

УДК 004.04:629.5.072.1

DOI: 10.31673/2412-9070.2023.049000

Т. В. ПИРОГОВСЬКА, аспірантка;

Т. П. ДОВЖЕНКО, канд. техн. наук,

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ

## РОЗРОБЛЕННЯ КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ ВВЕДЕННЯ КРИВОЛІНІЙНОЇ ТРАЄКТОРІЇ РУХУ ОБ'ЄКТА ЗІ ЗМІННОЮ ШВИДКІСТЮ

**Розглянуто розроблення інтерфейсу для максимально точного введення користувачем траєкторії криволінійного руху корабля в морському середовищі зі змінною швидкістю для подальшого вдосконалення програмного забезпечення моделювання морської акваторії. З огляду на відомості щодо фізичних сил, які впливають на рух корабля, було виведено формули розрахунку точок траєкторії, які в подальшому використовуватимуться як для візуального зображення, так і для розрахунків у моделюванні експерименту. Наведено приклад кодування виведених формул, а також продемонстровано розроблений інтерфейс.**

**Ключові слова:** інтерактивний інтерфейс; моделювання руху об'єкта; криволінійний рух зі змінною швидкістю; траєкторія руху корабля.

### ВСТУП

**Постановка проблеми.** Зазвичай зручне моделювання об'єктів морської акваторії потребує зручного інтерфейсу, який дає змогу користувачеві не лише задати довільну криволінійну траєкторію руху, а й перевіряє можливість руху заданого об'єкта відносно його розмірів та швидкості пересування. Тому для реалізації інтерактивного інтерфейсу введення криволінійної траєкторії корабля варто зважати на динаміку контрольованого об'єкта. У процесі оцінювання траєкторії мають бути задоволені фізичні обмеження судна, які визначають мінімальний радіус кривизни, дотичне прискорення та всі три швидкості (поздовжню, поперечну, кутову). Спроектowana траєкторія має бути плавною, а швидкості судна за всіма трьома осями — постійними.

Особливої уваги заслуговують повороти, які краще виконувати завчасно, до підходу до навігаційних небезпек, а саме:

- не потрібно здійснювати різких поворотів поблизу небезпек або входити у вузький канал на циркуляції;
- входити в підхідний канал порту треба завчасно, мористіше приймального буя і під гострим кутом;
- якщо після приймання лоцмана або зняття з якоря вхід у підхідний канал можливий тільки на крутій циркуляції, то краще відійти мористіше або виконати циркуляцію в бік моря.

**Аналіз останніх досліджень і публікацій.** У статті [1] описано один зі способів прогнозування траєкторії судна із застосуванням штучних нейронних мереж. Вони становлять універсальну систему апроксимації, відображаючи багатовимірні набори даних, а також мають здатність до навчання й адаптації до мінливих умов середовища. Штучні нейронні мережі здатні до узагальнення в різних галузях, включно з прогнозуванням часових рядів та ідентифікацією об'єктів керування. Алгоритм використовує узагальнену регресійну нейронну мережу GRNN.

Багато науковців у своїх працях використовують поліноміальні криві Безьє 3-го та вищих порядків для оцінювання шляху надводних суден. Хассані та Ланде [2] запропонували створити набір можливих шляхів і призначити функцію вартості кожному з них. Вони використовували властивість диференціальної площинності транспортного засобу, щоб реконструювати всі стани суден під час маневру на обмежених водних шляхах. Ними було застосовано криві Безьє 5-го порядку та розглянуто контрольні точки як змінні проекту під час процесу оптимізації. Янг і Чой [3] використовували кубічну криву Безьє, щоб зробити траєкторію S-кривою. Чой та ін. [4] взяв криві Безьє як вихідну функцію для алгоритму планування шляху як альтернативу кубічним сплайнам. Цей підхід, що охоплює нелінійні завдання оптимізації, дає змогу уникнути зіткнень.

Потрібно зауважити, що криві Безьє використовуються і для прогнозування майбутньої траєкторії корабля з метою контролю. Так, у [5] автори апроксимують траєкторію морського автономного надводного судна за допомогою раціональної кривої Безьє найнижчого порядку. Цей підхід дає змогу швидко передбачати майбутню траєкторію та потребує невеликої обчислювальної потужності. Застосування кривої траєкторії замість набору прямих відрізків поліпшує якість керування завдяки вильотам на поворотах і коливанням у відстеженні траєкторії після усунення повороту.

