

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Ю. Ф. Лазарєв, О. М. Сапегін

Алгоритми безплатформових інерціальних систем орієнтації прикладні методи синтезу і аналізу точності

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня магістра за спеціальністю
151 «Автоматизація та комп'ютерно-інтегровані технології»*

Київ
КПІ ім. Ігоря Сікорського
2018

Рецензенти: *Успенський В. Б.* д-р техн. наук, професор кафедри систем і процесів керування НТУ "ХПІ"

Головач С. В. – к. т. н., провідний інженер КП СПБ "Арсенал"

Відповідальний редактор: *Аврутов В. В.*, канд. техн. наук, доцент кафедри приладів і систем орієнтації та навігації КПІ ім. Ігоря Сікорського

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 3 від 22.11.2018 р.) за поданням Вченої ради Приладобудівного факультету (протокол № 9/18 від 29.10.2018 р.)

Електронне мережне навчальне видання

Лазарєв Юрій Федорович, канд. техн. наук, доц.
Сапегін Олександр Миколайович

Алгоритми
Безплатформових
інерціальних систем
орієнтації
прикладні методи синтезу і аналізу точності

Лазарєв Ю. Ф. Алгоритми безплатформових інерціальних систем орієнтації (прикладні методи синтезу і аналізу точності) [Електронний ресурс] : навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», спеціалізації «Комп'ютерно-інтегровані технології та системи навігації і керування» / Ю. Ф. Лазарєв, О. М. Сапегін ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 6,8 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 141 с.

Викладені основи методів побудови і дослідження точності алгоритмів оцінювання кутів поточної орієнтації основи у безплатформових інерціальних систем орієнтації (БІСО), що входять до складу безплатформових інерціальних навігаційних систем (БІНС). Детально розглянуті питання комп'ютерного моделювання БІСО з метою виявлення похибок досліджуваних алгоритмів.

Призначено для підготовки магістрів вищих навчальних закладів. Може бути використано в курсовому та дипломному проектуванні. Корисно для інженерів та наукових співробітників.

Зміст

ВСТУП.....	5
1. ПАРАМЕТРИ ВИЗНАЧЕННЯ КУТОВОГО ПОЛОЖЕННЯ ТІЛА.....	7
1.1. Кути Ейлера-Крилова і матриця напрямних косинусів.....	8
1.2. Вектор Ейлера єдиного повороту.....	12
1.3. Кватерніон повороту.....	16
1.4. Вектор повороту Гіббса.....	18
1.5. Вектор скінченного повороту Родріга.....	19
2. РІВНЯННЯ ОРІЄНТАЦІЇ	20
2.1 Кінематичні рівняння Ейлера	20
2.2. Матричне рівняння Пуассона	22
2.3. Векторні кінематичні рівняння орієнтації.....	22
2.3.1. Векторне рівняння орієнтації Ейлера.....	23
2.3.2. Кватерніонне рівняння орієнтації	24
2.3.3. Рівняння через вектор Гіббса	26
2.3.4. Рівняння через вектор Родріга	27
3. АНАЛІЗ ТОЧНОСТІ АЛГОРИТМІВ ЧЕРЕЗ КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ.....	28
3.1. СТРУКТУРА КОМП'ЮТЕРНОЇ МОДЕЛІ БІСО.....	28
3.2. РОЗРОБКА КЕРУВАЛЬНОЇ ПРОГРАМИ	30
3.3. ДОСЛІДЖЕННЯ ПОХИБОК АЛГОРИТМІВ	35
3.4. ПРО ФОРМУ ПОДАВАННЯ ХАРАКТЕРИСТИК ТОЧНОСТІ АЛГОРИТМІВ.....	40
3.5. ПРО ТОЧНІСТЬ ДЕЯКИХ ВІДОМИХ АЛГОРИТМІВ.....	43
4. СИНТЕЗ АЛГОРИТМІВ НА ОСНОВІ МЕТОДІВ РУНГЕ-КУТТИ	47
4.1. ГОЛОВНІ ФОРМУЛИ	48
4.2. ЗАВДАННЯ НА ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ	51
5. МЕТОД ШКАРА СИНТЕЗУ АЛГОРИТМІВ.....	53
5.1. ПОЛІНОМІАЛЬНА ІНТЕРПОЛЯЦІЯ КУТОВОЇ ШВИДКОСТІ.....	53
5.1.1. Апроксимація кутової швидкості за вимірювання її проєкцій.....	54

