єкт, тому обидві змінні після виконання операції присвоювання використовують однакове значення.

Python підтримує створення класів, зв’язку модулів, гнучку і зручну обробку виняткових ситуацій і багатопотокові методи обчислень.

Крім ООП, мова підтримує структурне, функціональне і аспектно-орієнтоване програмування [13].

Python дозволяє визначати тип змінної на етапі виконання програми. У зв’язку з цим замість присвоювання змінної певного значення більш коректним є використання фрази «зв’язування певного значення з конкретним ім’ям». У мові підтримуються такі вбудовані типи даних як: бінарний, строковий, Unicode, цілочисельний з довільно заданою ступінь точності, число з плаваючою комою, і ряд інших.

Із сучасних колекцій мова має підтримку таких інтегрованих структур як: список, словник, кортеж (модифікується список), безліч і ін. Слід зазначити, що всі значення, також як і в Ruby, є об’єктами, причому це характерно і для функцій, модулів, методів і класів.

Основною реалізацією мови є інтерпретатор CPython, який працює на значній кількості сучасних операційних систем. Він поширюється під вільною ліцензією,

що забезпечує можливість його використання без обмежень у будь-яких, навіть пропрієтарних застосунках. Розроблено версії інтерпретаторів для JVM, MSIL і ряду інших.

Додати в програмний код новий тип можна шляхом написання нового класу або визначення нового типу в спеціальному модулі розширення, який може бути розроблений на іншій мові. Система класів підтримує повноцінні механізми множинного і одиничного успадкування, а також функції метапрограмування. Можливим, також є пряме успадкування від переважної більшості вбудованих типів розширень.

Рядки і кортежі, що відносяться до вбудованих колекцій, не є змінними, а словники і списки-змінними, їх можна модифікувати динамічно. Тому кортежі в даній мові працюють істотно швидше ніж списки.

Переваги мови наступні.

- 1) інтерпретованість: дозволяє істотно спростити і полегшити налагодження програм, однак при цьому втрачається швидкість виконання програми;
- 2) динамічна типізація: підвищує загальну швидкість розробки коду;
- 3) гнучка підтримка модульності: розроблений модуль може бути легко інтегрований в код довільних програм;
- 4) наявність можливості розробки коду по парадигмі ООП. Планування структури застосунка ставати простіше і зручніше;
- 5) автоматичне «прибирання сміття», тобто звільнення розробника від необхідності стеження за витоками пам'яті;
- 6) підтримка великої кількості модулів, інтегрованих і сторонніх. У ряді поширених випадків для розробки програми досить просто знайти готові модулі і належним чином їх зв'язати між собою. Це дозволяє розробнику зосередити увагу на більш високому рівні розробки і проєктування, працюючи з готовими елементами;

7) кросплатформність – дозволяє запускати програмні застосунки, написані на Python, на практично будь-якій операційній системі, де встановлено інтерпретатор потрібної версії.

3.1.2 IDE PyCharm

PyCharm – це потужне інтегроване середовище розробки (IDE) для мови Python, розроблене компанією JetBrains. Воно забезпечує розширений набір інструментів для написання, налагодження та тестування програм на Python, а також підтримує роботу з веб-технологіями та базами даних. PyCharm пропонує користувачеві сучасний редактор коду з підсвічуванням синтаксису, автоматичним завершенням коду, швидким рефакторингом та навігацією по проекту. Основні особливості PyCharm [14]:

- 1) кросплатформність: доступний для Windows, macOS та Linux;
- 2) розширені можливості налагодження та профілювання коду, включаючи постановку точок зупинки, перегляд змінних і стеку викликів;
- 3) інтеграція з системами контролю версій (Git, Mercurial, SVN);
- 4) підтримка віртуальних середовищ Python та менеджерів пакетів (pip, conda);
- 5) зручний інтерфейс для роботи з базами даних та SQL-запитами;
- 6) інтегроване тестування та підтримка фреймворків (unittest, pytest, Django, Flask).

3.1.3 Мова розмітки QML

QML (Qt Meta Language or Qt Modeling Language) – декларативна мова програмування, заснована на JavaScript, призначена для дизайну застосунків, які роблять основний упор на призначений для користувача інтерфейс. Є частиною Qt Quick, середовища розробки призначеного для користувача інтерфейсу, поширюваної разом з Qt. В основному використовується для створення

застосунків, орієнтованих на мобільні пристрої з сенсорним керуванням та десктопні застосування [15].

QML-документ являє собою дерево елементів. QML елемент, так само, як і елемент Qt, являє собою сукупність блоків: графічних (таких, як rectangle, image) і поведінкових (таких, як state, transition, animation). Ці елементи можуть бути об'єднані, щоб побудувати комплексні компоненти, починаючи від простих кнопок і повзунків і закінчуючи повноцінними застосунками, що працюють з інтернетом.

QML елементи можуть бути доповнені стандартними JavaScript вставками шляхом вбудовування .js файлів. Також вони можуть бути розширені C++ компонентами через Qt framework [16].

Схема взаємодії QML з мовою програмування наведена на рис. 3.1.

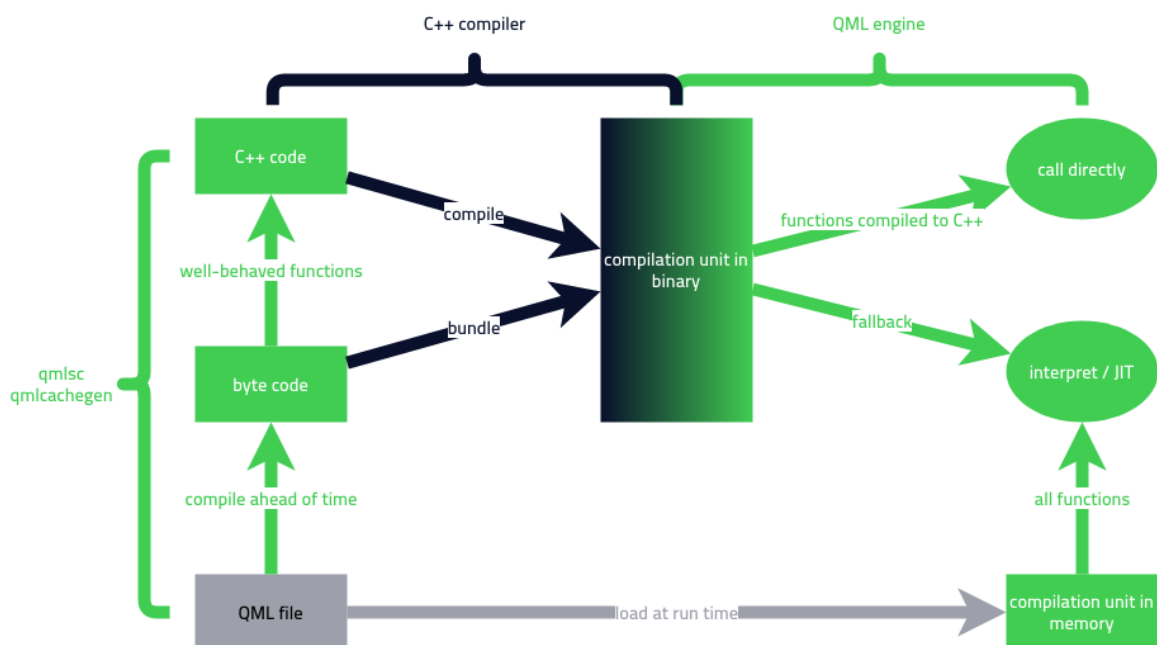


Рисунок 3.1 – Схема взаємодії QML з мовою програмування

3.2 Розробка проєкту програмного застосування

Аналіз і розрахунок полюса повороту судна під час маневрування є важливим етапом для забезпечення точного контролю над рухом судна і уникнення зіткнень. Полюс повороту – це уявна точка, відносно якої судно здійснює поворот.

Послідовність дій при аналізі і розрахунку полюса повороту можна представити в наступних кроках:

1) Збір початкових даних, зокрема на вимірювання чи введення значень швидкості судна як ключового параметру, який впливає на радіус повороту. Дані про швидкість можна отримати від GPS або швидкісного лагу. Також визначається початковий курс (азимут), за яким рухається судно. Дані отримуються з гірокомпаса або магнітного компаса.

Додатково потрібно завдати даних з рульового відхилення, тобто записується кут, на який відхилено рульове перо. Чим більше відхилення, тим сильніше буде поворот і менший радіус.

2) Визначення радіуса повороту, Який залежить від швидкості судна і кута відхилення керма. Слід врахувати, що для великих суден радіус може бути значним, особливо на високих швидкостях, тому потрібно завдати коефіцієнт оновлення.

3) Визначення траєкторії руху судна, що описується певною кривою (циркуляцією), наприклад, за допомогою фільтрів Калмана. Також необхідно визначити кути зміщення, тобто враховується зсув носа і корми судна під час повороту. Корабель завжди повертається довкола полюса повороту, який знаходиться не на самій осі судна.

4) Розрахунок ПП шляхом його визначення на певній відстані від центру судна. ПП розташований на продовженні поздовжньої осі судна, але не завжди збігається з геометричним центром. Для великих суден полюс може перебувати за межами корпусу судна, тому необхідно врахувати, що положення полюса визначається відстанню між центром тяжіння судна і центром повороту.

5) Необхідно здійснити визначення сил і моментів, які діють на судно. Зокрема, точного розрахунку ПП важливо врахувати вектори усіх сил, що діють на судно під час повороту: гідродинамічні сили на корпус судна, сили інерції та дію кермового пера.

Моменти, що впливають на поворот частіше всього є обертальними моментами та діють на корпус судна, через різницю в опорах води з обох боків судна.

Зовнішні умови можуть значно впливати на розрахунок полюса повороту. Течії і вітер можуть змінити траєкторію руху судна, тому потрібно враховувати ці чинники при маневруванні.

Стан руля, двигунів і пропелерів також може впливати на точність розрахунків і фактичну поведінку судна під час повороту.

Після початку маневрування необхідно порівняти фактичну траєкторію руху судна з розрахунковими даними. За потреби, маневр можна коригувати шляхом зміни кута відхилення руля або швидкості судна. Якщо маневр відбувається в умовах високого трафіку або поблизу небезпечних об'єктів, важливо контролювати зони ризику і, за потреби, коригувати їх.

Виконаємо проектування програми засобами мови моделювання UML. Розроблена діаграма варіантів використання програмного забезпечення наведена на рис. 3.2.

Користувач може виконувати розрахунок розташування координат ПП, візуалізувати переміщення ПП при зміні параметрів (маневруванні), переглядати поточне положення судна та попереднього положення, візуалізувати координати ПП графічним чином, візуалізувати результати розрахунків в вікні логу, у таблиці та у вигляді діаграми, здійснювати перерахунок координат супутникової антени судна, виконувати розрахунок розташування ПП при впливі зовнішніх сил та зберігати результати отриманих розрахунків.



Рисунок 3.2 – Діаграма варіантів використання програмного забезпечення

Діаграма головних класів програмного забезпечення наведена на рис. 3.3.

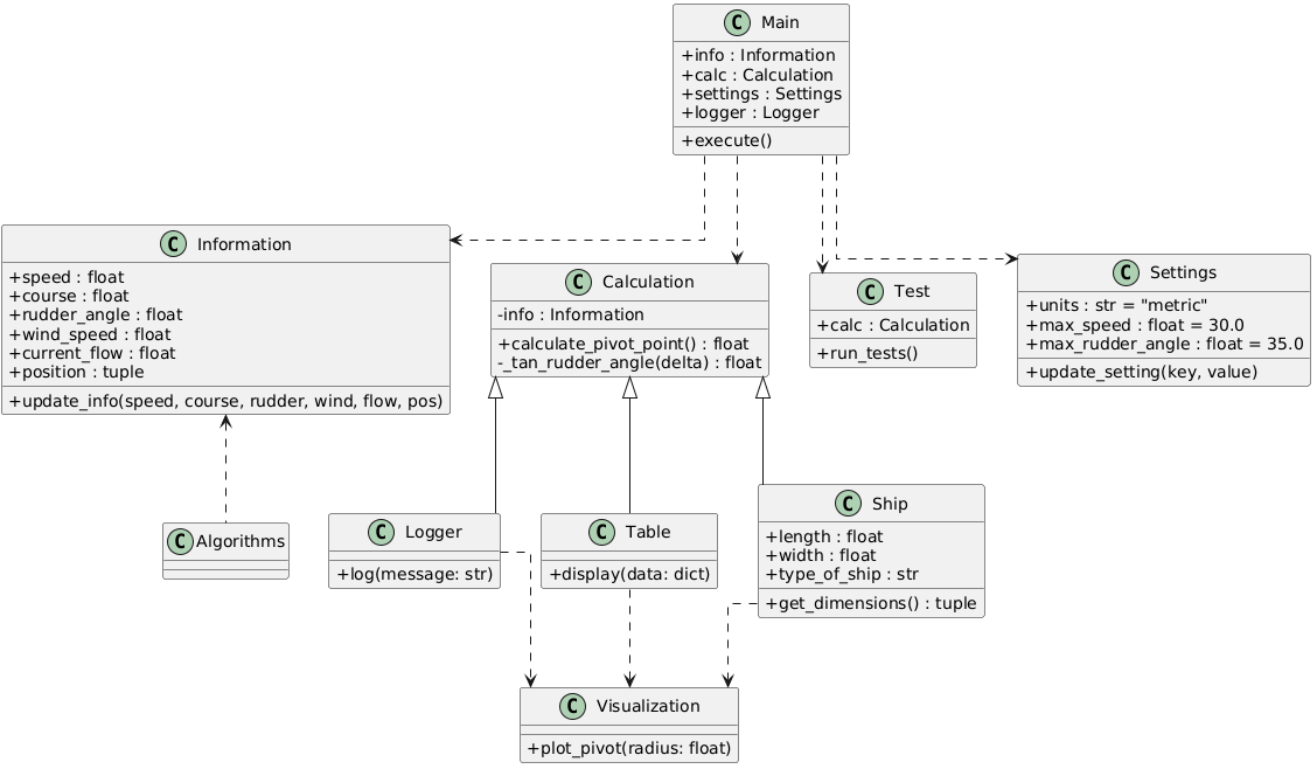


Рисунок 3.3 – Діаграма головних класів програмного забезпечення

Програмні класи в системі підтримки процесів маневрування морського судна та розрахунку полюса повороту на Python мають свої конкретні ролі та відповідальності, зокрема згідно до діаграми класів:

1) Клас `Information` відповідає за збір і зберігання інформації про поточні умови судна, зокрема параметри навігації, дані від датчиків та зовнішні фактори (вітер, течія). Контракт класу: агрегація даних про швидкість судна, кут повороту руля, глибину води, поточні координати тощо.