© Т. В. Пироговська, Т. П. Довженко, 2023

У статті [6] проаналізовано метод керування траєкторією судна під час повороту в замкнених водних шляхах. Результати досліджень показали, що значно меншого відхилення в повороті від окресленого шляху судна можна досягти в разі дотримання таких умов: знання маневреності судна, добре проаналізована навігація площини, ретельний план проходу для замкнених водних шляхів із точно визначеним радіусом повороту, нового курсу.

**Метою статті** є створення зручного користувацького інтерфейсу, який матиме змогу задавати довільну криволінійну траєкторію руху та одночасно перевіряти можливість пересування цього об'єкта відносно його розмірів та швидкості пересування. Здобуті точки мають використовуватись як і для візуального відображення, так і для обчислень моделювання середовища.

### ОСНОВНА ЧАСТИНА

Під час руху морського об'єкта, наприклад судна, на його шляху можуть зустрічатися різні перешкоди, наприклад, близькість берегу, мілина, острів, інші морські об'єкти тощо. Тому його рух може бути обмежено у маневрі. Для забезпечення навігаційної безпеки для суден програмується траєкторія з урахуванням усіх перешкод. Найбільш важливим у процесі моделювання траєкторії руху судна в морі є поворот. Це найпоширеніший маневр, яке судно може виконувати. Траєкторія руху судна зі зміною курсів може набирати різної форми залежно від критерію, закладеного в основу оптимізації повороту: час повороту, кут повороту керма, момент інерції судна. Процес повороту можна поділити на три фази (рис. 1).

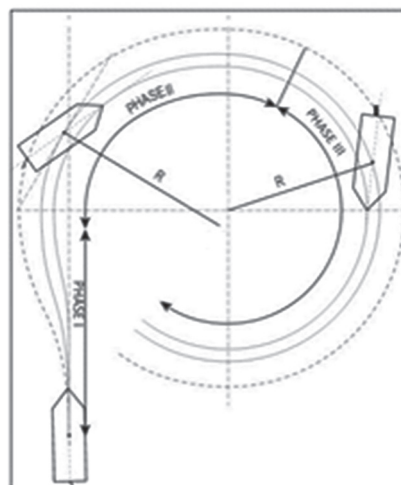


Рис. 1. Фази повороту судна

**Фаза 1.** Судно спочатку пливе по певній прямій лінії без будь-якого обертального руху. Через відсутність будь-яких поперечних сил і моментів швидкість «розгойдування» (рух з боку в бік або лівого правого борту) і обертання вважаються нульовими. Після повороту керма на деякий кут у поперечному напрямку з'являються деякі сили та моменти, які посилюються взаємодією гідродинамічних факторів. Після повороту кермо створює певну силу на правому борті. У відповідь на цю силу гідродинамічна сила створює рівну і протилежну силу реакції на лівому борті. Оскільки інші сили, що виникають у корпусі, відсутні під час цієї фази, то судно має тенденцію дрейфувати боком до лівого краю. Коли сили приходять у рівновагу, судно знову робить поворот правим бортом.

**Фаза 2.** Судно змінює курс на  $90^\circ$  від свого початкового курсу. Момент, спричинений корпусом, набуває певної переваги завдяки збільшенню кута дрейфу, викликаного моментом керма. Через деякий час момент керма врівноважує момент корпусу. На другій фазі не всі кутові та лінійні швидкості і прискорення дорівнюють нулю.

**Фаза 3.** Оскільки кермо постійно відхиляється, судно продовжує обертатися по колу. У третій фазі момент корпусу перевищує момент, створений кермом, і судно досягає стійкого стану повороту. Кут і лінійна швидкості залишаються постійними з обертанням судна по колу постійного радіуса. Оскільки всі зовнішні впливи зменшуються, параметри прискорення стають нульовими. На судно продовжує діяти відцентрова сила, і воно обертатиметься далі за тією самою траєкторією, якщо кермо не повернути в середнє положення.