5.1.2. Апроксимація кутової швидкості за вимірювання приростів квазікоординат	58
5.2. АЛГОРИТМИ НА ҐРУНТІ ВЕКТОРНОГО РІВНЯННЯ ЕЙЛЕРА	64
5.2.1. Випадок вимірювання приростів квазікоординат	64
5.2.2. Випадок вимірювання вектора кутової швидкості	75
5.3. АЛГОРИТМИ НА ҐРУНТІ РІВНЯННЯ ПУАССОНА	85
5.3.1. Випадок вимірювань кутової швидкості	88
5.3.2. Вимірюються прирости квазікоординат	107
6. СИНТЕЗ РОЗГІННИХ АЛГОРИТМІВ.....	123
6.1. ПОБУДОВА АЛГОРИТМІВ ДЛЯ КВАТЕРНІОННОГО РІВНЯННЯ ОРІЄНТАЦІЇ	123
6.1.1. Постановка задачі	123
6.1.2. Побудова наближених розв'язків допоміжного рівняння	124
6.1.3. Побудова алгоритмів	125
6.1.4. Конкретизація значень коефіцієнтів	127
6.1.5. Побудова алгоритмів	129
6.2. СТВОРЕННЯ КОМП'ЮТЕРНОЇ МОДЕЛІ	129
6.3. МОДЕЛЮВАННЯ РОБОТИ БІСО	134
6.3. ВИСНОВКИ	138
ЛІТЕРАТУРА.....	140

Вступ

Навчальний посібник призначений для підготовки магістрів і забезпечує навчальну дисципліну "Додаткові розділи теорії чутливих елементів". Його метою є надання майбутньому магістру теоретичний і практичний матеріал, який забезпечив би поглиблену прикладну підготовку у галузі проектування приладів і систем автономної орієнтації.

Головну увагу у посібнику приділено питанням синтезу алгоритмів безплатформових інерціальних систем орієнтації (БІСО) та аналізу їх точності.

БІСО входить до складу безплатформових інерціальних навігаційних систем (БІНС). Головною її задачею є визначення поточної кутової орієнтації основи, на якій встановлено БІНС, відносно місцевої географічної системи координат, з тим, щоб здійснювати перепроєціювання вектора поточного прискорення основи зі зв'язаної з основою системи координат на географічну систему координат.

Безплатформова інерціальна система орієнтації (БІСО) складається з двох головних блоків:

- 1) блоку вимірювачів, призначеного вимірювати за допомогою гіротахometrів (або інтегровальних гіротахometrів) значення або проєкцій абсолютної кутової швидкості основи в окремі моменти часу, розділені так званим *кроком опитування* вимірювачів, або інтегралів за часом від цих проєкцій (так званих *квазікоординат*) за один крок опитування;
- 2) бортового обчислювального пристрою, який за даними цього вимірювання обчислює (шляхом чисельного інтегрування за тим чи іншим алгоритмом) параметри поточної орієнтації основи.

Зазвичай алгоритми БІСО поділяють на *розгінні* і *безрозгінні*.

Розгінними називають алгоритми, які використовують задля визначення точної орієнтації основи вимірювальну інформацію на кількох попередніх кроках опитування вимірювачів. При цьому визначається орієнтація основи через один крок опитування вимірювачів.

До *безрозгінних* (самостартувальних) відносять алгоритми, для роботи яких не використовується вимірювальна інформація на попередніх кроках опитування, тому їм не потрібна операція "розгону" (стартування) алгоритму. Алгоритм у цьому випадку визначає орієнтацію основи через декілька кроків опитування. Тому крок інтегрування H диференціальних рівнянь орієнтації складається з кількох кроків h опитування $H = m \cdot h$, де величина m називатимемо «кроковістю» алгоритму.

У сучасних БІСО розгінні алгоритми практично не використовуються. Тому у подальшому мова буде йти про безрозгінні алгоритми.

У посібнику вивчення властивостей моделей алгоритмів і БІСО ґрунтується на використанні можливостей комп'ютерної системи MATLAB.

1. Параметри визначення кутового положення тіла

Існують кілька можливостей математичного подання кутового положення тіла у просторі [1, с 16-41]:

- 1) за допомогою кутів Ейлера-Крилова, тобто трійки значень кутів послідовних поворотів навколо трьох координатних осей;
- 2) матрицею напрямних косинусів (МНК) між координатними осями базової системи координат і системи координат, пов'язаної з тілом;
- 3) параметрами вектора Ейлера єдиного повороту;
- 4) кватерніоном повороту від вихідної системи координат до кінцевої.