Містить інтерфейси для отримання даних від зовнішніх систем або сенсорів та методи для оновлення і зберігання даних у реальному часі. Фрагмент програмного коду класу наведено нижче.

```
class Information:
    def __init__(self):
        self.speed = 0.0 # Швидкість судна
        self.course = 0.0 # Поточний курс
        self.rudder_angle = 0.0 # Кут повороту руля
        self.wind_speed = 0.0 # Швидкість вітру
        self.current_flow = 0.0 # Сила течії
        self.position = (0.0, 0.0) # Координати судна (широта, довгота)

    def update_info(self, speed, course, rudder_angle, wind_speed,
current_flow, position):
        """Оновлення інформації про стан судна."""
        self.speed = speed
        self.course = course
        self.rudder_angle = rudder_angle
        self.wind_speed = wind_speed
        self.current_flow = current_flow
        self.position = position
```

2) Клас `Main`, є основним класом системи, який викликає ключові модулі та контролює роботу програми. Взаємодіє з іншими класами для виконання головних завдань (маневрування судном, розрахунок полюса повороту).

Основний зміст полягає у ініціалізації всіх інших класів, здійснює контракт контролю та керування процесами, отримання вхідних даних і передача їх у відповідні модулі.

Фрагмент програмного коду класу наведено нижче.

```
class Main:
    def __init__(self):
        self.info = Information()
        self.calc = Calculation(self.info)
```

```
self.settings = Settings()
self.logger = Logger()

def execute(self):
    """Головна логіка роботи системи."""
    self.logger.log("Запуск системи маневрування")
    self.info.update_info(12.5, 270, 15, 5, 1, (45.32, 24.45))
    result = self.calc.calculate_pivot_point()
    self.logger.log(f"Полюс повороту: {result}")
```

3) Клас **Calculation**, відповідає за всі розрахунки, пов'язані з маневруванням судна, зокрема розрахунок полюса повороту. Містить різні методи для обчислення радіуса повороту, кутової швидкості, та інших параметрів маневру, в тому числі основний метод для розрахунку полюса повороту судна та його форматування згідно вимог.

Фрагмент програмного коду класу наведено нижче.

```
class Calculation:
    def __init__(self, info):
        self.info = info

    def calculate_pivot_point(self):
        """Розрахунок полюса повороту судна."""
        V = self.info.speed
        delta = self.info.rudder_angle
        g = 9.81 # Прискорення вільного падіння
        R = (V ** 2) / (g * self._tan_rudder_angle(delta))
        return R

    def _tan_rudder_angle(self, delta):
        """Внутрішній метод для обчислення тангенса кута."""
        return abs(delta) * 0.01745 # Переведення в радіани
```

4) Клас **Test**, використовується для тестування системи, перевірки правильності обчислень і відповідності очікуваним результатам, містить модульні та валідаційні тести для перевірки процесів обробки даних різними модулями програми, призначений також для тестування окремих методів та функцій або їх комбінацій. Фрагмент програмного коду класу наведено нижче.

```
class Test:
    def __init__(self):
        self.calc = Calculation(Information())

    def run_tests(self):
        """Запуск тестів на різні сценарії."""
        assert self.calc.calculate_pivot_point() > 0, "Помилка у розрахунку полюса повороту"
```

```
print("Усі тести пройшли успішно.")
```

5) Клас `Settings`, зберігає та керує конфігураціями та наборами налаштувань програмного застосування, які можуть бути змінені користувачем або адміністратором. Містить такі константи, як одиниці вимірювання (метри, милі), режими роботи системи, дозволені межі швидкостей і кутів повороту, а також деякі статичні показники. Фрагмент програмного коду класу наведено нижче.

```
class Settings:
    def __init__(self):
        self.units = "metric" # Метрична система вимірювань
        self.max_speed = 30.0 # Максимальна швидкість
        self.max_rudder_angle = 35.0 # Максимальний кут відхилення руля

    def update_setting(self, key, value):
        """Оновлення налаштувань."""
        setattr(self, key, value)
```

6) Клас `Logger`, відповідає за запис і збереження логів (подій, помилок, результатів обчислень) під час роботи системи. Здійснює послідовний запис подій у файл або консоль згідно до проведених обчислень даних, через що використовується для журналізації різних видів помилок і попереджень, що виникли під час використання програмного застосування.

Фрагмент програмного коду класу наведено нижче.

```
class Logger:
    def log(self, message):
        """Запис повідомлень у файл або консоль."""
        print(f"[LOG]: {message}")
```

7) Клас `Table`, відповідає за відображення даних у табличній формі (наприклад, таблиці результатів обчислень або поточних параметрів судна). Реалізує ряд методів для побудови і оновлення таблиць з інформацією та результатами розрахунків показників маневрування при їх динамічній зміні за різних умов.

Фрагмент програмного коду класу наведено нижче.

```
class Table:
    def display(self, data):
        """Виведення таблиці даних."""
        print(f"| {' | '.join(data.keys())} |")
        print(f"| {' | '.join(map(str, data.values()))} |")
```

8) Клас `Ship`, його призначення полягає у відображенні та інтерпретації моделі судна з його характеристиками, яка використовується для розрахунків і

аналізу маневрів. Містить поля, що характеризують основні параметри судна, які в свою чергу можуть впливати на процеси маневрування (тип руля, конфігурація двигуна).

Фрагмент програмного коду класу наведено нижче.

```
class Ship:
    def __init__(self, length, width, type_of_ship):
        self.length = length
        self.width = width
        self.type_of_ship = type_of_ship

    def get_dimensions(self):
        return (self.length, self.width)
```

9) Клас Visualization, необхідний для виконання візуалізації процесів маневрування, руху судна і результатів розрахунку полюса повороту, а також рендерінгу графічного зображення судна. Містить ряд програмних методів для побудови графіків і діаграм для візуалізації траєкторії руху судна, повороту та інших динамічних процесів.

Фрагмент програмного коду класу наведено нижче.

```
import matplotlib.pyplot as plt
class Visualization:
    def plot_pivot(self, radius):
        """Побудова графіка повороту."""
        fig, ax = plt.subplots()
        circle = plt.Circle((0, 0), radius, color='blue', fill=False)
        ax.add_artist(circle)
        ax.set_xlim(-radius, radius)
        ax.set_ylim(-radius, radius)
        plt.show()
```

Структура створених конфігурацій проекту в середовищі розробки наведено на рис. 3.4.

Для відображення процесів маневрування судна в інтерфейсі, побудованому за допомогою QML (Qt Modeling Language), конфігурація файлів QML повинна містити різні графічні елементи та анімації, що відображають стан судна, напрямок його руху, швидкість, кути повороту, полюс повороту тощо.

Основні компоненти QML для відображення процесів маневрування:

1) main.qml. Головний файл, що відповідає за структуру інтерфейсу і розміщення основних елементів для візуалізації процесів маневрування. Canvas

використовується для рендерингу графічних елементів судна та траєкторії його руху. Slider необхідний для зміни параметрів швидкості та кута повороту руля. Button починає процес маневру;

2) shipComponent.qml. Окремий компонент для рендерингу судна на графічному полотні. Він відображає поточний стан судна, включаючи його орієнтацію та полюс повороту. Метод drawShip() малює судно на полотні Canvas. Враховується курс судна та позиція, а позиція ПП позначається відповідним кольоровим колом;

3) maneuverData.qml. Файл для зберігання стану процесів маневрування. Відповідає за передачу даних між компонентами інтерфейсу. position зберігає поточну позицію судна на Canvas, pivotPoint містить координати полюса повороту, а метод startManeuve() реалізує оновлення позиції судна і розрахунку полюса повороту;

Main.qml відповідає за розмітку основного інтерфейсу і взаємодію з користувачем (зміна параметрів, запуск маневрів).

ShipComponent.qml відповідає за розмітку компонентів графічного відображення судна і полюса повороту.

ManeuverData.qml необхідний для розмітки логіки зберігання стану процесів маневрування і розрахунків.

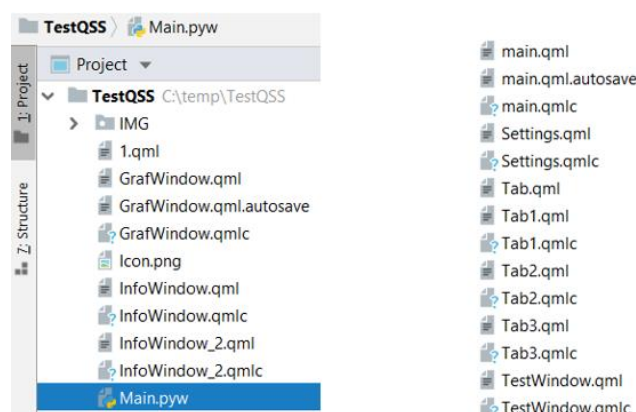


Рисунок 3.4 – Структура конфігурацій проєкту в середовищі розробки

На базі розробленого проєкту програмного застосування стає можливим

подальше наповнення його функціональними програмними можливостями, що буде описано далі.

3.3 Розробка ключових методів програмної логіки

Наведемо опис деяких програмних методів з обробки подій у інтерфейсі програмного застосування.

- `on_speed_slider_changed()`, обробляє зміну значення слайдера швидкості судна. Кожного разу, коли користувач змінює положення слайдера швидкості, цей метод отримує нове значення і передає його для подальших розрахунків полюса повороту;

- `on_rudder_angle_changed()`, обробляє зміну кута повороту руля через слайдер. Метод активується при зміні кута повороту руля і запускає перерахунок параметрів маневрування;

- `on_start_maneuver()`, обробляє натискання кнопки початку маневру. Цей метод запускає основний процес маневрування судна, активуючи розрахунки і оновлення графічного інтерфейсу.

- `on_stop_maneuver()`, обробляє натискання кнопки зупинки маневру. Зупиняє процес маневрування судна та зупиняє таймер, що відповідає за оновлення положення судна;

- `on_timer_tick()`, оновлює положення судна в інтерфейсі на основі таймера. Метод активується через регулярні проміжки часу і відповідає за періодичне оновлення положення судна під час маневру.

- `on_mouse_click()`, обробляє події натискання миші на графічному дисплеї. Цей метод дозволяє користувачеві клікнути по дисплею, щоб змінити точку старту або показати додаткову інформацію.

- `on_key_press()`, обробляє події натискання клавіш. Користувач може використовувати клавіатуру для зміни параметрів маневрування, наприклад, збільшення або зменшення швидкості судна;

– `on_reset()`, скидає всі параметри маневрування до початкових значень. Обробляє подію натискання кнопки очищення даних та повертає всі налаштування до початкового стану;

– `on_save_maneuver_data()`, зберігає параметри маневрування в файл для подальшого аналізу, викликається при натисканні кнопки збереження, записуючи всі дані маневру у файл;

– `on_load_maneuver_data()`, завантажує раніше збережені параметри маневрування з файлу. Обробляє подію завантаження даних та оновлює інтерфейс на основі завантажених даних.

Використаний патерн розподілу програмної логіки проєкту по шарах наведено на рис. 3.5.

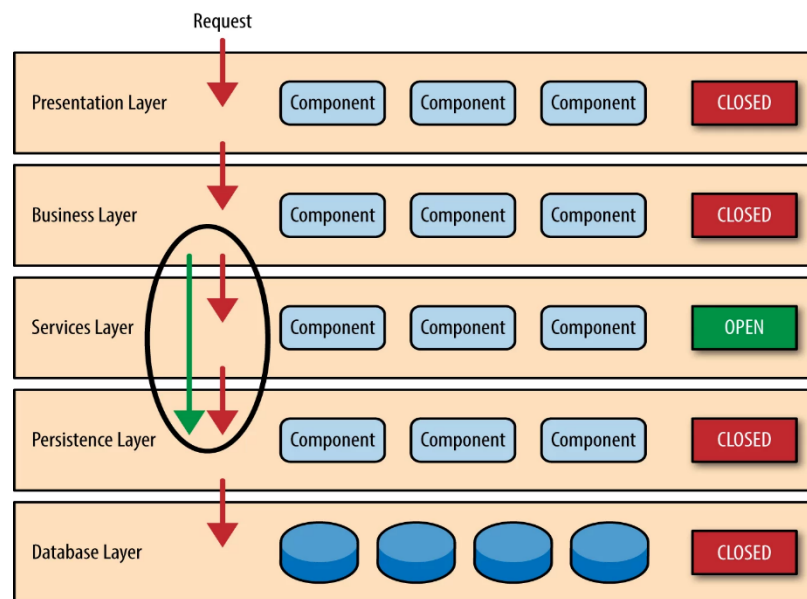


Рисунок 3.5 – Використаний патерн розподілу програмної логіки проєкту по шарах

Поділ програмної логіки застосування автоматизації процесу маневрування морським судном на основі шарової архітектури дозволяє забезпечити чітку структуру, розподіл обов'язків між компонентами та підвищити гнучкість і масштабованість системи. У контексті клієнт-серверної архітектури застосування, шарова модель може виглядати так:

1) Клієнтський рівень, це є інтерфейс користувача, який безпосередньо взаємодіє з екіпажем судна та операторами. Відповідає за введення даних користувачем, відображення важливої інформації, графічний інтерфейс для керування маневруванням. Прикладами елементів є панель керування/моделювання судна, дисплей навігаційної карти, індикатори стану судна (швидкість, напрямок).

2) Рівень логіки застосування, призначений для обробки бізнес-логіки, пов'язаної з маневруванням судна. Цей рівень відповідає за виконання всіх алгоритмів і правил, що визначають поведінку системи.

Функціями є: обчислення траєкторії руху судна на основі введених параметрів (напрямок, швидкість, дані з датчиків), автоматичне коригування курсу судна залежно від зовнішніх умов (вітер, течія, ПП), оцінка ризиків зіткнень або неправильного маневрування на основі інформації з радарів, системи AIS, взаємодія з іншими судновими системами через API для отримання даних від GPS[11], інерційної навігаційної системи та інших сенсорів, обчислення оптимального курсу, керування рухом для уникнення перешкод.

3) Рівень керування даними, відповідає за зберігання, обробки та керування всіма даними, необхідними для роботи програми. Виконує роботу з БД, що містить інформацію про поточні й попередні маршрути, карти, технічні характеристики судна. Може бути застосований для зберігання і обробка даних з датчиків (наприклад, дані з гіроскопів, ехолотів, GPS), які впливають на процес маневрування, а також для роботи з потоковими даними для забезпечення реального часу в прийнятті рішень.

4) Серверний рівень забезпечує доступ до всіх необхідних ресурсів і обчислювальних потужностей, обслуговує запити з боку клієнтського рівня та логіки застосування. Необхідний для обробки запитів клієнта, надання даних і результатів обчислень, проводить взаємодію з зовнішніми сервісами, наприклад, оновлення карт або погодних даних, передача даних до берегових служб через

мережу, а також здійснює керування мережею передачі даних між судном і береговими службами або іншими суднами (через AIS або інші протоколи).

5) Інтеграційний рівень забезпечує роботу застосування синхронно з іншими системами і пристроями судна. Інтеграція з навігаційними системами (AIS, радар, GPS) для отримання даних у реальному часі, взаємодія з системами безпеки, щоб забезпечити реагування на екстрені ситуації, підтримка протоколів передачі даних для обміну інформацією між судном і береговими службами.