Параметри, які оцінюються під час повороту, такі (рис. 2):

- **просування (Advance)** — це відстань, на яку судно переміщується вперед після повороту керма на деякий кут. Судно виявляє всі гідродинамічні ефекти, пов'язані з ним, здійснюючи будь-яку зміну свого руху. Судно не може відразу повернути після застосування відхилення керма, тому, пройшовши деяку відстань, воно робить це посту-

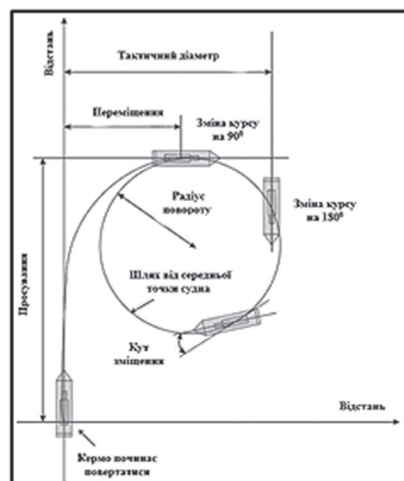


Рис. 2. Процес повороту корабля

пово. Однак поточні засоби керування судном у морі потребують швидшої реакції на наказ оператора. Тому завжди шукають мінімальний критерій просування;

- **переміщення (Transfer)** — це поперечна відстань, яку пройшло судно під час зміни його курсу на  $90^\circ$ . Це відстань між початковим вектором напрямку судна та точкою, коли корабель повністю попрямував правим бортом у своїй другій фазі повороту;

- **тактичний діаметр (Tactical Diameter)** — це показник ступеня розвороту судна. Вимірюється як відстань між початковим вектором напрямку судна за постійного курсу та вектором на завершальній фазі його повороту, коли досягається постійний стан, тобто судно змінило свій курс на  $180^\circ$ . Це відстань, пройдена під час повного розвороту з постійним відхиленням керма, визначається цією довжиною. Оскільки відстань майже дорівнює геометричному діаметру його шляху, її називають тактичним діаметром.

Тактичний діаметр має бути щонайбільше менш ніж у п'ять разів довжини судна будь-якого загального призначення:

$$T_d < 5 \cdot \text{довжина судна.}$$

Для будь-якого судна загального призначення попереднє просування має бути менш ніж у 4,5 раза більше за довжину судна;

$$A_d < 4,5 \cdot \text{довжина судна.}$$

Загалом, чим довшим є судно, тим більший радіус повороту. Тип керма та кінцевий ефект кермування визначатимуть кінцевий діаметр, причому просвіт між кермом і корпусом матиме великий вплив. Чим менший зазор між кермом і корпусом, тим ефективніший поворот.

У сучасних суднах повороти виконуються автоматично із заданими параметрами: кутвою швидкістю та радіусом повороту, де поворот із заданим радіусом зводиться до задачі із заданою кутвою швидкістю. Розглядатимемо рух судна за умов, що вітер, хвилювання та течія малі і їхній вплив на траєкторію незначний.

Фактори, що впливають на поворот:

- конструкція та довжина судна;
- осадка та диферент судна;
- розмір та потужність суднової силової установки;
- кут перекидання керма;
- наявна глибина моря.

Судно з диферентом на корму буде краще слухатися керма, але тактичний діаметр циркуляції збільшиться.

Диферент на ніс зменшить діаметр циркуляції, але стернувати судно буде важче.

Судно з креном буде мати затримки з поворотом. Більший діаметр повороту буде тоді, коли повертати будемо в бік крену.

Критерії оптимізації повороту судна такі:

- оптимальний кут зміни керма  $10^\circ$ - $15^\circ$ , а максимальний —  $35^\circ$ ;
- траєкторія повороту має форму дуги, якщо кут відхилення пера керма постійний;
- момент інерції залежить від маси судна, лінійної швидкості та гідрометеоумов.

Радіус циркуляції під час повороту судна можна записати у вигляді

$$R = L/\omega, \quad (1)$$

де  $\omega$  — кутова швидкість;  $L$  — довжина судна.

Найменший радіус відповідає найбільшому куту перекидання керма. У разі великих кутів коригувати рух судна на циркуляції важко, а за малих кутів збільшується час повороту і, відповідно, зростає і похибка його виконання.