Розглянемо взаємозв'язок між цими видами визначення орієнтації тіла і побудуємо комп'ютерні засоби, що забезпечують перехід від одного виду до інших

1.1. Кути Ейлера-Крилова і матриця напрямних косинусів

У якості прикладу за основу прийнемо послідовність кутів Ейлера-Крилова, наведену на рисунку 1.1

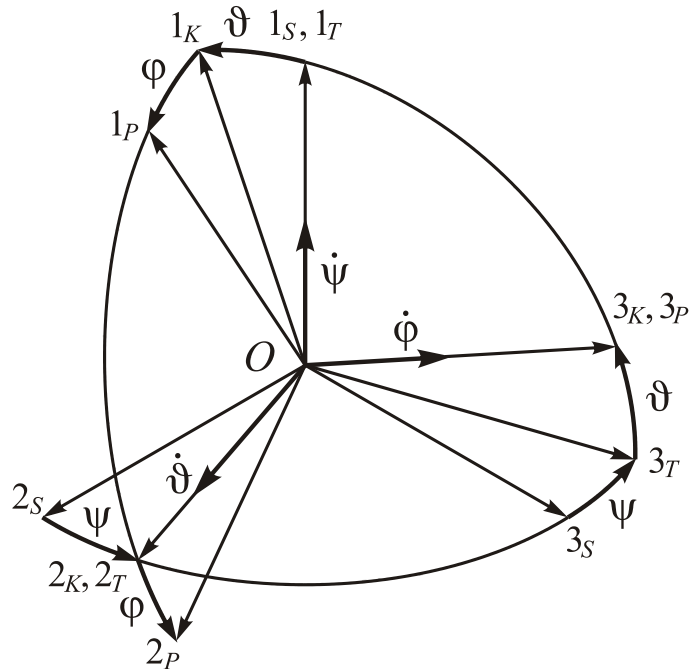


Рис. 1.1. Послідовність поворотів на кути Ейлера-Крилова

Відповідна матриця напрямних косинусів переходу від базової системи координат S до кінцевої P має такий вигляд:

C^{SP}	1_P	2_P	3_P
1_S	$\cos\vartheta\cos\varphi$	$-\cos\vartheta\sin\varphi$	$\sin\vartheta$
2_S	$\cos\psi\sin\varphi + \sin\psi\cos\varphi\sin\vartheta$	$\cos\psi\cos\varphi - \sin\psi\sin\varphi\sin\vartheta$	$-\sin\psi\cos\vartheta$
3_S	$\sin\psi\sin\varphi - \cos\psi\cos\varphi\sin\vartheta$	$\sin\psi\cos\varphi + \cos\psi\sin\varphi\sin\vartheta$	$\cos\psi\cos\vartheta$

Складемо універсальну (тобто яку можна використати для будь якої послідовності поворотів) програму, яка обчислює елементи матриці напрямних косинусів за відомими значеннями кутів поворотів навколо координатних осей, номери яких вказано у векторі k . Для цього спочатку створимо процедуру **mnke** обчи-

слення елементарної матриці напрямних косинусів одного повороту навколо вісі, номер якої задано:

```
function y=mnke(x,k)
% Функція формування Матриці Направних Косинусів (y)
% за заданим значення куту (x) та номера повороту (к)
% осі, навколо якої здійснюється поворот
% Лазарєв Ю.Ф. 18-07-1998р. 19-09-1998р.
y = zeros(3,3); y(k,k) = 1;
if k==1
    y(2,2) = cos(x); y(3,3) = y(2,2);
    y(3,2) = sin(x); y(2,3) = - y(3,2);
end
if k==2
    y(1,1) = cos(x); y(3,3) = y(1,1);
    y(1,3) = sin(x); y(3,1) = - y(1,3);
end
if k==3
    y(1,1) = cos(x); y(2,2) = y(1,1);
    y(2,1) = sin(x); y(1,2) = - y(2,1);
end
```

Тепер універсальна програма обчислення МНК виглядатиме просто:

```
function y = ug2mnk(x,k)
% Процедура знаходження Матриці Направних Косинусів за кутами Ейлера
% y = ug2mnk(x,k)
% обчислює матрицю напрямних косинусів 'y'
% за заданими значеннями вектора 'x' трьох кутів x(1), x(2), x(3)
% послідовних поворотів та вектора 'k' номерів осей k(1), k(2), k(3),
% навколо яких ці повороти здійснюються
% Використовується процедура-функція 'mnke'
% (с) Лазарєв Ю.Ф. 18-07-1998р. 09-06-2014
y = mnke(x(1),k(1))*mnke(x(2),k(2))*mnke(x(3),k(3));
```