Така шарова архітектура дозволяє чітко розділити відповідальності та функціональність системи підтримки маневрування судном [30], що сприяє гнучкості, простоті підтримки та модернізації програмного забезпечення. Кожен шар виконує свою специфічну роль, і їх взаємодія забезпечує плавну роботу системи в цілому.

3.4 Розробка ключових функціональних можливостей та опис процесу використання програмного застосування

Розроблене програмне забезпечення складається з окремих функціональних підсистем, що реалізують різні функції, кожна з яких функціонально виконана у вигляді однієї вкладки на головній формі. Головний клас `main.pyw` містить головний метод з точкою входу в програму, а також програмний код ініціалізації проєкту і підключення необхідних програмних бібліотек. Склад підсистем:

- 1) модуль реалізації навігаційного пристрою розрахунку розташування полюса повороту і візуалізація його переміщення при маневруванні;
- 2) модуль перерахунку координат супутникової антени судна на центр ваги;
- 3) модуль напівавтоматичного навігаційного пристрою розрахунку розташування полюса повороту при впливі зовнішніх сил і результуючої всіх керуючих впливів на корпус судна.

Головне вікно завантаженої системи з відкритим модулем розрахунку розташування полюса повороту судна наведено на рис. 3.6.

Для проведення необхідних операцій в рамках даного програмного модуля користувачеві необхідно ввести вхідні дані у вигляді 6 основних параметрів:

- L – довжина судна між перпендикулярами в метрах;
- V_t – швидкість течії в вузлах;
- K_t – напрямок течії;
- V_n – тангенціальна швидкість носа щодо ґрунту у вузлах;
- V_k – тангенціальна швидкість корми щодо ґрунту у вузлах;
- IK – істинний курс (кут руху).

Перераховані дані вводяться в передбачені поля за допомогою клавіатури, можливе введення дробових і негативних значень для відповідних параметрів, в яких це логічно передбачено.

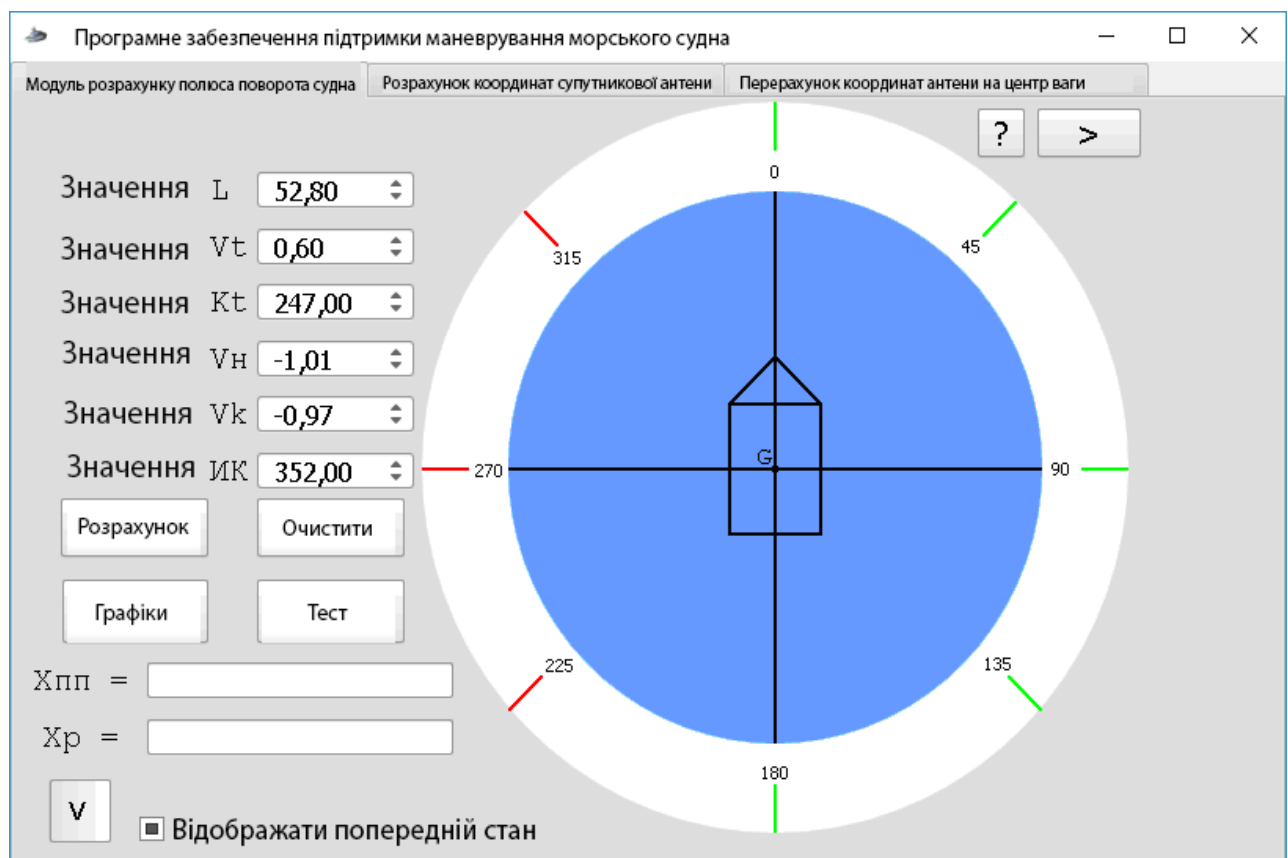


Рисунок 3.6 – Головне вікно завантаженої системи

Після введення даних користувач може натиснути на наступні кнопки:

1) розрахунок – ініціює подія розрахунку розташування абсциси полюса повороту судна ($X_{пп}$), абсцису точки прикладання рівнодіючої бічних сил (X_p), з відповідним знаком «+» в сторону носа і «-» в бік корми щодо центра ваги, а також виконує динамічну візуалізацію координат в контейнері графічного представлення схематичного відображення судна;

2) скидання – ініціює подія очищення полів введення даних і обнуляє візуалізацію розташування;

3) графіки – ініціює подія відкриття форми перегляду результатів виконаних розрахунків в табличному і графічному (у вигляді діаграм та графіків) вигляді;

4) тест – ініціює подія відкриття вікна для вибору тестових вибірок вихідних даних по вхідних параметрів і відображення результатів розрахунку.

Приклад результатів розрахунку і візуалізації розташування абсциси полюса повороту судна і абсциси точки прикладання рівнодіючої бічних сил наведено на рис. 3.7.

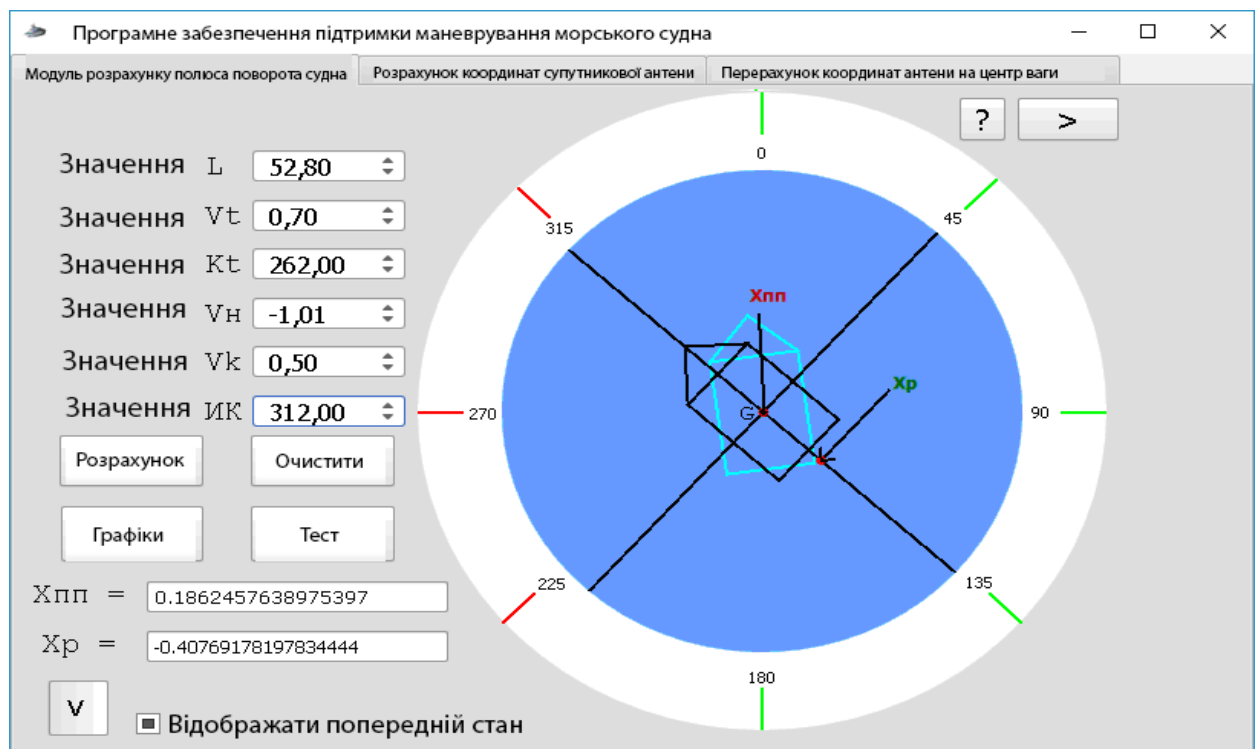


Рисунок 3.7 – Приклад розрахунку і візуалізації розташування абсцис полюса повороту судна і точки прикладання рівнодіючої бічних сил

При бажанні користувач може включити опцію відображення стану положення судна на попередньому етапі розрахунку для виконання аналітичного порівняння. Для цього передбачений відповідний компонент вибору з підписом «Відображати попередній стан». Поруч з даними компонентом розташована програмна кнопка з написом у вигляді стрілки, що дозволяє включити відображення результатів виконаних розрахунків в табличному вигляді. При перемиканні за стовпцями даної таблиці здійснюється динамічна візуалізація результатів розрахунку в контейнері графічного представлення абсциси полюса повороту судна і абсциси точки прикладання рівнодіючої бічних сил. Прибрати цю таблицю можна шляхом повторного натискання на вищезгадану кнопку.

Приклад візуалізації результатів проведених розрахунків в табличній формі наведено на рис. 3.8.

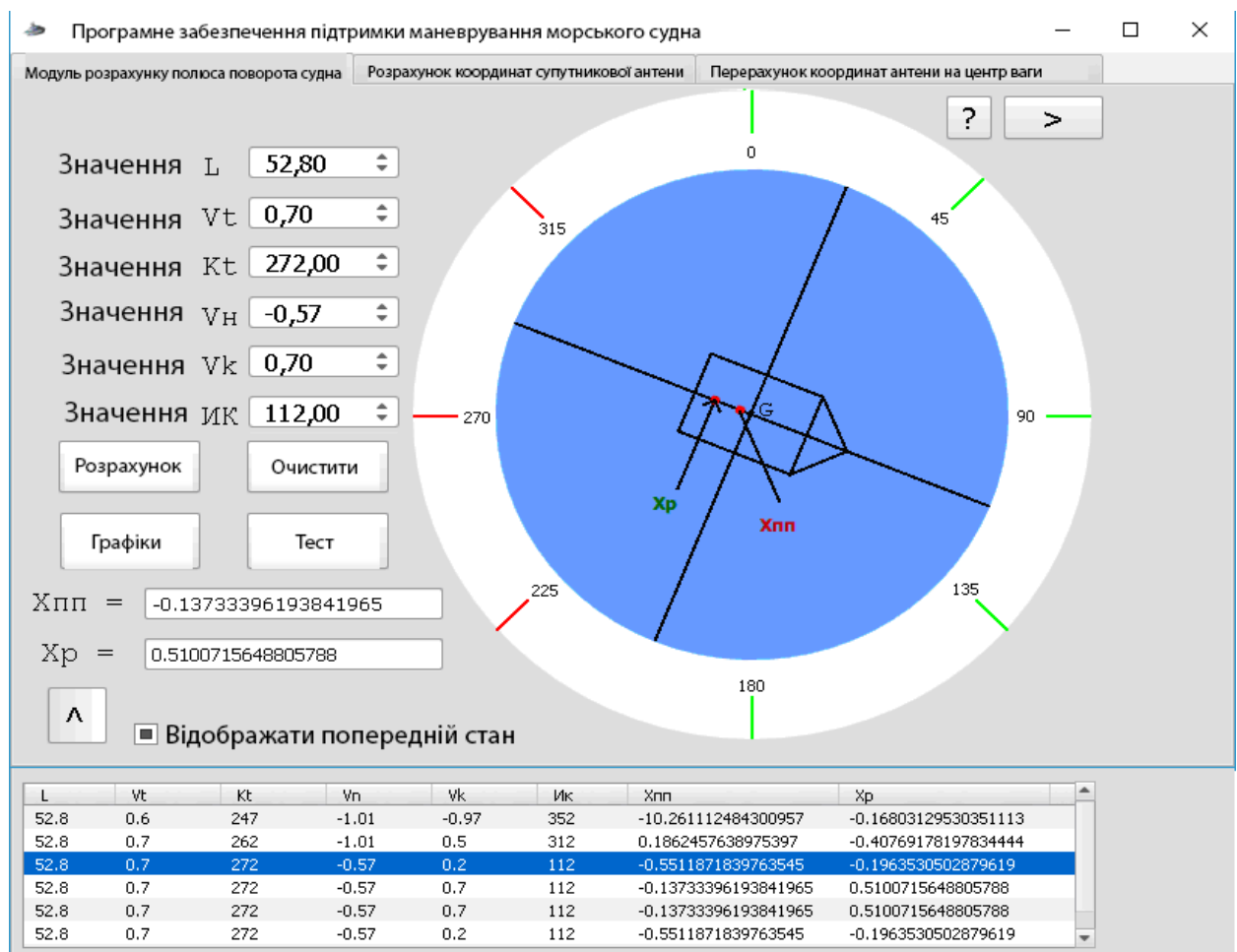


Рисунок 3.8 – Приклад візуалізації результатів проведених розрахунків в табличній формі

Для здійснення покрокового відстеження процесу розрахунку вихідних значень передбачена можливість виклику балки виконаних операцій. Це можливо шляхом натискання на відповідну програмну кнопку з написом у вигляді стрілки в правому верхньому куті форми даного модуля. Після цього відображається текстова панель з результатами розрахунку проміжних значень. Приховування даної панелі можливо шляхом повторного натискання по даній кнопці.

Приклад ітераційного відображення процесу розрахунку в формі текстового балки наведено на рис. 3.9.