### Поворот судна з постійним радіусом

Кутова швидкість обчислюється за формулою (рис. 3):

$$\omega = \frac{57,3}{60} \cdot \frac{V}{R}, \quad (2)$$

де  $V$  — швидкість судна відносно ґрунту, вузол;  $R$  — радіус повороту, морська миля.

Відстань від точки перекидання керма до точки, де поворот стає ефективним, приблизно дорівнює одній довжині судна.

Відстань за перпендикуляром від точки перекидання керма до продовження нового курсу визначається за виразом

$$P = F \cdot \sin\theta + R(1 - \cos\theta), \quad (3)$$

де  $R$  — радіус повороту;  $F$  — просування (відстань, яку проходить судно після перекидання керма до початку повороту, тобто відстань, яку судно долає за інерцією);  $\theta$  — кут зміни курсу ( $D + P$  — відстань,

на якій треба прокласти паралельний індекс від точки перекладання керма;  $D$  — відстань, на якій треба прокласти паралельний індекс від нового курсу, яку отримують з карт).

Відшукання точки перекладання керма передбачає такі етапи:

- 1)  $F = 0,1$  to  $0,15 NM$ ;
- 2)  $P = R(1 - \cos\theta) + F \sin\theta$ ;
- 3)  $\omega = 0,96 \cdot V/R$ , де  $V$  — швидкість судна;
- 4) відстань до нового курсу:  $R \operatorname{tg} \theta/2$ ;
- 5) точка перекладання керма:  $F +$  відстань до нового курсу.

**Алгоритм визначення траєкторії за трьома відомими точками**

З огляду на наведені раніше теоретичні відомості, цю задачу можна спростити до задачі пошуку дуги кола, вписаного в кут, утворений трьома точками траєкторії, беручи до уваги швидкість пересування об'єкта.

**Крок 1.** Визначимо центр кола за трьома точками  $A, B, C$  (рис. 4).

Для цього запишемо загальні рівняння прямих  $AB$  та  $BC$ :

$$AB: A_1x + B_1y + C_1 = 0, \text{ де } \begin{cases} A_1 = y_a - y_b, \\ B_1 = x_b - x_a, \\ C_1 = x_a \cdot y_b - y_a \cdot x_a. \end{cases}$$

$$BC: A_2x + B_2y + C_2 = 0, \text{ де } \begin{cases} A_2 = y_b - y_c, \\ B_2 = x_c - x_b, \\ C_2 = x_b \cdot y_c - y_b \cdot x_c. \end{cases}$$

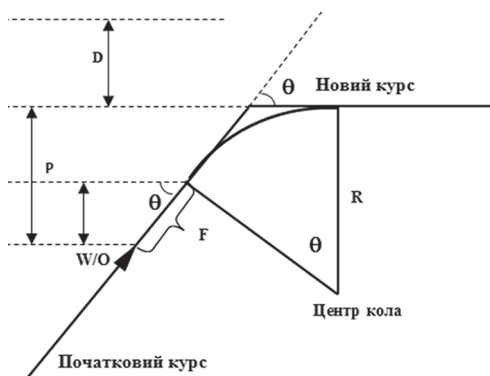


Рис. 3. Поворот судна з постійним радіусом



Рис. 4. Перший крок алгоритму: знаходження центра кола

Далі побудуємо серединні перпендикуляри до  $AB$  та  $BC$  у такий спосіб:

$$I_{AB} = R[i] \sqrt{A_1^2 + B_1^2 + C_1^2},$$

$$I_{BC} = R[i] \sqrt{A_2^2 + B_2^2 + C_2^2},$$

де  $R[i]$  — тактичний радіус, який обчислюється для кожної точки траєкторії,  $R[i] = 10 \cdot v$  (тут  $v$  — швидкість об'єкта).