Наведемо приклад застосування цієї процедури. Нехай послідовність поворотів така, яка вказана на рисунку 1.1. Тоді, вочевидь, вектор номерів осей послі-

довних поворотів буде таким $k=[1 \ 2 \ 3]$. Задамо значення кутів послідовності $\psi = 0,4$; $\vartheta = -0,3$; $\varphi = 1$. При цьому вектор значень кутів повороту набуде вигляду $x = [0.4 \ -0.3 \ 1]$.

Складемо керувальну програму:

$x = [0.4 \ -0.3 \ 1]$; $k=[1 \ 2 \ 3]$;

$C = \text{ug2mnk}(x,k)$

В результаті отримаємо матрицю напрямних косинусів у вигляді

$C =$

5.1617e-001 -8.0389e-001 -2.9552e-001

7.1287e-001 5.9449e-001 -3.7203e-001

4.7475e-001 -1.8638e-002 8.7992e-001

Задля перевірки правильності результату потрібно утворити процедуру зворотного переходу від заданої МНК до значень кутів послідовних поворотів навколо заданих (номерами) осей. Текст її наведено нижче

```
function y= mnk2ug(x,kv)
% MNK2UG перетворює матрицю напрямних косинусів у кути Ейлера
% Звернення:      y = mnk2ug(x,kv)
% дозволяє обчислити вектор "y", що складається зі
% значень y(1),y(2),y(3) кутів послідовних поворотів
% навколо координатних осей відповідно з номерами kv(1), % kv(2) і kv(3)
% за заданою матрицею напрямних косинусів "x" розміром (3*3)
% між осями початкової та кінцевої % систем координат
%   x(i,j) = cos(Ip,Jk),
%   де Ip - позначення осі з номером "i" початкової,
%   а Jk - осі з номером "j" кінцевої систем координат.
% Примітка:
%   kv(1) має не дорівнювати kv(2), а kv(2) має не дорівнювати kv(3)
%   Ю.Ф.Лазарєв,      Останні змінювання 5-12-2001
k1=kv(1); k2=kv(2); k3=kv(3);
k = k1*10^2+k2*10+k3;
if k==123 | k==312 | k==231,
    y(1) = atan2(-x(k2,k3),x(k3,k3));    y(2) = asin(x(k1,k3));
```

```

y(3) = atan2(-x(k1,k2),x(k1,k1));
sp= x(k1,k3);  cp=sqrt(1-x(k1,k3)^2);
st=sin(y(3));  ct=cos(y(3));
if (x(k1,k1)*ct)<0
    y(1) = atan2(x(k2,k3),-x(k3,k3));    y(2) = atan2(sp,-cp);
    y(3) = atan2(x(k1,k2),-x(k1,k1));
end
elseif k==132 | k==321 | k==213,
    y(2) = asin(-x(k1,k3));    y(1) = atan2(x(k2,k3),x(k3,k3));
    y(3) = atan2(x(k1,k2),x(k1,k1));
else y(2) = acos(x(k1,k3));
    if k==121,
        y(1) = atan2(x(2,1),-x(3,1));    y(3) = atan2(x(1,2),x(1,3));
    elseif k==313,
        y(1) = atan2(x(1,3),-x(1,2));    y(3) = atan2(x(3,1),x(3,2));
    elseif k==232,
        y(1) = atan2(x(3,2),-x(1,2));    y(3) = atan2(x(2,3),x(2,1));
    elseif k==212,
        y(1) = atan2(x(1,2),x(3,2));    y(3) = atan2(x(2,1),-x(2,3));
    elseif k==323,
        y(1) = atan2(x(2,3),x(1,3));    y(3) = atan2(x(3,2),-x(3,1));
    elseif k==131,
        y(1) = atan2(x(3,1),x(2,1));    y(3) = atan2(x(1,3),-x(1,2));
    end
end
end

```

Якщо звернутися до цієї процедури

$Ug = mnk2ug(C,k)$,

то отримаємо

$$Ug = \begin{matrix} & 0.4 & -0.3 & 1 \end{matrix}$$

Як бачимо, виходить той самий вектор кутів повороту, що й вихідний. Отже, створені програмні засоби працюють добре.