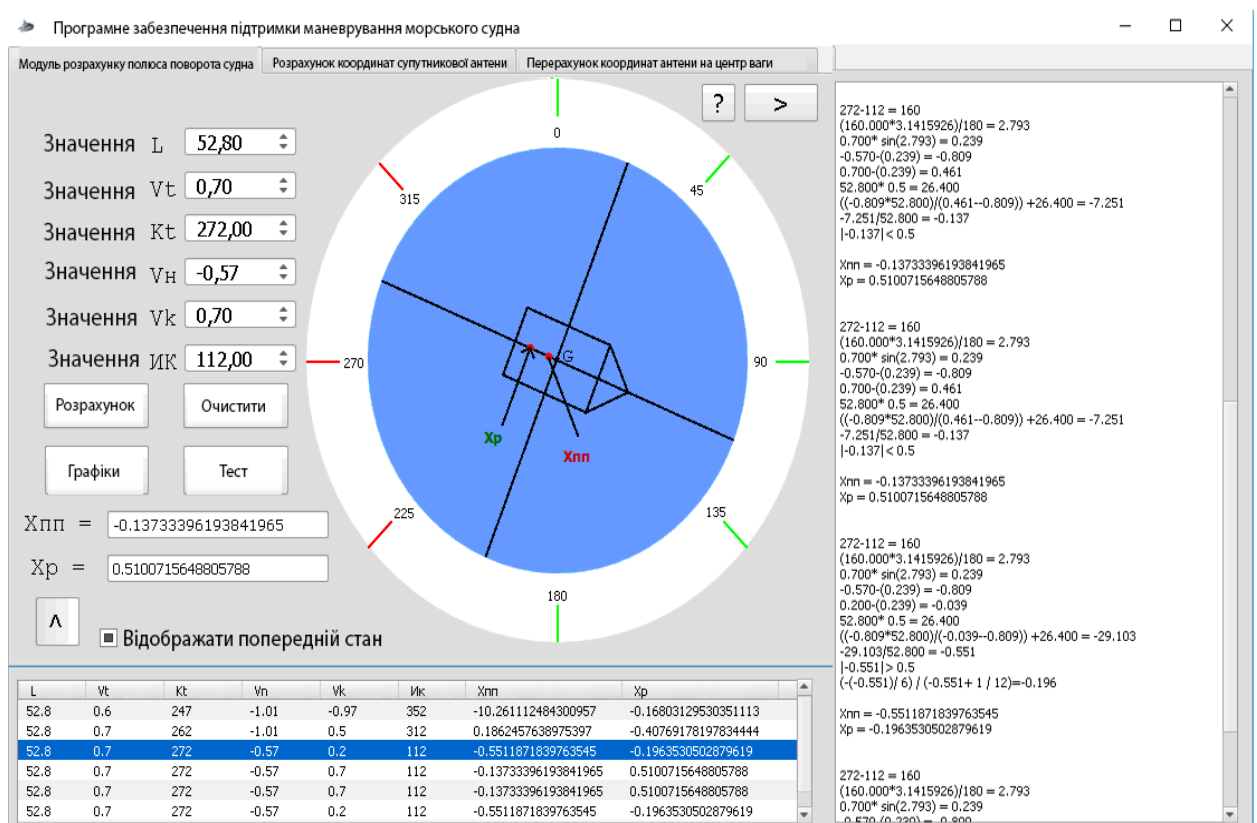


Рисунок 3.9 – Приклад ітераційного відображення процесу розрахунку в формі текстового балки

При натисканні користувачем на кнопку з написом «Графіки» здійснюється виклик вікна, призначеного для відображення всіх отриманих результатів розрахунку і їх графічного відображення у вигляді точкової і стовпчастої діаграм.

При переміщенні по рядках таблиці проводиться підсвічування поточної ітерації розрахунку для більшої наочності. Діаграми забезпечені легендою підписи даних, у відповідність з якою зеленим кольором позначені значення абсциси полюса повороту судна, а червоним абсциси точки прикладання рівнодіючої бічних сил.

Вікно відображення отриманих результатів розрахунку в графічному вигляді в формі діаграм наведено на рис. 3.10.

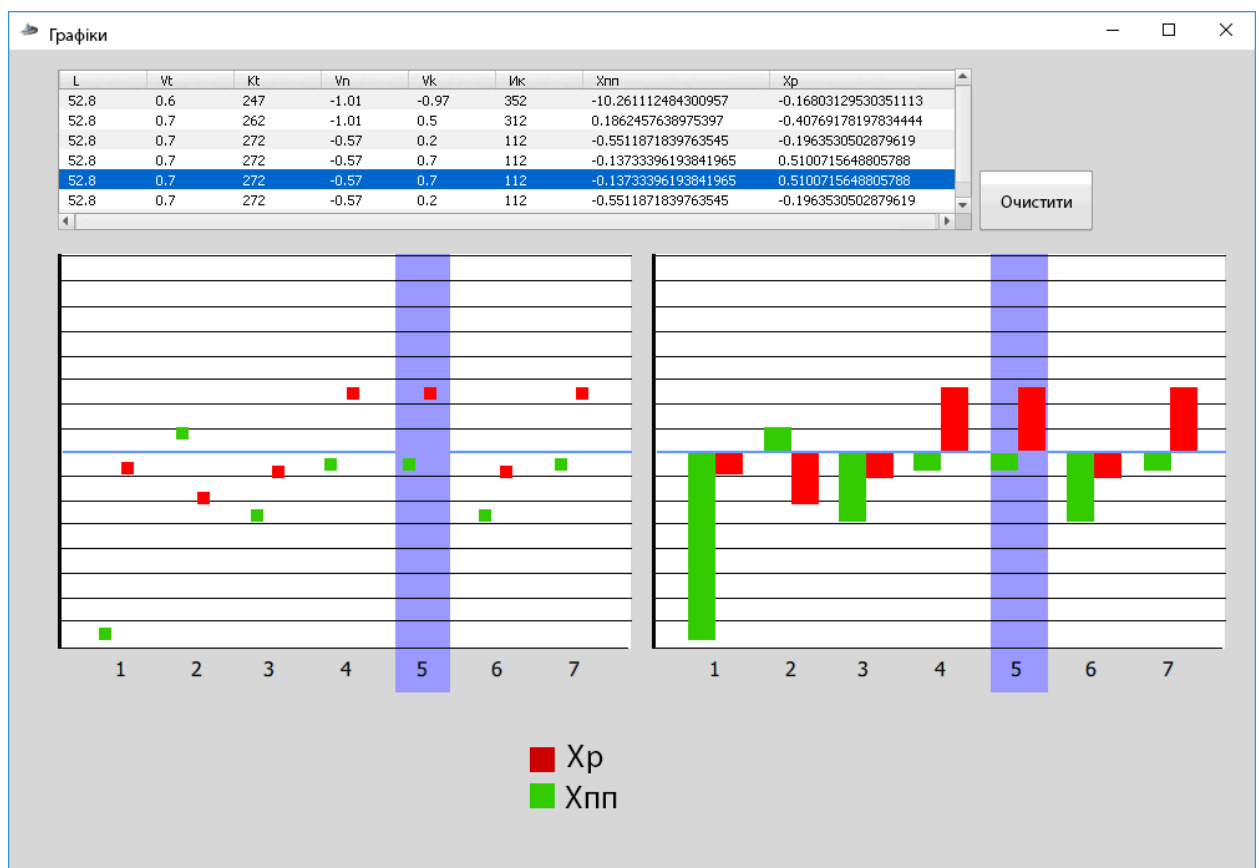


Рисунок 3.10 – Вікно відображення отриманих результатів розрахунку в графічному вигляді в формі діаграм

У разі натискання користувачем по кнопці з написом «Тест» відкривається форма з можливістю вибору однієї з 10 різних тестових вибірок вхідних даних для швидкого розрахунку підсумкових вихідних значень, в разі, коли графічна візуалізація не потрібно або не є принциповою.

Форма швидкого розрахунку абсциси полюса повороту і абсциси точки прикладання рівнодіючої бічних сил на тестових вибірках відповідним чином приведена на рис. 3.11.

The screenshot shows a software window titled 'Тестування' (Testing). On the left, there is a list of test samples (Тестова вибірка) from №1 to №10, each with a checkbox. Below the list are two buttons: 'Тест' (Test) and 'Очистити' (Clear). The main area of the window displays calculation results for three selected samples: №1, №4, and №8. The calculations include coordinates (Xnp, Xp), parameters (L, Vn, Vt, Vk, Kt, IK), and intermediate steps for determining the coordinates of the point of application of the resultant force.

Тестування

☒ Тестова вибірка №1
☐ Тестова вибірка №2
☐ Тестова вибірка №3
☒ Тестова вибірка №4
☐ Тестова вибірка №5
☐ Тестова вибірка №6
☐ Тестова вибірка №7
☒ Тестова вибірка №8
☐ Тестова вибірка №9
☐ Тестова вибірка №10

Тест **Очистити**

Xnp = -10.261112484300957
Xp = -0.16803129530351113

№4
L= 52.8, Vn= -0.58, Vt= 0.6, Vk= -1.53, Kt= 247, IK= 1.
247-1 = 246
(246.000*3.1415926)/180 = 4.294
0.600* sin(4.294) = -0.548
-0.580-(-0.548) = -0.032
-1.530-(-0.548) = -0.982
52.800* 0.5 = 26.400
((-0.032*52.800)/(-0.982--0.032)) +26.400 = 28.171
28.171/52.800 = 0.534
|0.534|> 0.5
(-(0.534)/ 6) / (0.534+ 1 / 12)=-0.144
Xnp = 0.5335502560925534
Xp = -0.14415206403018735

№8
L= 52.8, Vn= 0.11, Vt= 0.6, Vk= -1.81, Kt= 247, IK= 25.
247-25 = 222
(222.000*3.1415926)/180 = 3.875
0.600* sin(3.875) = -0.401

Рисунок 3.11 – Форма швидкого розрахунку абсциси полюса повороту і абсциси точки прикладання рівнодіючої бічних сил на тестових вибірках

Для ознайомлення з математичними основами розрахунків у програмі реалізовано форму «Інформація». Її можна викликати, натиснувши на кнопку зі знаком питання (?) у верхньому правому куті вікна.

Довідкова інформація за концепцією проведення розрахунків і короткий відображення алгоритму роботи модуля розрахунку розташування полюса повороту і візуалізація його переміщення при маневруванні наведені на рис. 3.12 – 3.14 відповідно.

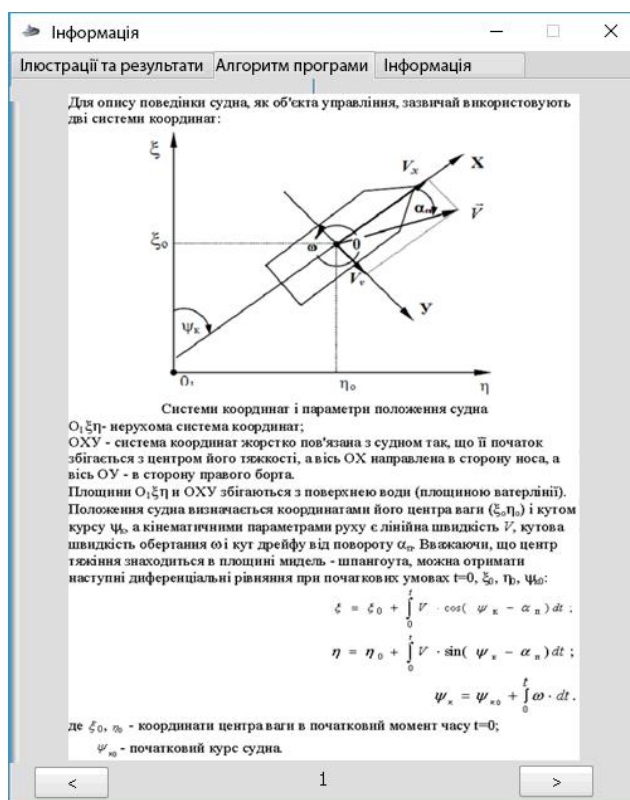


Рисунок 3.12 – Довідкова інформація за концепцією проведення розрахунків



Рисунок 3.13 – Довідкова інформація за концепцією проведення розрахунків

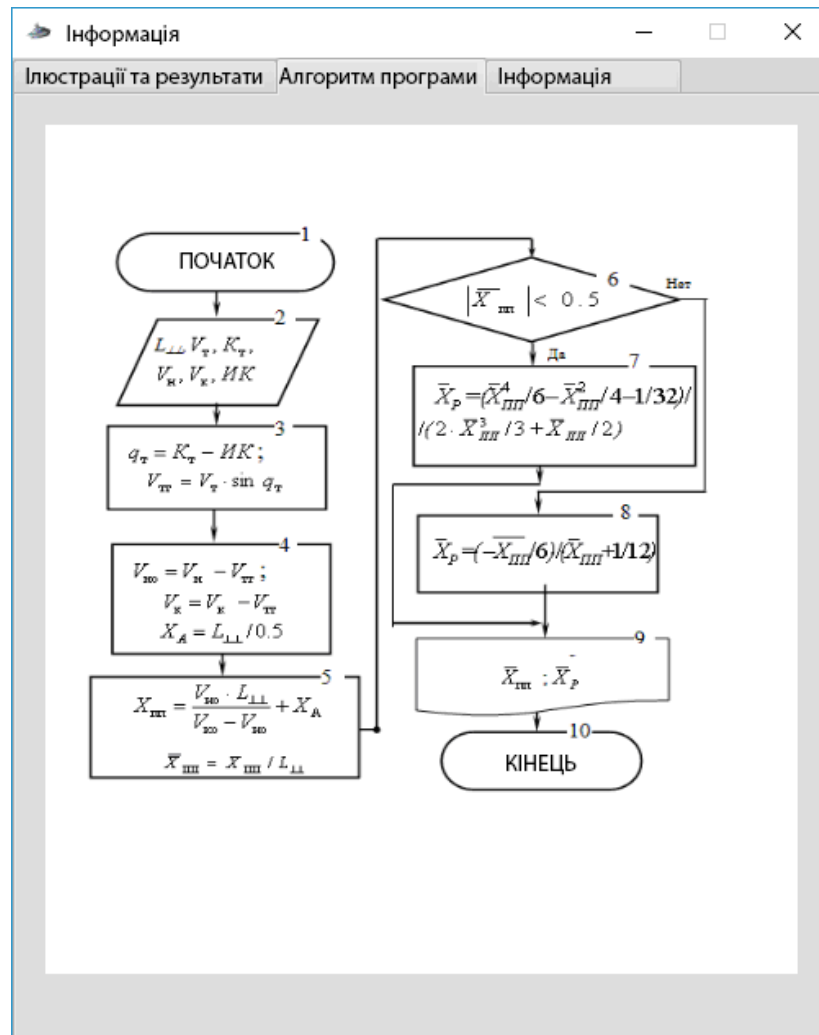


Рисунок 3.14 – Коротке відображення алгоритму роботи модуля розрахунку розташування полюса повороту і візуалізація його переміщення при маневруванні

Головне вікно завантаженої системи з відкритим модулем перерахунку координат супутникової антени судна на центр ваги наведено на рис. 3.15. Користувач повинен ввести значення координат за певними супутниковою системою даними ϕ_a і λ_a , вказати напрямок (південь / північ для ϕ_a або захід / схід для λ_a) які попередньо розраховуються при курсі K в межах $0^0 \leq K \leq 90^0$.