Центр кола  $O(x_o, y_o)$  визначатиметься за системою рівнянь:

$$\begin{cases} x_o = \frac{l_1 - B_1 \cdot y_p}{A_1}, \\ y_o = \frac{l_2 - l_1 \cdot \frac{A_2}{A_1}}{B_2 - B_1 \cdot \frac{A_2}{A_1}}. \end{cases}$$

Частинні випадки обчислюватимуться за виразами

- 1) якщо  $A_1 = 0$ , то  $\begin{cases} x_o = \frac{l_2 - B_2 \cdot y_p}{A_2}, \\ y_o = \frac{l_1}{B_1}; \end{cases}$
- 2) якщо  $B_1 = 0$ , то  $\begin{cases} x_o = \frac{l_1}{A_1}, \\ y_o = \frac{l_1 - A_2 \cdot x_p}{B_2}; \end{cases}$
- 3) якщо  $A_2 = 0$ , то  $\begin{cases} x_o = \frac{l_1 - B_1 \cdot y_p}{A_1}, \\ y_o = \frac{l_2}{B_2}; \end{cases}$
- 4) якщо  $B_2 = 0$ , то  $\begin{cases} x_o = \frac{l_2}{A_2}, \\ y_o = \frac{l_1 - A_1 \cdot x_p}{B_1}. \end{cases}$

**Крок 2.** Запишемо загальні рівняння серединних перпендикулярів  $l_1$  та  $l_2$  до прямих  $AB$  та  $BC$  (рис. 5):

$$OD: A_3x + B_3y + C_3 = 0, \text{ де } \begin{cases} A_3 = B_1, \\ B_3 = -A_1, \\ C_3 = -A_1 \cdot y_o - B_1 \cdot x_o, \end{cases} \quad OE: A_4x + B_4y + C_4 = 0, \text{ де } \begin{cases} A_4 = B_2, \\ B_4 = -A_2, \\ C_4 = -A_2 \cdot y_o - B_2 \cdot x_o. \end{cases}$$

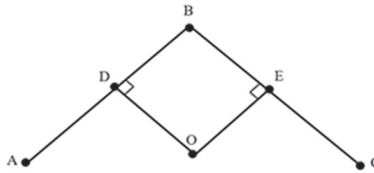


Рис. 5. Другий крок: побудова перпендикулярів із центра кола до прямих кута

Потім визначимо точки початку та кінця повороту — точки перетину кола та прямих кута:

$$CL_1: A_5x + B_5y + C_5 = 0, \text{ де } \begin{cases} A_5 = A_1, \\ B_5 = B_1, \\ C_5 = -C_1, \end{cases} \quad CL_2: A_6x + B_6y + C_6 = 0, \text{ де } \begin{cases} A_6 = A_3, \\ B_6 = B_3, \\ C_6 = C_3, \end{cases} \quad \begin{cases} x = \frac{C_5 - B_5 \cdot y}{A_5}, \\ y = \frac{C_6 - C_5 \cdot \frac{A_6}{A_5}}{B_6 - B_5 \cdot \frac{A_6}{A_5}}. \end{cases}$$

Частинні випадки:

$$\begin{aligned} 1) \text{ якщо } A_5 = 0, \text{ то } \begin{cases} x = \frac{C_6 - B_6 \cdot y}{A_6}, \\ y = \frac{C_5}{B_5}; \end{cases} & \quad 3) \text{ якщо } A_6 = 0, \text{ то } \begin{cases} x = \frac{C_6 - B_5 \cdot y}{A_5}, \\ y = \frac{C_6}{A_6}; \end{cases} \\ 2) \text{ якщо } B_5 = 0, \text{ то } \begin{cases} x = \frac{C_5}{A_6}, \\ y = \frac{C_6 - A_6 \cdot x}{B_6}; \end{cases} & \quad 4) \text{ якщо } B_6 = 0, \text{ то } \begin{cases} x = \frac{C_6}{A_6}, \\ y = \frac{C_5 - A_5 \cdot x}{B_5}. \end{cases} \end{aligned}$$

**Крок 3.** Далі будемо траєкторію руху об'єкта (рис. 6):

- пряма до точки початку дуги  $AP_1$ :

$$\rho(A, P_1) = \rho_{A, P_1} = \sqrt{(x_{p1} - x_a)^2 + (y_{p1} - y_a)^2};$$

- поступовий рух по прямій із появою нових точок:

$$\begin{cases} x = v \cdot \frac{x_{p1} - x_a}{\rho_{A, P_1}}, \\ y = v \cdot \frac{y_{p1} - x_a}{\rho_{A, P_1}}; \end{cases}$$

- рух по дузі

$$\begin{cases} x = x_o + R + \cos(\angle AP_1O), \\ y = y_o + R + \sin(\angle AP_1O) \end{cases}, \text{ де } \angle AP_1O = \text{atng}(P_1 - O).$$