Якщо система координат S спочатку за допомогою поворотів переходить у систему координат T , а потім шляхом інших поворотів – в систему координат P , то операцію знаходження параметрів повороту, які забезпечують поворот від сис-

теми S безпосередньо до системи P , прийнято називати операцією складання поворотів.

Переваги застосування послідовності трьох кутів – наочність подання поворотів

Недоліки застосування послідовності трьох кутів – непристосованість до безпосереднього проєціювання векторів, і не існує операції складання поворотів.

Переваги застосування матриці напрямних поворотів – добра пристосованість до проєціювання векторів та існування простого правила складання поворотів

$$C^{SP} = C^{ST} \cdot C^{TP}, \quad (1.1)$$

МНК результуючого повороту дорівнює добутку МНК складових поворотів, записаних зліва направо у порядку здійснення самих поворотів.

Недоліки застосування матриці напрямних поворотів – не наочність подання поворотів і велика кількість параметрів, що описують кожний поворот.

1.2. Вектор Ейлера єдиного повороту

Відповідно до теореми Ейлера довільне кутове положення однієї системи координат P відносно іншої системи координат S можна подати як єдиний поворот на кут δ (кут Ейлера) навколо певно орієнтованої осі (осі Ейлера). При цьому (див. [1, с. 28]) кут Ейлера доволі просто визначається з матриці C^{SP} напрямних косинусів за формулою

$$\cos \delta = \frac{1}{2} [Tr(C^{SP}) - 1], \quad (1.2)$$

де $Tr(C^{SP})$ – позначення сліду (trace) матриці напрямних косинусів, тобто суми її діагональних елементів. З самого визначення єдиного повороту випливає, що напрямні косинуси e_{1S} , e_{2S} і e_{3S} осі Ейлера відносно осей 1_S , 2_S , 3_S вихідної сис-

теми координат S дорівнюють напрямним косинусам e_{1P} , e_{2P} і e_{3P} цієї осі відносно кінцевої системи координат P , тому можна зняти позначення індексу системи координат і користуватися позначеннями

$$e_1 = e_{1S} = e_{1P}; \quad e_2 = e_{2S} = e_{2P}; \quad e_3 = e_{3S} = e_{3P}.$$

Ці напрямні косинуси визначаються співвідношеннями

$$e_1 = \frac{c_{32}^{SP} - c_{23}^{SP}}{2 \sin \delta}; \quad e_2 = \frac{c_{13}^{SP} - c_{31}^{SP}}{2 \sin \delta}; \quad e_3 = \frac{c_{21}^{SP} - c_{12}^{SP}}{2 \sin \delta}. \quad (1.3)$$

Зворотний перехід від параметрів єдиного повороту Ейлера до матриці напрямних косинусів можна подати у вигляді

$$\mathbf{C}^{SP} = \mathbf{E} + \sin \delta \cdot (\mathbf{e} \times) + (1 - \cos \delta) \cdot (\mathbf{e} \times)^2, \quad (1.4)$$

де \mathbf{E} є одиничною матрицею

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

\mathbf{e} – одиничним вектором (матрицею-стовпцем)

$$\mathbf{e} = [e_1, \quad e_2, \quad e_3]^T,$$

а $(\mathbf{e} \times)$ є позначенням кососиметричної матриці

$$(\mathbf{e} \times) = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}.$$

Якщо ввести поняття *вектора Ейлера*

$$\boldsymbol{\varphi} = \delta \cdot \mathbf{e}$$

і врахувати, що довжина цього вектора дорівнює

$$\varphi = |\boldsymbol{\varphi}| = \delta,$$

то співвідношення (1.4) можна подати у вигляді функції цього вектора

$$\mathbf{C}^{SP} = \mathbf{E} + \frac{\sin \varphi}{\varphi} \cdot (\boldsymbol{\varphi} \times) + \frac{1 - \cos \varphi}{\varphi^2} \cdot (\boldsymbol{\varphi} \times)^2. \quad (1.5)$$

Користуючись цими відомостями, створимо кілька комп'ютерних процедур, що здійснюють ці перетворення:

- перетворення матриці напрямних косинусів у параметри єдиного повороту:

```
function [d,e]=MKN2VEul(C)
% Процедура преобразования матрицы направляющих косинусов
% в вектор Ейлерового поворота
% [d,e]=MKN2VEul(C)
% где d - величина угла поворота вокруг Эйлеровой оси,
% а e - вектор направляющих косинусов этой оси

% Лазарев Ю. Ф. 19-11-2014
cd=(trace(C)-1)/2; d=acos(cd);
if d==0
    e=[1 0 0];
else
    zn=2*sin(d);
    e(1)=(C(3,2)-C(2,3))/zn;
    e(2)=(C(1,3)-C(3,1))/zn;
    e(3)=(C(2,1)-C(1,2))/zn;
end
```