Також, користувач повинен ввести значення проєкцій місця розташування антени L_{xm} та L_{ym} в метрах на відповідну суднову вісь. При цьому позитивне значення проєкції прийнято по осі Y зі знаком плюс в сторону правого борта від центра ваги і по осі X в бік носа.

Рисунок 3.15 – Інтерфейс модуля перерахунку координат супутникової антени судна на центр ваги

Можливі чотири випадки розташування антени по відношенню до центру тяжіння:

- в сторону носа з правого борту;
- в сторону носа з лівого борту;
- в бік корми з правого борту;
- в бік корми з лівого борту.

Для вибору варіанту розташування антени для кожного з цих випадків передбачені відповідні компоненти в інтерфейсі модуля, реалізовані у вигляді підписаних чекбоксів. З урахуванням географічних координат можливі 64 різні розрахункові схеми.

Після того, як користувач задав все вхідні дані може натиснути на кнопку з написом «Розрахунок», після чого відбувається відображення результатів

перекладу проєкцій в морські милі і точного перерахунку координат супутникової антени судна на центр ваги (ϕ_G і λ_G). Приклади відображення отриманих результатів розрахунків наведені на рис. 3.16 – 3.17.

Рисунок 3.16 – Приклад відображення результатів перерахунку координат супутникової антени судна на центр ваги при розташуванні антени зліва по носу

Аналогічно модулю розрахунку розташування полюса повороту і візуалізації його переміщення при маневруванні в даному програмному модулі передбачена можливість перегляду довідкових матеріалів і математичних формул.

Вікна довідкової інформації по концепції проведення розрахунків наведено на рис. 3.18 – 3.19. Для перегляду цієї інформації можна використовувати кнопки перемикання між інформаційними сегментами, виконані за принципом програмної пагінації.

Рисунок 3.17 – Приклад відображення результатів перерахунку координат супутникової антени судна на центр ваги при розташуванні антени справа по носу

Форму відображення алгоритму роботи модуля перерахунку координат супутникової антени судна на центр тяжіння наведено на рис. 3.21.

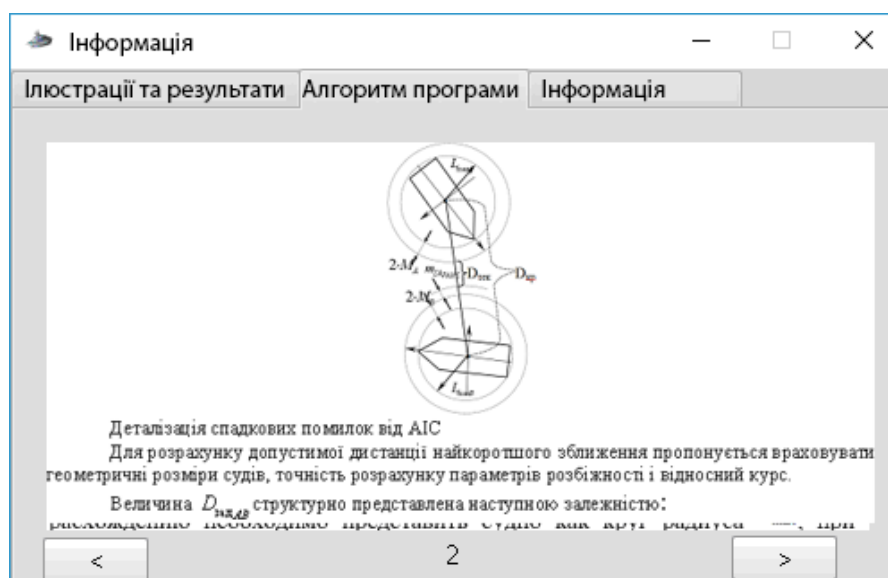


Рисунок 3.18 – Вікно довідкової інформації по концепції проведення розрахунків

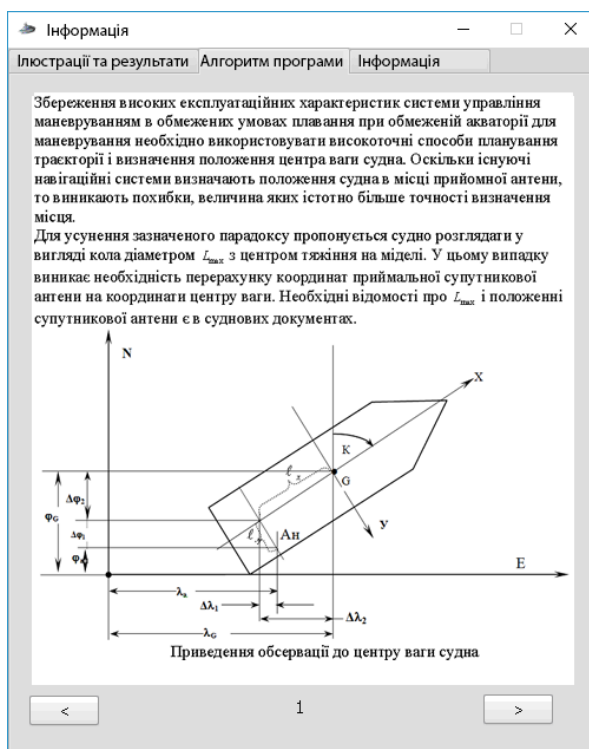


Рисунок 3.19 – Вікно довідкової інформації по концепції проведення розрахунків

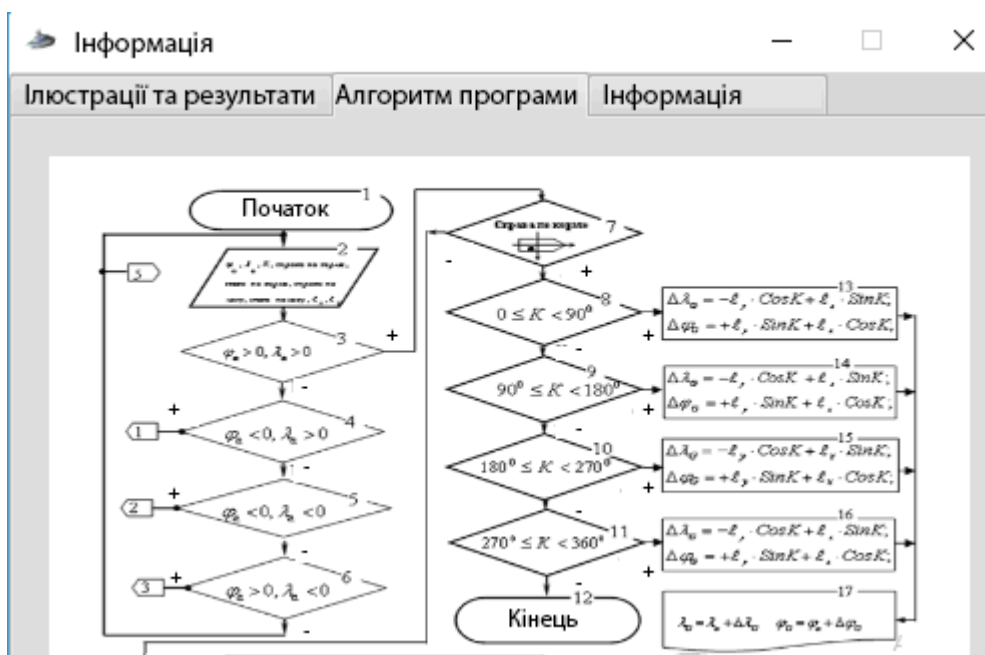


Рисунок 3.20 – Форма відображення алгоритму роботи модуля перерахунку координат супутникової антени судна на центр ваги

Головне вікно завантаженої системи з відкритим модулем напівавтоматичного навігаційного пристрою розрахунку розташування полюса

повороту при впливі зовнішніх сил і результуючої всіх керуючих впливів на корпус судна наведено на рис. 3.21.

Програмне забезпечення підтримки маневрування морського судна

Модуль розрахунку полюса повороту судна | Розрахунок координат супутникової антени | Перерахунок координат антени на центр ваги

$L = 0,00$ $q_w = 0,00$ $S_H = 0,00$ $M_j = 0,00$ $I_{цп} = 0,00$

$P_B = 0,00$ $P_j = 0,00$ $V_k = 0,00$ $C_{ay} = 0,00$

$P = 0,00$ $I = 0,00$ $a = 0,00$ $P_a = 0,00$ $M = 0,00$ ☐

$P = 0,00$ $I = 0,00$ $a = 0,00$ $P_a = 0,00$ $M = 0,00$ ☐

$P = 0,00$ $I = 0,00$ $a = 0,00$ $P_a = 0,00$ $M = 0,00$ ☐

$P = 0,00$ $I = 0,00$ $a = 0,00$ $P_a = 0,00$ $M = 0,00$ ☐

$P = 0,00$ $I = 0,00$ $a = 0,00$ $P_a = 0,00$ $M = 0,00$ ☐

$P = 0,00$ $I = 0,00$ $a = 0,00$ $P_a = 0,00$ $M = 0,00$ ☐

$P = 0,00$ $I = 0,00$ $a = 0,00$ $P_a = 0,00$ $M = 0,00$ ☐

$X_p =$

$X_n =$

Розрахунок

Рисунок 3.21 – Інтерфейс модуля напівавтоматичного навігаційного пристрою розрахунку розташування полюса повороту при впливі зовнішніх сил

Для виконання розрахунку необхідно ввести наступні основні вхідні дані:

- L – довжина судна між перпендикулярами;
- q_w – курсовий кут уявного вітру;
- S_H – площа проєкції надводної частини корпусу судна;
- M_j – момент бічної сили;
- $I_{цп}$ – абсциса центру парусності надводної частини корпусу;
- P_B – масова щільність повітря;
- P_j – величина упору гвинта;
- V_k – швидкість удаваного вітру;
- C_{ay} – безрозмірний аеродинамічний коефіцієнт.

Згідно реалізованому математичного алгоритму передбачена можливість введення до 7 різних наборів даних по бічних силам (шляхом активації відповідних чекбоксів на формі в правій її частині), таким як:

- бічна сила від керма;
- бічна сила від керма і носового підрулює пристрою;
- бічна сила від керма і кормового підрулює пристрою;
- бічна сила від носового і кормового підрулює пристрою;
- бічна сила від носового підрулює пристрою;
- бічна сила від кормового підрулює пристрою;
- сумарна бічна сила від керма, носового і кормового підрулює пристрою.

Як параметр a виступає напрямок дії сили упору, а P_a її скалярне значення.

Приклад відображення результатів короткого розрахунку розташування полюса повороту при впливі зовнішніх сил наведено на рис. 3.22.

Програмне забезпечення підтримки маневрування морського судна

Модуль розрахунку полюса поворота судна | Розрахунок координат супутникової антени | Перерахунок координат антени на центр ваги

$L = 52,00$ $q_w = 10,00$ $S_n = 21,00$ $M_j = 7,00$ $I_{цп} = 6,00$

$P_v = 5,00$ $P_j = 11,00$ $V_k = 3,00$ $C_{ay} = 7,00$

$P = 0,00$ $I = 0,00$ $a = 0,00$ $P_a = 0,00$ $M = 0,00$ ☐

$P = 0,00$ $I = 0,00$ $a = 0,00$ $P_a = 0,00$ $M = 0,00$ ☐

$P = 0,00$ $I = 0,00$ $a = 0,00$ $P_a = 0,00$ $M = 0,00$ ☐

$P = 0,00$ $I = 0,00$ $a = 0,00$ $P_a = 0,00$ $M = 0,00$ ☐

$P = 0,00$ $I = 0,00$ $a = 0,00$ $P_a = 0,00$ $M = 0,00$ ☐

$P = 0,00$ $I = 0,00$ $a = 0,00$ $P_a = 0,00$ $M = 0,00$ ☐

$P = 0,00$ $I = 0,00$ $a = 0,00$ $P_a = 0,00$ $M = 0,00$ ☐

$X_p = 0.25448$ $X_n = 8.22676$

Рисунок 3.22 – Приклад відображення результатів короткого розрахунку розташування полюса повороту при дії зовнішніх сил

Текстові поля, в які здійснюється вивід отриманих результатів розрахунку розташування абсциси полюса повороту при впливі зовнішніх сил і результуючої всіх керуючих впливів на корпус судна, а також абсциси точки прикладання рівнодіючої бічних сил, розташовані в лівому нижньому кутку форми. Отримані значення можуть бути скопійовані з текстового поля в буфер обміну даними для подальшої обробки або вставки.

Приклад відображення результатів докладного розрахунку розташування полюса повороту при впливі зовнішніх сил наведено на рис. 3.23.

Програмне забезпечення підтримки маневрування морського судна

Модуль розрахунку полюса поворота судна | Розрахунок координат супутникової антени | Перерахунок координат антени на центр ваги

$L = 52,00$ $qW = 15,00$ $SH = 22,00$ $Mj = -3,00$ $I_{цп} = 10,00$
 $Pв = 1,00$ $Pj = 29,00$ $Vk = -4,00$ $Cay = 7,00$
 $P = 3,00$ $I = -2,00$ $a = 4,00$ $Pa = -3,00$ $M = 2,00$ ☐
 $P = 5,00$ $I = -3,00$ $a = 10,00$ $Pa = 7,00$ $M = 3,00$ ☐
 $P = 7,00$ $I = 6,00$ $a = 10,00$ $Pa = -4,00$ $M = -4,00$ ☐
 $P = 3,00$ $I = 2,00$ $a = -5,00$ $Pa = -5,00$ $M = -4,00$ ☐
 $P = -2,00$ $I = -4,00$ $a = 2,00$ $Pa = 4,00$ $M = 1,00$ ☐
 $P = -4,00$ $I = 3,00$ $a = 5,00$ $Pa = 3,00$ $M = 4,00$ ☐
 $P = 2,00$ $I = -5,00$ $a = 4,00$ $Pa = 1,00$ $M = 1,00$ ☐

$Xp = 0.27235$
 $Xn = 8.80387$

Розрахунок

Рисунок 3.23 – Приклад відображення результатів докладного розрахунку розташування полюса повороту при впливі зовнішніх сил

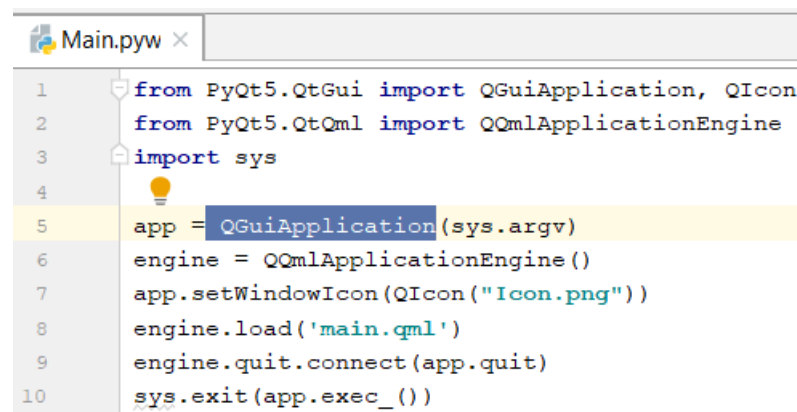
При керуванні процесом маневрування в звичайних умовах судноводій діє на рівні стійкого досвіду.