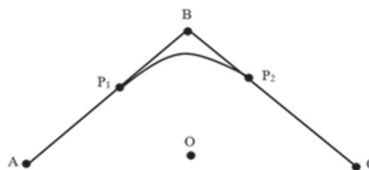


Рис. 6. Крок третій: побудова дуги кола з огляду на швидкість руху

**Програмування описаних розрахунків**

Перед обрахунком точок у коді обчислюються масиви потрібних параметрів, а саме: масив точок центра повороту, рівняння перпендикулярів для обчислення точок початку та кінця повороту, масив тактичних радіусів з огляду на швидкість повороту судна. Перед початком будь-яких обрахунків масив точок, наданий користувачем, перевіряється за умови побудови поворотів, зазначених у розрахунках (рис. 7 – рис. 9).

```

Point3D P = new Point3D( ) { X = 0, Y = 0, Z = 0 };

// Перевірка чи три точки належать одній прямій, і знаходяться по осі
if (a.X == b.X && a.Y == b.Y)
    return P;
if (b.X == c.X && b.Y == c.Y)
    return P;

if (a.X == b.X && a.X == c.X)
    return P;
if (a.Y == b.Y && a.Y == c.Y)
    return P;

LineEquation ab = new LineEquation
{
    A = (float)a.Y - (float)b.Y,
    B = (float)b.X - (float)a.X,
    C = (float)a.X * (float)a.Y - (float)a.Y * (float)b.X
};
LineEquation bc = new LineEquation
{
    A = (float)b.Y - (float)c.Y,
    B = (float)c.X - (float)b.X,
    C = (float)b.X * (float)c.Y - (float)b.Y * (float)c.X
};
    
```

```

// Перевірка чи три точки належать одній прямій
// Line Bisector abA * X + abB * Y =
float hFromAB = ((float)(1) * Math.Sqrt(Math.Pow(ab.A, 2) + Math.Pow(ab.B, 2)) - ab.C);
// Line Bisector bcb * X + bcb * Y =
float hFromBC = ((float)(1) * Math.Sqrt(Math.Pow(bc.A, 2) + Math.Pow(bc.B, 2)) - bc.C);

if (ab.A == 0 && bc.A == 0)
    return P;
else
if (ab.B == 0 && bc.B == 0)
    return P;
else
if (ab.A == 0)
{
    P.Y = ((float)(hFromAB / ab.B));
    P.X = ((float)((hFromBC - bc.B * P.Y) / bc.A));
}
else
if (ab.B == 0)
{
    P.X = ((float)(hFromAB / ab.A));
    P.Y = ((float)((hFromBC - bc.A * P.X) / bc.B));
}
else
if (bc.A == 0)
{
    P.Y = ((float)(hFromBC / bc.B));
    P.X = ((float)((hFromAB - ab.B * P.Y) / ab.A));
}
else
if (bc.B == 0)
{
    P.X = ((float)(hFromBC / bc.A));
    P.Y = ((float)((hFromAB - ab.A * P.X) / ab.B));
}
else
{
    P.Y = ((float)((hFromBC - hFromAB * bc.A / ab.A) / (bc.B - ab.B * bc.A / ab.A));
    P.X = ((float)((hFromAB - ab.B * P.Y) / ab.A));
}

ab.C = ((float)(Math.Pow(a.X - P.X, 2) + Math.Pow(a.Y - P.Y, 2)));
bc.C = ((float)(Math.Pow(b.X - P.X, 2) + Math.Pow(b.Y - P.Y, 2)));
    
```

Рис. 7. Програмний код знаходження центра кола

```

dummySQRT = Math.Sqrt(
    Math.Pow(dummyCross[2 * i + 2].X - dummyCross[2 * i + 1].X, 2) +
    Math.Pow(dummyCross[2 * i + 2].Y - dummyCross[2 * i + 1].Y, 2));
dx = cp.W * (dummyCross[2 * i + 2].X - dummyCross[2 * i + 1].X) / dummySQRT;
dy = cp.W * (dummyCross[2 * i + 2].Y - dummyCross[2 * i + 1].Y) / dummySQRT;

deltaN = 0;
do
{
    cp.X += dx;
    cp.Y += dy;
    deltaN++;
    if (++n > MAX_POINTS)
        goto EndC;
}
while (!IsLastPointOnLine(cp, dummyCross[2 * i + 2], (int)cp.W));
    
```