- перетворення вектора Ейлера у матрицю напрямних косинусів

```
function C=VEul2MKN(d,e)
% Перетворення вектору Ейлера єдиного повороту у матрицю напрямних
% косинусів

% Лазарев Ю.Ф. 19-11-2014 02-09-2015
if d==0
    e=[1 0 0];
end
E=eye(3,3);
cd=cos(d); sd=sin(d);
ex=[0 -e(3) e(2); e(3) 0 -e(1); -e(2) e(1) 0];
C = E + sd*ex + (1-cd)*ex*ex;
```

Перевіримо ці процедури у дії за допомогою програми

```
x = [0.4 -0.3 1]; k=[1 2 3];
```

```
C = ug2mnk(x,k)
```

```
[d,e]=MNK2VEul(C)
```

```
C1=VEul2MNK(d,e)
```

```
dC= C1-C
```

В результаті її виконання отримаємо

C =

0.51617	-0.80389	-0.29552
0.71287	0.59449	-0.37203
0.47475	-0.018638	0.87992

d =

1.0526

e =

0.20339	-0.44333	0.87298
---------	----------	---------

C1 =

0.51617	-0.80389	-0.29552
0.71287	0.59449	-0.37203
0.47475	-0.018638	0.87992

dC =

1.1102e-016	-1.1102e-016	5.5511e-017
-1.1102e-016	-1.1102e-016	0
0	0	1.1102e-016

Похибка проведення операцій, як можна впевнитися, не перевищує $2 \cdot 10^{-16}$, що є вельми високою точністю (на межі подання чисел у системі Matlab).

Переваги застосування вектора Ейлера – наочність подання повороту і кількість параметрів, близька до мінімальної.

Недоліки застосування вектора Ейлера – не встановлено операції складання поворотів, не пристосованість до проєціювання векторів.

1.3. Кватерніон повороту

Кватерніоном називають (див. [1, с. 29-36]) гіперкомплексне число вигляду

$$\mathbf{Q} = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3,$$

де q_0, q_1, q_2, q_3 – дійсні числа, які у сукупності називають параметрами Родріга-Гамільтона; $\mathbf{i}, \mathbf{j}, \mathbf{k}$ – уявні одиниці, які мають такі властивості

$$\mathbf{i} \circ \mathbf{i} = \mathbf{j} \circ \mathbf{j} = \mathbf{k} \circ \mathbf{k} = -1; \quad \mathbf{i} \circ \mathbf{j} = \mathbf{k} = -\mathbf{j} \circ \mathbf{i}; \quad \mathbf{j} \circ \mathbf{k} = \mathbf{i} = -\mathbf{k} \circ \mathbf{j}; \quad \mathbf{k} \circ \mathbf{i} = \mathbf{j} = -\mathbf{i} \circ \mathbf{k};$$

(\circ – позначка кватерніонного добутку).

Величина q_0 складає скалярну частину кватерніону, а величини q_1, q_2, q_3 можна розглядати як складові деякого вектора – матриці-рядка

$$\mathbf{q} = [q_1, q_2, q_3]. \quad (1.6)$$

Кватерніон єдиного повороту на деякий кут δ має вигляд

$$\mathbf{Q}^{SP} = \cos \frac{\delta}{2} + \mathbf{e}^{SP} \sin \frac{\delta}{2}, \quad (1.7)$$

де $\mathbf{e}^{SP} \{e_1, e_2, e_3\}$ – одиничний вектор осі повороту.

Переходячи до подання кватерніону в комп'ютерній системі Matlab, подаватимемо кватерніон у вигляді вектора довжиною у чотири елементи

$$\mathbf{Q} = [q_0, q_1, q_2, q_3] = [q_0, \mathbf{q}]. \quad (1.8)$$

Отже, вираз (1.8) дає можливість визначити кватерніон повороту через параметри вектора Ейлера у такий спосіб

$$\mathbf{Q}^{SP} = \left[\cos \frac{\delta}{2}, \sin \frac{\delta}{2} \mathbf{e} \right], \quad (1.9)$$

де $\mathbf{e} \{e_1, e_2, e_3\}$ – одиничний вектор осі повороту.