В умовах обмеженого простору збільшується число і склад елементарних операцій, а в екстремальних умовах необхідно використання розумових здібностей для прийняття рішення, що призводить до ускладнення процесу керування.

Фрагмент логіки конфігурації об'єкту судна наведено нижче.

```
model: libraryModel
  onCurrentRowChanged: {
    var id
    id = tableView.currentRow
    if (id === 0){ rectangle7.x = 28;rectangle19.x = 28 }
    if (id === 1){ rectangle7.x = 89;rectangle19.x = 89 }
    if (id === 2){ rectangle7.x = 148;rectangle19.x = 148 }
    if (id === 3){ rectangle7.x = 209;rectangle19.x = 209}
    if (id === 4){ rectangle7.x = 272;rectangle19.x = 271 }
    if (id === 5){ rectangle7.x = 332;rectangle19.x = 330}
    if (id === 6){ rectangle7.x = 394;rectangle19.x = 392 }
  }
}
```

Фрагмент структури головного класу виклику зв'язки програмного застосування наведена на рис. 3.24.



```
Main.pyw x
1 from PyQt5.QtGui import QApplication, QIcon
2 from PyQt5.QtQml import QmlApplicationEngine
3 import sys
4 
5 app = QApplication(sys.argv)
6 engine = QmlApplicationEngine()
7 app.setWindowIcon(QIcon("Icon.png"))
8 engine.load('main.qml')
9 engine.quit.connect(app.quit)
10 sys.exit(app.exec_())
```

Рисунок 3.24 – Структура головного класу виклику зв'язків програмного застосування

Швидкоплинність процесу керування і недовік часу для виконання необхідних розрахунків для отримання коректної інформації про процес руху вимагає попередньої підготовки команди містка для керування коштами руху і маневрування.

При зміні точки прикладання буксирів і засобів керування становище ПП змінюється, що суттєво впливає на характер маневрування. Знання причин і закономірностей переміщення положення ПП дозволить визначити стан ПП заздалегідь, до маневру, що часто є вкрай важливим, особливо при маневруванні в каналах і фарватерах.

Фрагмент програмного коду виконання перерахунку полюса повороту

наведено нижче.

```
if (xp>0) {  
    var j  
    j = 158-height_xp;  
  
    if(i==0){r1.y = j; r1.height = height_xp;r8.y = j;}  
    if(i==1){r2.y = j; r2.height = height_xp;r9.y = j;}  
    if(i==2){r3.y = j; r3.height = height_xp;r10.y = j;}  
    if(i==3){r4.y = j; r4.height = height_xp;r11.y = j;}  
    if(i==4){r5.y = j; r5.height = height_xp;r12.y = j;}  
    if(i==5){r6.y = j; r6.height = height_xp;r13.y = j;}  
    if(i==6){r7.y = j; r7.height = height_xp;r14.y = j;}}  
else{  
  
    if(i==0){r1.y = 160; r1.height = height_xp;r8.y = 150+height_xp;}  
    if(i==1){r2.y = 160; r2.height = height_xp;r9.y = 150+height_xp;}  
    if(i==2){r3.y = 160; r3.height = height_xp;r10.y = 150+height_xp;}  
    if(i==3){r4.y = 160; r4.height = height_xp;r11.y = 150+height_xp;}  
    if(i==4){r5.y = 160; r5.height = height_xp;r12.y = 150+height_xp;}  
    if(i==5){r6.y = 160; r6.height = height_xp;r13.y = 150+height_xp;}  
    if(i==6){r7.y = 160; r7.height = height_xp;r8.y = 150+height_xp;}}  
}
```

Однак існує цілий ряд операцій, коли завчасно виконати розрахунки не представляється можливим.

З цієї причини пропоноване навігаційний пристрій дозволяє оперативно вирішувати визначення абсциси ПП для розстановки буксирів, оцінки ширини смуги при прямолінійній і криволінійній русі для забезпечення безаварійного плавання.

При цьому автоматичний пристрій є кращим, оскільки не вимагає від оператора відволікатися на введення необхідних даних, які автоматично проступають від доплерівського лага.

Складовою частиною вироблення оптимальних рішень в процесі керування рухом суден у морі є розрахунки, пов'язані з їх маневруванням, які необхідно вирішувати в режимі реального часу. Збільшення числа сучасних суден, їх водотоннажності і швидкостей руху в водних акваторіях при обмежених умовах плавання, вимагає прискорення процесу прийняття рішення по маневруванню, а в деяких випадках екстреного виконання маневру.

В умовах, що склалися перспективним напрямком розвитку навігаційно-керуючого обладнання стає створення інтегрованих систем орієнтації, навігації та

прийняття рішення, що володіють високим рівнем точності і автоматизації виконуваних функцій. При цьому їх можливості розширюються і можуть доповнюватися введенням функцій, пов'язаних з підтримкою прийняття рішень з керування судном.

Підвищення точності визначення місця за рахунок перерахунку координат дозволить підвищити безпеку розходження суден за рахунок введення обґрунтованої допустимої дистанції зближення і створення автоматизованого пристрою попередження посадки на міліну. Виконання розрахунків вручну для попередження посадки на міліну практично неможливо, а поправки перерахунку координат на центр ваги на порядок перевищують похибки визначення місця судна супутниковими системами в диференціальному режимі.

Таким чином, розроблене програмне забезпечення дозволяє здійснювати підтримку маневрування морським судном у напіваавтоматичному режимі.

Висновки до розділу 3

В результаті виконання третьому розділу виконана розробка програмного забезпечення оцінки показників навігаційного маневрування морського транспорту.

Обґрунтовано застосування обраних засобів програмної розробки. Проведена розробка проєкту програмного застосування, структури та ключових методів програмної логіки, основних функціональних можливостей та опис процесу використання програмного застосування.

На базі отриманих результатів стає можливим подальше дослідження використання програмного застосування при моделюванні різних сценаріїв маневрування, що буде проведено у наступному розділі роботи.

4 ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ МОДЕЛЮВАННЯ СЦЕНАРІЇВ МАНЕВРУВАННЯ СУДНОМ

4.1 Формалізація процесу використання фільтра Калмана для дослідження сценаріїв моделювання маневрування

Для проведення дослідження до розробленого функціоналу програмного забезпечення вирішено створити програмний скрипт моделювання процесу побудови оптимальної траєкторії та симуляційну модель руху судна за нею з урахуванням даних розрахунку полюсу повороту. Це стає можливим шляхом адаптації використання у якості основи процесу моделювання фільтра Калмана [28].

Фільтр Калмана застосовується для маневрування судном для оцінки його положення, швидкості та інших параметрів, використовуючи дані від датчиків, таких як GPS, гіроскопи та акселерометри. Судно, як складна система, схильна до впливу різних факторів: вітру, течій, помилок датчиків і неточностей керування. Фільтр Калмана допомагає обробляти ці дані та передбачати майбутнє становище судна з урахуванням існуючих шумів та невизначеностей.

На першому етапі роботи створеного модуля фільтр Калмана використовує математичну модель руху судна, щоб передбачити його майбутні координати і швидкість. Це важливо для прогнозування можливих варіантів розташування судна у просторі для його подальшого трекінгу та прокладення шляху руху.

Коли надходять дані з датчиків (в нашому випадку це результати розрахунків полюса повороту судна з основного програмного застосування, що збережені до табличного файлу), фільтр порівнює передбачені значення з реальними та коригує розрахунки, щоб точніше оцінити поточне положення судна. Це допомагає мінімізувати помилки, спричинені шумами та неточностями даних.

На етапі прогнозування фільтр використовує рівняння руху судна для прогнозування його майбутнього стану. Зазвичай це включає математичне моделювання динаміки судна, яке описується системою рівнянь:

$$x_{k|k-1} = Ax_{k-1|k-1} + Bu_{k-1}, \quad (4.1)$$

де: $x_{k|k-1}$ – передбачений стан системи в момент часу;

A – матриця стану, що описує динаміку системи (наприклад, зміна положення та швидкості судна);

$x_{k-1|k-1}$ – оцінка стану в попередній момент часу;

B – матриця керування, яка описує, як зовнішні впливи (наприклад, рух керма) впливають на систему;

u_{k-1} – керуюча дія (наприклад, сигнал на керування судном).

Одночасно обчислюється передбачення підступності помилки:

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q, \quad (4.2)$$

де $P_{k|k-1}$ – це матриця коваріації передбачуваної помилки, яка оцінює невизначеність у передбаченому стані, а Q – коваріаційна матриця процесу, що відбиває шум моделі.

Коли надходять нові дані від датчиків (інерційної навігаційної системи), фільтр коригує передбачений стан на основі спостережень:

$$K_k = P_{k|k-1}H^T (HP_{k|k-1}H^T + R)^{-1} \quad (4.3)$$

де K_k – матриця посилення Калмана, яка визначає, наскільки нові дані впливатимуть на коригування передбаченого стану;

H – матриця спостережень, яка пов'язує вимірювання зі станом системи;

R – коваріаційна матриця шуму вимірювань.

Потім оновлюється оцінка стану:

$$x_{k|k} = x_{k|k-1} + K_k(z_k - Hx_{k|k-1}) \quad (4.3)$$

де z_k – це нові дані від датчиків.

Оцінка підступності помилки також коригується:

$$P_{k|k} = (I - K_k H) P_{k|k-1} \quad (4.4)$$

Використовуючи фільтр Калмана, система керування судном може передбачити поточне положення на основі попереднього курсу та швидкості, а потім коригувати його, коли надходять нові дані від навігаційних приладів. Це допомагає уникнути різких змін курсу та покращити плавність маневрування.

4.2 Дослідження роботи моделі симуляції руху судна за заданою траєкторією руху

Проведено розробку моделі руху судна на базі реалізації фільтра Калмана засобами системи Matlab [29]. Для цього було створено програмні скрипти імпорту даних розрахунків створеного програмного застосування з файлу формату *xlsx*, їх обробки, обчислення матриць та реалізації процесу моделювання руху судна у просторі.

Програмний скрипт проведення сесії симуляції наведено нижче.

```
% Simulation of vessel motion using Kalman filter
% Simulation parameters
dt = 0.1; % Time step (sec)
T = 100; % Total simulation time (sec)
% Initial conditions (vessel position and speed)
x_real = [0; 0]; % Current position and speed
x_est = [0; 0]; % State estimate using Kalman filter
% System matrices (vessel motion)
A = [1 dt; 0 1]; % Linear motion model
B = [0.5*dt^2; dt]; % Control action model
H = [1 0]; % Observation matrix (we can measure only position)
% Control action (vessel acceleration, e.g. engine signal)
u = 1; % Constant acceleration
% System and observation noise
Q = [1 0; 0 1] * 0.01; % Process noise covariance
R = 0.1; % Observation noise covariance
% Error covariance initialization
P = eye(2);
% Initialization of arrays for storing results
X_real = zeros(2, T/dt); % True state
X_est = zeros(2, T/dt); % Kalman filter estimate
Z = zeros(1, T/dt); % Observations
% Ship motion simulation
for k = 1:T/dt
    % True ship dynamics
    x_real = A*x_real + B*u + [0.5*randn; randn]*sqrt(Q(1,1)); % Motion noise
    % Observations (e.g. noisy GPS data)
    z = H*x_real + randn*sqrt(R);
    % Kalman filter prediction step
    x_pred = A*x_est + B*u; % Predicted state
    P_pred = A*P*A' + Q; % Predicted covariance
```

```
% Correction based on new observations
K = P_pred*H' / (H*P_pred*H' + R); % Kalman gain
x_est = x_pred + K*(z - H*x_pred); % Update state estimate
P = (eye(2) - K*H)*P_pred; % Update covariance
% Save results
X_real(:,k) = x_real;
X_est(:,k) = x_est;
Z(k) = z;
End
```

Результат запуску скрипту реалізації фільтра Калмана в середовищі розробки Matlab та вихідні логи консолі наведено на рис. 4.1.

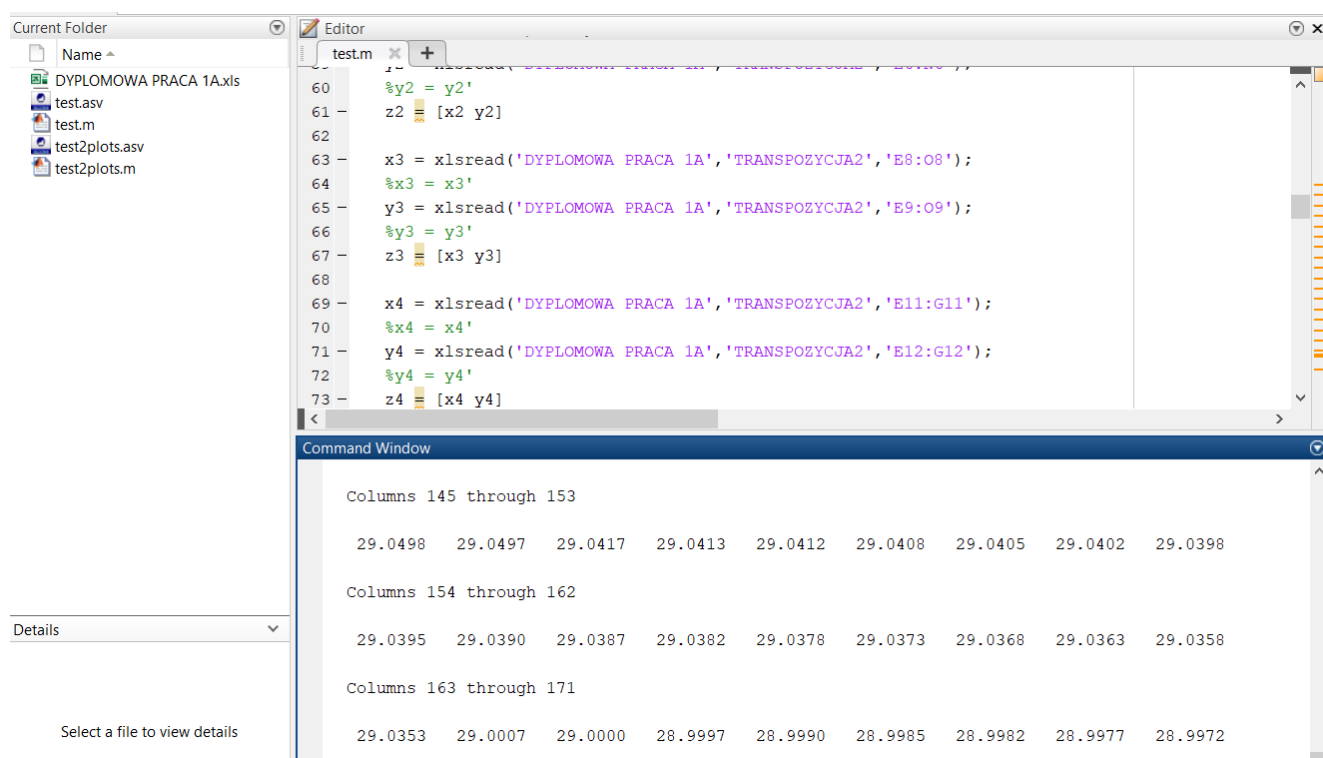


Рисунок 4.1 – Результат запуску скрипту реалізації фільтра Калмана

Масив розрахованих матриць даних для реалізації фільтра Калмана наведено на рис. 4.2. Як можна побачити, дані структуровано за 3 ознаками: x, y, z, кожна з яких відповідає одному з сценаріїв зміни полюсу повороту судна в просторовому стані.