Рис. 8. Програмний код знаходження точок руху до сегмента з дугою

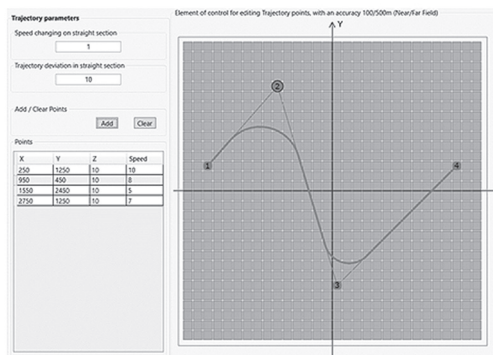


Рис. 10. Приклад інтерфейсу користувача

```

for (int i = 0; i < dummyCentr.Count - 1; i++)
{
    #region STEP 3 - Рухаємось по дузі
    if (dummyCentr[i].X != 0 && dummyCentr[i].Y != 0)
    {
        cp.W = 5[i + 1];

        a1 = GetAngle(dummyCross[2 * i + 0], dummyCentr[i]);
        a2 = GetAngle(dummyCross[2 * i + 1], dummyCentr[i]);
        CrPr = CrossProduct(
            dummyCross[2 * i + 0],
            dummyCross[2 * i + 1],
            dummyCentr[i]);

        if (a1 < a2 && CrPr >= 0)
            clockwise = 1;
        if (a1 < a2 && CrPr < 0)
            clockwise = -1;

        if (a1 > a2 && CrPr < 0)
            clockwise = -1;
        if (a1 > a2 && CrPr >= 0)
            clockwise = 1;
        do
        {
            a1 += clockwise * Math.PI / 180;
            cp.X = dummyCentr[i].X + R[i + 1] * Math.Cos(a1);
            cp.Y = dummyCentr[i].Y + R[i + 1] * Math.Sin(a1);
            cp.Z = dummyP[i + 1].Z;

            // Перевірка висоти точки
            if (!CheckDepthPoint(cp, bath, objSize))
                wrongway = true;

            if (++n > MAX_POINTS)
                goto EndC;
            else
                dummyVTraj.Add(cp);
        }
        while (!IsLastPointOnLine(cp, dummyCross[2 * i + 1], (int)cp.W));
    }
}
    
```

Рис. 9. Програмний код обчислень руху по дузі кола

Отже, у наведених рис. 7 – рис. 9 міститься програма реалізація описаних раніше розрахунків. У кодї змінні та функції мають такі значення: *dummyCross* — масив точок перетину кола з прямою від початку руху (пряма  $AP_1$  на рис. 6); *dummyCentr* — масив обрахованих точок кола обертання під час поворотів; *CrossProduct* — функція обчислення векторного добутку між векторами від центра кола до точок перетину кола з прямими (на рис. 6  $OP_1$ ,  $OP_2$ ). Векторний добуток, зокрема, використовується для встановлення напрямку повороту.

Визначення точок дуги здійснюється в такий спосіб: розпізнаються кути перетину між центром і точками перетину дуги з прямими. Починаючи з першого кута, визначаються координати точки, а перший кут поступово нарощується у напрямку другого кута. Тобто якщо поглянути на рис. 6, кут поступово нарощується від  $OP_1$ , поки не досягне значення кута  $OP_2$ , і, отже, пройшовши всю дугу повороту.

### ВИСНОВКИ

Обрахунок приблизного можливого маневрування корабля у водному середовищі потребує зазначення не тільки координат руху, а й швидкості об'єкта, особливо на відрізку повороту, що значно впливатиме на зміну радіуса повороту. Приклад розробленого інтерфейсу користувача зображено на рис. 10. У лівому стовпчику можна додати або видалити точки зміни напрямку судна. У разі додання точки праворуч сторони на полі створюється точка з відповідним номером, і користувач може як задати специфічні координати у таблиці, так і змінити розташування перетягуванням кружечка.