Зворотне перетворення кватерніона повороту у матрицю напрямних косинусів дається формулою [1, (3.30), с. 35] :

$$\mathbf{C}^{SP} = \mathbf{E} + 2[q_0 \mathbf{E} + (\mathbf{q} \times)](\mathbf{q} \times), \quad (1.10)$$

де $(\mathbf{q} \times)$ є позначенням кососиметричної матриці

$$(\mathbf{q} \times) = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}. \quad (1.11)$$

Перейдемо до комп'ютерного програмування.

Перетворення кватерніона у матрицю напрямних косинусів може бути здійснено процедурою

```
function C= kwat2mnk(kw)
% Перетворює кватерніон kw у матрицю напрямних косинусів C
% Звернення: y = quat2ug(x)
% дозволяє обчислити матрицю напрямних косинусів "y"
% за заданим кватерніоном "x"
% Лазарев Ю.Ф. 03-09-2015 р.
q0=kw(1); qx=[0 -kw(4) kw(3); kw(4) 0 -kw(2); -kw(3) kw(2) 0];
E=eye(3,3);
C = E + 2*[q0*E + qx]*qx;
```

Зворотнє перетворення можна здійснити функцією

```
function kw = mnk2kwat(C)
% Перетворює матрицю напрямних косинусів C у кватерніон kw
% Звернення: y = mnk2kwat(x)
% дозволяє обчислити кватерніон "y"
% за заданою матрицею напрямних косинусів "x"
% Лазарев Ю.Ф. 03-09-2015 р.
[d,e]=MNK2VEul(C);
cd2= cos(d/2); sd2= sin(d/2);
q0=cd2; q=sd2*e;
kw=[q0 q];
```

Створимо програму перевірки правильності роботи цих функцій

```
k=[1 2 3];
Ug= [0.5 0.3 -1];
C= ug2mnk(Ug,k);
```

```

kw = mnk2kwat(C);
C1= kwat2mnk(kw);
Ug1=mnk2ug(C1,k)
dUg=Ug1-Ug

```

В результаті її роботи виходить

```

Ug1 =      0.5      0.3      -1
dUg = 1.1102e-016  5.5511e-017 -2.2204e-016

```

що впевнює у адекватності створених програм-функцій і високій точності здійснення перетворень і обчислень.

Переваги застосування кватерніонів –

- 1) *кількість параметрів близька до мінімальної;*
- 2) *існує досить просте правило складання поворотів:*

$$\mathbf{Q}^{SP} = \mathbf{Q}^{ST} \circ \mathbf{Q}^{TP},$$

а саме, – кватерніон вислідного повороту дорівнює кватерніонному добутку кватерніонів складальних поворотів, записаних зліва направо у порядку самих поворотів;

- 3) *пристосованість до перепроєціювання векторів.*

Недоліки застосування кватерніонів – відсутність наочності в поданні повороту.

1.4. Вектор повороту Гіббса

Вектор Гіббса тісно пов'язаний з вектором Ейлера і визначається у такий спосіб

$$\mathbf{g}^{SP} = \operatorname{tg} \frac{\delta}{2} \cdot \mathbf{e}, \quad (1.12)$$

де δ і \mathbf{e} – параметри вектора Ейлера.

Переваги застосування вектора Гіббса –

- 1) мінімальна кількість параметрів;
- 2) існує правило складання поворотів:

$$\mathbf{g}^{SP} = \frac{\mathbf{g}^{ST} + \mathbf{g}^{TP} + \mathbf{g}^{ST} \times \mathbf{g}^{TP}}{1 - \mathbf{g}^{ST} \cdot \mathbf{g}^{TP}}, \quad (1.13)$$

Недоліки застосування вектора Гіббса – складне правило складання поворотів та непристосованість до перепросціювання векторів.

1.5. Вектор скінченного повороту Родріга

Найбільш розповсюдженим є подання повороту у вигляді так званого вектору скінченного повороту (вектора Родріга):

$$\mathbf{p}^{SP} = 2tg \frac{\delta}{2} \cdot \mathbf{e}, \quad (1.14)$$

де δ і \mathbf{e} – параметри вектора Ейлера.

Переваги і недоліки застосування вектора Родріга ті самі, що й вектора Гіббса.

2. Рівняння орієнтації

В теоретичній механіці під *кінематичними рівняннями* розуміють диференціальні рівняння, що пов'язують проекції кутової швидкості тіла, рух якого досліджується, на осі той чи іншої системи координат з похідними від параметрів повороту.