Workspace		Workspace		Workspace	
Name ^	Value	Name ^	Value	Name ^	Value
i	334	y	1x334 double	z1	[41.2527,29.1322]
x	1x334 double	y1	29.1322	z10	1x34 double
x1	41.2527	y10	1x17 double	z11	1x48 double
x10	1x17 double	y11	1x24 double	z12	1x46 double
x11	1x24 double	y12	1x23 double	z13	1x72 double
x12	1x23 double	y13	1x36 double	z14	1x54 double
x13	1x36 double	y14	1x27 double	z15	1x22 double
x14	1x27 double	y15	1x11 double	z16	1x24 double
x15	1x11 double	y16	1x12 double	z17	1x76 double
x16	1x12 double	y17	1x38 double	z2	1x20 double
x17	1x38 double	y2	[29.1292,29.1290,...	z3	1x22 double
x2	[41.2342,41.2338,...	y3	1x11 double	z4	[41.1808,41.1805,...
x3	1x11 double	y4	[29.0818,29.0815,...	z5	1x72 double
x4	[41.1808,41.1805,...	y5	1x36 double	z6	1x72 double
x5	1x36 double	y6	1x36 double	z7	1x42 double
x6	1x36 double	y7	1x21 double	z8	1x38 double
x7	1x21 double	y8	1x19 double	z9	1x18 double
x8	1x19 double	y9	[29.0523,29.0518,...		
x9	[41.0758,41.0753,...				

Рисунок 4.2 – Масив розрахованих матриць даних для реалізації фільтра Калмана

Візуалізація результатів побудови траєкторії руху судна за періодами часу реалізована на базі застосування бібліотеки matplotlib, програмний скрипт наведено нижче.

```
% Plotting
time = 0:dt:T-dt;
figure;
subplot(2,1,1);
plot(time, X_real(1,:), 'b-', 'LineWidth', 2); hold on;
plot(time, Z, 'g.', 'LineWidth', 1.5);
plot(time, X_est(1,:), 'r--', 'LineWidth', 2);
legend('True position', 'Observations (GPS)', 'Kalman filter estimate');
xlabel('Time (s)');
ylabel('Position (m)');
title('Ship position');
```

Проміжні та кінцевий варіанти прокладання траєкторії руху судна наведено на рис. 4.3. Як можна побачити траєкторія руху змінюється через різні параметри розрахунку полюсу-повороту, відбувається автоматичне масштабування зображення руху у координатах, кінцева траєкторія будується на базі з'єднання розрахованих опірних точок.

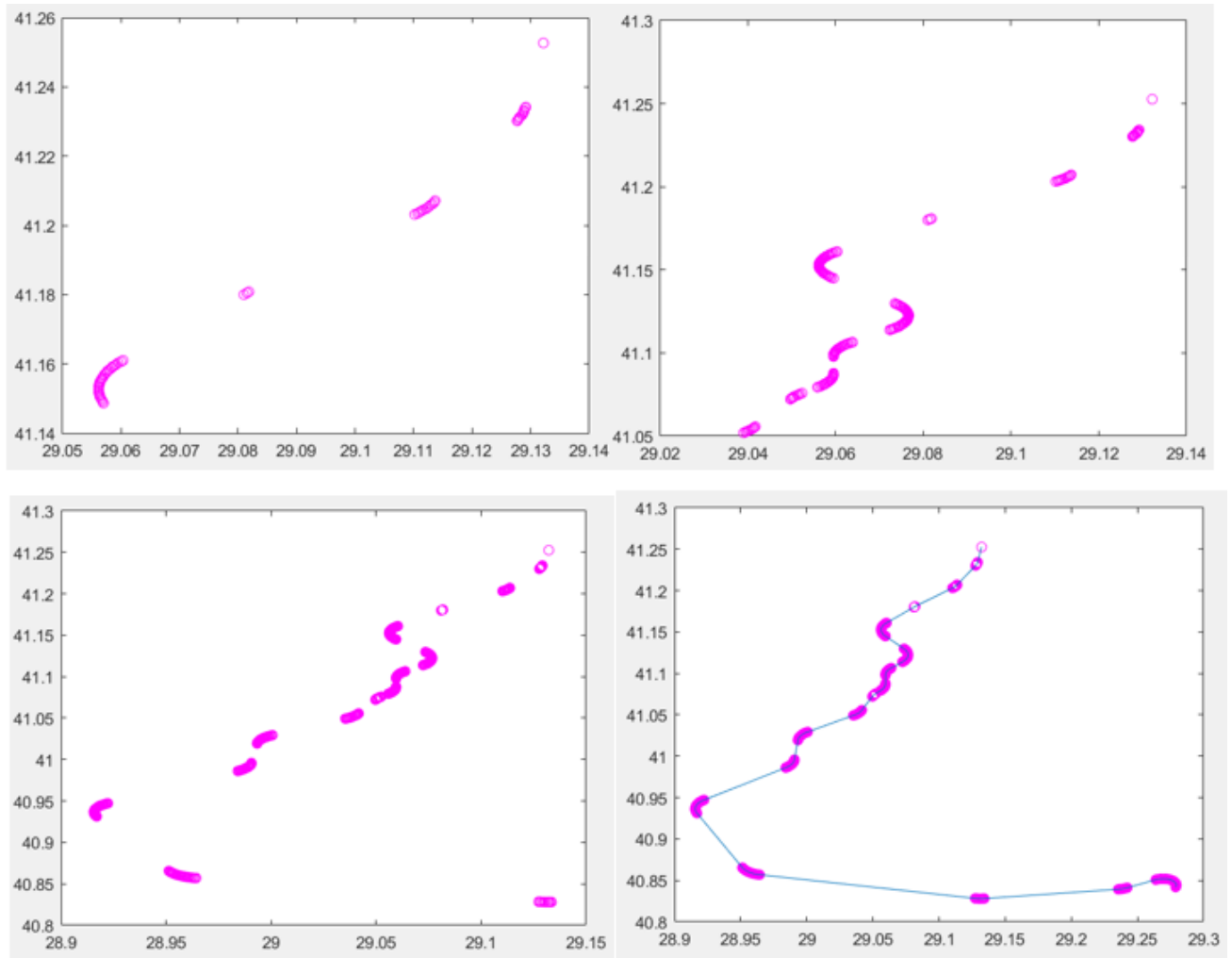


Рисунок 4.3 – Проміжні та кінцевий варіанти прокладання траєкторії руху судна

Для автоматизації процесу дослідження використання фільтру Калмана вирішено побудувати модель симуляції цього процесу на базі застосування бібліотеки візуального моделювання Simulink. Це надає можливості з формування віртуального стенду проведення завдань моделювання без необхідності створення відповідних компонентів інтерфейсу та виконання обчислень засобами звичайних високорівневих мов програмування, що економить час для проведення емпіричних дослідів. Створена модель симуляції процесу побудови траєкторії у Simulink наведена на рис. 4.4.

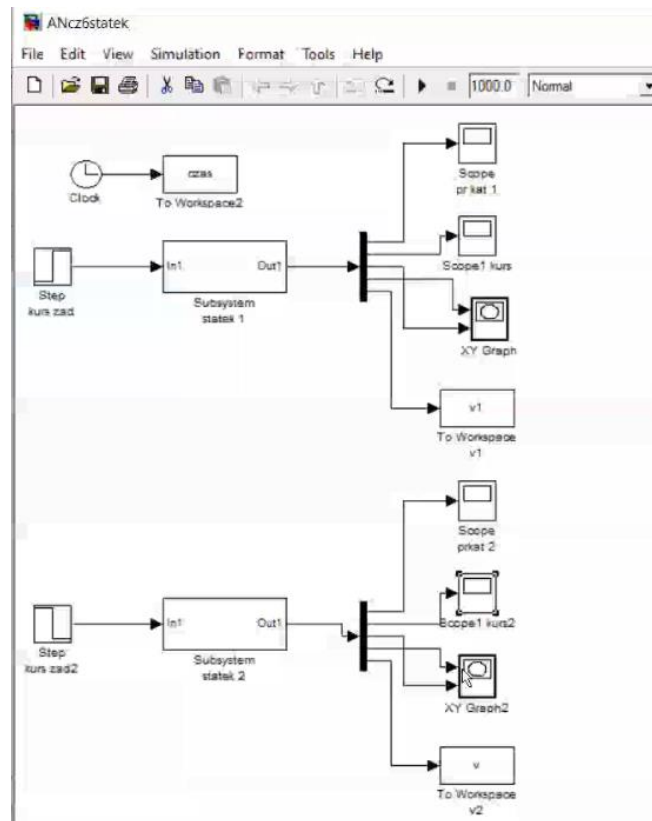


Рисунок 4.4 – Модель симуляції процесу побудови траєкторії у Simulink у загальному вигляді

Модель побудована на базі використання 2 елементів вкладених блоків, для побудови окремих траєкторій руху, їх функціональний склад є ідентичним, змінюються лише вхідні параметри та вивід формованих сигналів (залежностей) здійснюється до окремих елементів Scope.

Склад елементів блоку Subsystem для побудови першої траєкторії руху наведено на рис. 4.5.

Як можна побачити використано генератор імпульсів, який обробляє вхідні дані, робить відповідні перетворення, що було наведено у формулах 4.1 – 4.4, враховуються значення вірогідного куту полюсу повороту судна для перерахунку траєкторії. Отримані дані проходять процедуру агрегації та подаються одним масивом даних до вихідного елементу, що подає їх до відповідного Scope.

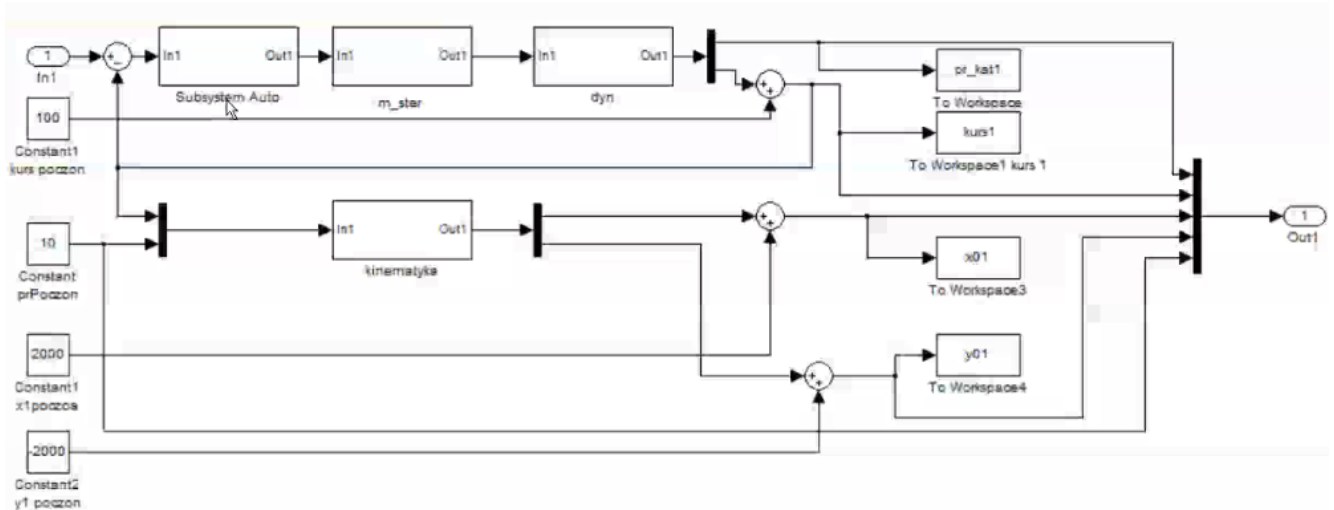


Рисунок 4.5 – Склад елементів блоку Subsystem для побудови першої траєкторії руху

Після запуску моделі симуляції до виконання можна отримати графічне відображення розрахунку значень моделювання процесу розмежування руху суден на різних траєкторіях (рис. 4.6).

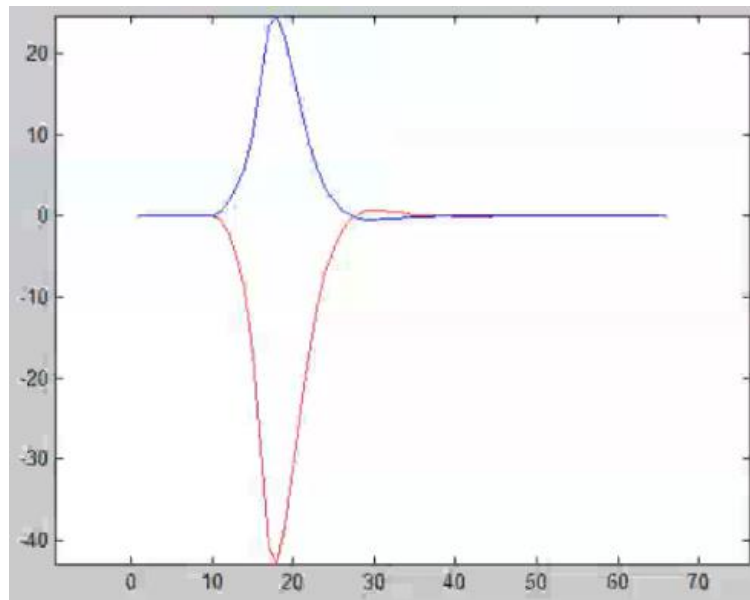
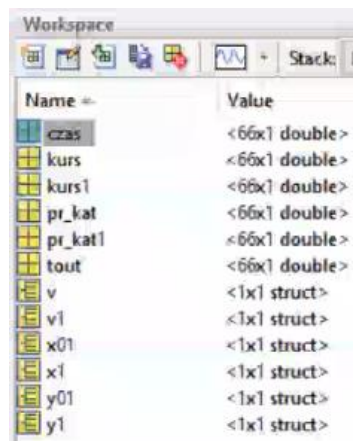


Рисунок 4.6 – Графічне відображення результату виконання моделювання процесу розмежування руху суден на різних траєкторіях

Як можна побачити кориговане значення (нижній графік має більше відхилення на 20 одиниць, що свідчить на більш точне урахування полюсу повороту судна при зміні траєкторії руху у разі маневрування.

В процесі виконання моделювання створюється окремий масив даних, що може бути експортований до окремого датасету чи таблиці. Перелік створених змінних в процесі симуляції наведено на рис. 4.7.