Оскільки зміна швидкості пересування безпосередньо впливає на виконання поворотів, було ухвалення рішення збирати інформацію саме про бажану користувачем швидкість у точці зміни курсу для коректного відображення маневру. У разі можливості побудови маневру праворуч від точок червоною товстою лінією будується маневр судна та зберігається для подальших обчислень.

#### Список використаної літератури

1. **Borkowski P.** *The Ship Movement Trajectory Prediction Algorithm Using Navigational Data Fusion // Sensors.* 2017. 17(6). P. 1432. URL: <https://doi.org/10.3390/s17061432>
2. **Hassani V., Lande S. V.** *Path planning for marine vehicles using Bezier curves // IFAC Pap.* 2018. 51. P. 305–310.
3. **Yang G. J., Choi B. W.** *Smooth trajectory planning along Bezier curve for mobile robots with velocity constraints // Int. J. Control Autom.* 2013. 6. P. 225–234.
4. **Choi J. W., Curry R., Elkaim G.** *Path planning based on bézier curve for autonomous ground vehicles // Proceedings of the Advances in Electrical and Electronics Engineering-IAENG Special Edition of the World Congress on Engineering and Computer Science 2008, San Francisco, CA, USA, 22–24 October 2008.* P. 158–166.
5. **Miller A., Walczak S.** *Maritime Autonomous Surface Ship's Path Approximation Using Bézier Curves // Symmetry.* 2020. 12(10). P. 1704. URL: <https://doi.org/10.3390/sym12101704>
6. **Methodology for Controlling the Ship's Path during the Turn in Confined Waterways / S. Vujičić, R. Mohović, I. Đ. Tomaš // Scientific Journal of Maritime Research.** 2018. 32. P. 28–35.
7. **Mobility pattern analysis of ship trajectories based on semantic transformation and topic model / L. Huang, Y. Wen, W. Guo [et al.] // Ocean Eng.** 2020. 201. 107092.
8. **Plessen M. M. G., Bemporad A.** *Reference trajectory planning under constraints and path tracking using linear time-varying model predictive control for agricultural machines // Biosyst. Eng.* 2017. 153. P. 28–41.
9. **Generation of temporal-spatial Bezier curve for simultaneous arrival of multiple unmanned vehicles / W. Yu, W. Shuo, W. Rui, T. Min // Inf. Sci.** 2017. 418. P. 34–45.
10. **Model predictive control based trajectory generation for autonomous vehicles-an architectural approach / M. Nolte, M. Rose, T. Stolte, M. Maurer // Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017.** P. 798–805.
11. **Fossen T. I.** *Guidance and Control of Ocean Vehicles.* John Wiley & Sons: New York, NY, USA, 1994.
12. **Borkowski P.** *Data fusion in a navigational decision support system on a sea-going vessel // Pol. Marit. Res.* 2012. 19. P. 78–85.

T. Pyrohovska, T. Dovzhenko

#### DEVELOPMENT OF A USER INTERFACE FOR INPUT OF A CURVILINEAR OBJECT MOVEMENT TRAJECTORY WITH VARIABLE SPEED

The article considers the development of an interface for the most accurate input by the user of the trajectory of the curvilinear movement of the ship in the marine environment with variable speed for the further development of software for modeling the marine water area. Convenient modeling of objects in the sea water area requires a convenient interface that allows the user not only to set an arbitrary curvilinear trajectory of movement, but also checks the possibility of movement of the given object in relation to its size and speed of movement. Therefore, for the implementation of an interactive interface for entering a curvilinear trajectory of a ship, it is necessary to take into account the dynamics of the controlled object. When evaluating the trajectory, the physical limitations of the vessel must be satisfied. They determine the minimum radius of curvature, tangential acceleration and all three speeds (longitudinal, transverse, angular). The projected trajectory should be smooth, and its speeds along all three axes should be constant. Taking into account information about the physical forces affecting the movement of the ship, the formulas for calculating the points of the trajectory were derived, which are subsequently used both for visual representation and for calculations in the simulation of the experiment. The problem was divided to 3 parts: linear movement, the movement along the arc of the circle, movement after the turn. The algorithm for getting the position of the ship at any point of time during the turn was provided/. The article also provides an example of coding derived formulas using C# programming language. The results were provided by demonstrating the developed interface.

**Keywords:** interactive interface; modeling of object movement; curvilinear movement with variable speed; ship trajectory; interface development, C#.