Name	Value
czas	<66x1 double>
kurs	<66x1 double>
kurs1	<66x1 double>
pr_kat	<66x1 double>
pr_kat1	<66x1 double>
tout	<66x1 double>
v	<1x1 struct>
v1	<1x1 struct>
x01	<1x1 struct>
x1	<1x1 struct>
y01	<1x1 struct>
y1	<1x1 struct>

Рисунок 4.7 – Перелік створених змінних в процесі симуляції

У разі необхідності можна перейти до перегляду значень зміни курсу руху кожним судном по відповідній траєкторії шляхом обирання відповідного Scope елементу. Візуальне відображення залежності зміни значень куту полюса повороту судна при зміні параметрів фільтра Калмана наведено на рис. 4.8.

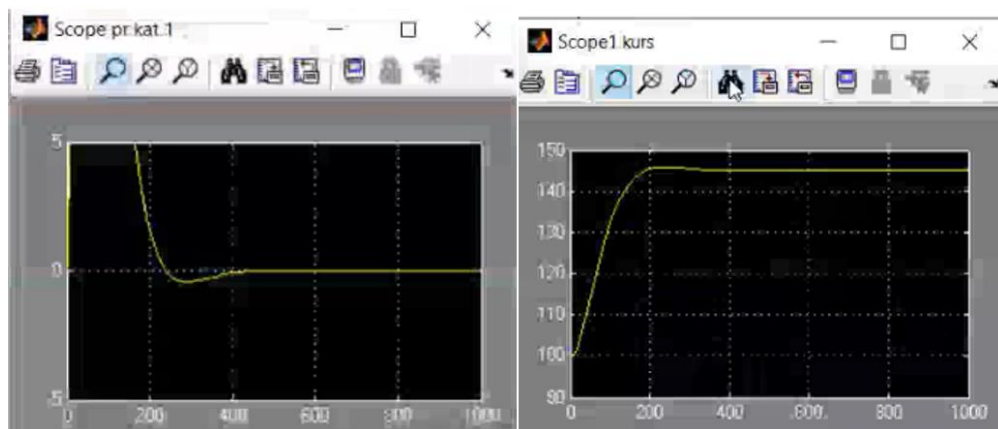


Рисунок 4.8 – Візуальне відображення залежності зміни значень куту полюса повороту судна при зміні параметрів фільтра Калмана

Таким чином в результаті проведених досліджень слід зазначити, що реалізація модуля фільтра Калмана надає оптимальний спосіб цих шумних даних, які виникають в процесі маневрування судном, дозволяючи:

- оцінювати справжній стан судна (становище, швидкість, курс) з урахуванням недетермінованих вимірів;
- передбачати майбутнє положення судна, що особливо важливо для точного розрахунку кута полюса повороту;
- передбачати зміну курсу та швидкість обертання судна, що є критично важливим для коректного обчислення полюса повороту. На основі передбаченої динаміки судна можна точно визначити кут та радіус повороту судна;
- оцінювати поточні параметри руху з мінімальною затримкою, забезпечуючи своєчасне коригування кута повороту при зміні курсу та зовнішніх впливів;
- згладжувати дані про маневр, знижуючи вплив випадкових флуктуацій та помилок у вимірах. Це допомагає уникнути помилкових спрацьовувань при розрахунку полюса повороту, який чутливий до раптових змін даних.

Висновки до розділу 4

В четвертому розділі наведено результати дослідження можливостей моделювання сценаріїв маневрування судном на базі застосування системи Matlab. Виконана формалізація процесу використання фільтра Калмана для дослідження сценаріїв моделювання маневрування, наведено результати дослідження роботи моделі симуляції руху судна за заданою траєкторією руху, описано отримані результати.

Використання фільтра Калмана дозволяє більш точно та стабільно керувати судном, запобігаючи різким відхиленням від курсу та помилкам у маневруванні, особливо у складних погодних умовах або при керуванні великими суднами. Це підвищує безпеку на борту та знижує ймовірність аварійних ситуацій, пов'язаних із неправильним розрахунком кута повороту або раптовими маневрами.

ВИСНОВКИ

В результаті виконання роботи досягнута мета, яка полягала у підвищенні ефективності інформаційної підтримки процесів маневрування судном.

Головним завданням було розробка програмного забезпечення підтримки процесу маневрування морського судна за допомогою розрахунку розташування полюса повороту і візуалізації його переміщення при маневруванні.

Виконано наступні завдання роботи:

- проведено аналіз проблематики керування судном та обґрунтування актуальності роботи, можливостей застосування суднових навігаційних систем, особливостей виконання маневрування судном, обґрунтування переваг використаних засобів програмної розробки, що дозволило визначити напрямки подальшого дослідження тематики;

- здійснена формалізація методологічних положень та алгоритмів з логіки виконання розрахунків, що дозволило скласти алгоритмічну базу для виконання розрахунків;

- проведена розробка проєкту та структури програмного забезпечення, опис ключових методів програмної логіки, що дозволило забезпечити базис для імплементації та тестування програмних модулів;

- виконана розробка ключових функціональних можливостей та опис процесу використання програмного застосування, що дозволило задокументувати роботу застосування у якості керівництва користувача;

- проведено дослідження можливостей використання програмного забезпечення, що дозволило встановити доцільність та ефективність застосування даного програмного забезпечення для завдань моделювання різних сценаріїв маневрування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- 1) Cockcroft A. N., Lameijer J. N. F. A Guide to the Collision Avoidance Rules: International Regulations for Preventing Collisions at Sea. Oxford : Butterworth-Heinemann, 2012. 333 p.
- 2) Salthouse C. The Complete Yachtmaster: Sailing, Seamanship and Navigation for the Modern Yacht Skipper. London : Adlard Coles Nautical, 2006. 566 p.
- 3) DYNAFLOW, Inc. DynaSIM – Ship Traffic Simulator: вебсайт. URL: https://dynaflow-inc.com/Products/Software/DynaSIM/Ship-Traffic-Simulator-DynaSIM.htm?utm_source=chatgpt.com (дата звернення: 08.09.2025).
- 4) MARIN & Deltares. SHIPMA – Fast-Time Ship Manoeuvring Simulator: вебсайт. URL: https://www.marin.nl/en/about/facilities-and-tools/software/shipma?utm_source=chatgpt.com (дата звернення: 08.09.2025).
- 5) Вагущенко Л. Л. Суднові навігаційно-інформаційні системи. Одеса : НУ «ОМА», 2016. 238 с.
- 6) Голиков В. В. Алгоритм визначення положення полюса повороту морського судна. Херсон : Видавництво ХДМА, 2013. С. 21–27.
- 7) Товстокорий О. М. Визначення положення полюса повороту за допомогою доплерівського. Одеса : ИздатІнформ, 2016. С. 183–190.
- 8) Hughes E. C. Shiphandling for the Mariner. Atglen : Schiffer Publishing, Ltd, 2018. 224 p.
- 9) Мальцев А. С. Маневрування судів під час розбіжності. Одеса : Морський тренажерний центр, 2002. 208 с.
- 10) Вагущенко Л. Л. Автоматизовані комплекси судноводіння. Київ : КВІЦ, 2000. 292 с.
- 11) Гофман В. Л. Глобальна система позиціонування GPS. Теорія та практика. Київ : Наукова думка, 1996. 314 с.
- 12) Eryilmaz K. Principles of Maritime Navigation. Cham : Springer, 2020. 278 p.
- 13) Van Rossum G., Drake F. L. The Python Language Reference. Python Software Foundation, 2023. 242 p.

- 14) IDE PyCharm: вебсайт. URL: <https://www.jetbrains.com/pycharm/> (дата звернення: 12.09.2025).
- 15) Qt Company. QML Applications: вебсайт. URL: <https://doc.qt.io/qt-6/qmlapplications.html> (дата звернення: 13.09.2025).
- 16) Qt Company. QML – Qt Modeling Language: вебсайт. URL: <https://doc.qt.io/qt-6/qmlreference.html> (дата звернення: 13.09.2025).
- 17) House D. J. Marine Navigation and Safety of Sea Transportation. London : Routledge, 2020. 650 p.
- 18) Fossen T. I. Handbook of Marine Craft Hydrodynamics and Motion Control. 2nd ed. Hoboken : Wiley-IEEE Press, 2021. 600 p.
- 19) Bhatia S. Electronic Navigation Systems: Principles and Applications. New York : CRC Press, 2022. 312 p.
- 20) Zhang P., Zhao Y., Zhu J. Ship Trajectory Prediction Based on AIS Data and Deep Learning. Journal of Navigation. 2022. Vol. 75, Iss. 2. P. 345–362.
- 21) Norris A. ECDIS and Positioning. London : The Nautical Institute, 2020. 210 p.
- 22) Molland A. F., Turnock S. R., Hudson D. A. Ship Resistance and Propulsion: Practical Estimation of Ship Propulsive Power. Cambridge : Cambridge University Press, 2022. 560 p.
- 23) Artyszuk J. Ship Manoeuvring in Shallow and Confined Waters. Sensors. 2021. Vol. 21, № 5. P. 1–24.
- 24) Yuan Y., Li K., Zhang W. Integrated INS/GNSS Navigation for Marine Applications. Ocean Engineering. 2021. Vol. 221. Article 108543.
- 25) Liu C., Wang X. Estimation of Ship Pivot Point Position Based on Real-Time Maneuvering Data. Applied Ocean Research. 2023. Vol. 130. Article 103421.
- 26) Yan X., Wang S., Ma F. Modeling of Wind and Current Effects on Ship Maneuverability. Journal of Marine Science and Technology. 2021. Vol. 26. P. 112–125.
- 27) Chen H., Liu J. Correction of GNSS Antenna Position Offset for Accurate Ship Center of Gravity Tracking. IEEE Access. 2022. Vol. 10. P. 45012–45020.

28) Grewal M. S., Andrews A. P. Kalman Filtering: Theory and Practice with MATLAB. 5th ed. Hoboken : Wiley, 2020. 640 p.

29) Li Z., Wu B. Simulation of Ship Maneuvering Motion Using MATLAB/Simulink. International Journal of Naval Architecture and Ocean Engineering. 2021. Vol. 13. P. 45–56.

30) Simpson P. Bridge Team Management: A Practical Guide. 2nd ed. London : The Nautical Institute, 2023. 180 p.

ДОДАТОК А (Main.pyw)

```
from PyQt5.QtGui import QApplication, QIcon
from PyQt5.QtQml import QQmlApplicationEngine
import sys
```

```
app = QApplication(sys.argv)
engine = QQmlApplicationEngine()
app.setWindowIcon(QIcon("Icon.png"))
engine.load('main.qml')
engine.quit.connect(app.quit)
sys.exit(app.exec_())
```

```
import QtQuick 2.5
import QtQuick.Controls 1.5
import QtQuick.Layouts 1.2
import QtQuick.Controls.Styles 1.4
import QtQuick.Window 2.2
```

```
Window{
    id: anotherWindow
    signal signalExit
    modality: Qt.ApplicationModal
    width:500
    height:650
    maximumHeight: 600
    minimumHeight:600
    maximumWidth: 500
    minimumWidth:500
    Rectangle {
        id: rectangle
```

```
color: "#d0d0d0"  
anchors.rightMargin: 0  
anchors.bottomMargin: 0  
anchors.leftMargin: 0  
anchors.topMargin: 0  
anchors.fill: parent
```

```
TabView {  
    id: tabView  
    currentIndex: 0  
    anchors.rightMargin: 0  
    anchors.bottomMargin: 0  
    anchors.leftMargin: 0  
    anchors.topMargin: 0  
    anchors.fill: parent
```

```
Tab {  
    title: "Результати дослідження"  
    source: "Результати_дослідження.qml"  
}
```

```
Tab {  
    id: ____  
    title: "Алгоритм програмної реалізації"  
    source: "Алгоритм_програмної_реалізації.qml"  
}
```

```
Tab {  
    id: _
```

```
        title: "Загальна інформація"
        source: " Загальна_інформація.qml"
    }
}
}
}

import QtQuick 2.15

Item {
    id: shipComponent
    property var maneuverData: null

    function drawShip(ctx, data) {
        ctx.save()

        // Встановлення орієнтації судна
        ctx.translate(data.position.x, data.position.y)
        ctx.rotate(data.course * Math.PI / 180)

        // Малюємо корпус судна
        ctx.beginPath()
        ctx.moveTo(-20, -60)
        ctx.lineTo(20, -60)
        ctx.lineTo(30, 0)
        ctx.lineTo(20, 60)
        ctx.lineTo(-20, 60)
        ctx.lineTo(-30, 0)
        ctx.closePath()
        ctx.fillStyle = "blue"
```

```
ctx.fill()

// Поліус повороту
ctx.beginPath()
ctx.arc(data.pivotPoint.x, data.pivotPoint.y, 5, 0, 2 * Math.PI)
ctx.fillStyle = "red"
ctx.fill()

ctx.restore()
}
}

import QtQuick 2.15
import QtQuick.Controls 2.15

ApplicationWindow {
    visible: true
    width: 1024
    height: 768
    title: "Система маневрування судна"

    // Основна сцена для відображення процесів маневрування
    Canvas {
        id: maneuverCanvas
        anchors.fill: parent
        onPaint: {
            var ctx = getContext("2d")
            ctx.clearRect(0, 0, width, height)
```

```
// Виклик функції для малювання судна
shipComponent.drawShip(ctx, maneuverData)
}
}

// Панель налаштувань і параметрів
Column {
    anchors.right: parent.right
    spacing: 10
    padding: 20

    Text {
        text: "Швидкість: " + maneuverData.speed + " вузлів"
    }

    Text {
        text: "Курс: " + maneuverData.course + "°"
    }

    Slider {
        id: speedSlider
        from: 0
        to: 30
        value: maneuverData.speed
        onValueChanged: maneuverData.speed = value
    }

    Slider {
        id: rudderSlider
```

```
        from: -30
        to: 30
        value: maneuverData.rudderAngle
        onValueChanged: maneuverData.rudderAngle = value
    }

    Button {
        text: "Почати маневр"
        onClicked: {
            maneuverData.startManeuver()
        }
    }
}

QObject {
    id: maneuverData
    property real speed: 10 // Швидкість судна
    property real course: 90 // Курс судна
    property real rudderAngle: 0 // Кут повороту руля
    property var position: { "x": 512, "y": 384 } // Початкове положення
    property var pivotPoint: { "x": 512, "y": 300 } // Полюс повороту

    function startManeuver() {
        // Простий алгоритм для симуляції зміни позиції судна
        position.x += Math.cos(course * Math.PI / 180) * speed
        position.y += Math.sin(course * Math.PI / 180) * speed

        // Розрахунок полюса повороту
        pivotPoint.x = position.x + 100 * Math.cos((course + 90) * Math.PI / 180)
```

```
pivotPoint.y = position.y + 100 * Math.sin((course + 90) * Math.PI / 180)

// Оновлення Canvas
maneuverCanvas.requestPaint()
}
}
```