У цьому посібнику йдеться про так звані безплатформові інерціальні системи орієнтації (БІСО), які призначені визначати поточну орієнтацію основи, на якій вони встановлені, використовуючи вимірювання проекцій кутової швидкості основи на осі системи координат, пов'язаної з основою. Тому під *рівняннями орієнтації* розумітимемо такі кінематичні рівняння, в яких похідні від параметрів орієнтації (поворотів) виражаються через проекції ω_{1P}^{PS} , ω_{2P}^{PS} і ω_{3P}^{PS} вектора ω^{PS} кутової швидкості основи (система координат P) відносно вихідної системи координат S .

Наведемо [1, с. 42-50] рівняння орієнтації, виражені через подані раніше види параметрів орієнтації. Задля однаковості їх зручно подавати у стандартній формі Коши, тобто у вигляді

$$\frac{dy}{dt} = \mathbf{Z}(t; \mathbf{y}), \quad (2.1)$$

де \mathbf{y} – вектор фазових змінних, а $\mathbf{Z}(t; \mathbf{y})$ – вектор-функція часу і фазових змінних.

2.1 Кінематичні рівняння Ейлера

Кінематичні рівняння Ейлера, відомі в теоретичній механіці, можна в наших цілях подати у вигляді (див. рис. 1.1):

$$\begin{cases} \frac{d\psi}{dt} = (\omega_{1P}^{PS} \cdot \cos \varphi - \omega_{2P}^{PS} \cdot \sin \varphi) / \cos \vartheta; \\ \frac{d\vartheta}{dt} = \omega_{2P}^{PS} \cdot \cos \varphi + \omega_{1P}^{PS} \cdot \sin \varphi; \\ \frac{d\varphi}{dt} = \omega_{3P}^{PS} - \operatorname{tg} \vartheta \cdot (\omega_{1P}^{PS} \cdot \cos \varphi - \omega_{2P}^{PS} \cdot \sin \varphi). \end{cases} \quad (2.2)$$

Якщо ввести вектор фазових змінних у вигляді

$$\mathbf{y} = [y_1, y_2, y_3] = [\psi, \vartheta, \varphi], \quad (2.3)$$

а під вектор-функцією розуміти вектор

$$\mathbf{Z}(t; \mathbf{y}) = [Z_1, Z_2, Z_3], \quad (2.4)$$

де

$$\begin{aligned} Z_1 &= [\omega_1(t) \cos y_1 - \omega_2(t) \sin y_1] / \cos y_2; \\ Z_2 &= \omega_2(t) \cdot \cos y_1 + \omega_1(t) \cdot \sin y_1; \\ Z_3 &= \omega_3(t) - [\omega_1(t) \cos y_1 - \omega_2(t) \sin y_1] \operatorname{tg} y_2; \end{aligned} \quad (2.5)$$

то кінематичні рівняння Ейлера набудуть стандартної форми Коши (2.1).

Наведемо приклад процедури обчислення правих частин кінематичного рівняння Ейлера (тут і далі всюди – мовою системи Matlab):

```
function z=UgEul(t,y)
% Процедура обчислення правих частин кінематичних рівнянь Ейлера
%      t – значення поточного моменту часу
%      y – вектор значень кутів повороту
%      z – вектор значень похідних за часом від кутів повороту
% Лазарев Ю. Ф.      15-09-2015
[psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(t);
cf=cos(y(3)); sf=sin(y(3)); ct=cos(y(2)); st=sin(y(2));
z(1)=(omX*cf-omY*sf)/ct;
z(2)=omY*cf+omX*sf;
z(3)=omZ-st*z(1);
```

2.2. Матричне рівняння Пуассона

Кінематичне рівняння орієнтації Пуассона використовує матрицю напрямних косинусів і має форму

$$\frac{d\mathbf{C}^{SP}}{dt} = \mathbf{C}^{SP} \cdot (\boldsymbol{\omega}_P^{PS} \times). \quad (2.6)$$

Тут використано позначення кососиметричної матриці

$$(\boldsymbol{\omega}_P^{PS} \times) = \begin{bmatrix} 0 & -\omega_{3P}^{PS} & \omega_{2P}^{PS} \\ \omega_{3P}^{PS} & 0 & -\omega_{1P}^{PS} \\ -\omega_{2P}^{PS} & \omega_{1P}^{PS} & 0 \end{bmatrix}. \quad (2.7)$$

Рівнянню (2.6) можна надати форми (2.1), якщо вважати

$$\mathbf{y} = \mathbf{C}^{SP}; \quad \mathbf{Z} = \mathbf{y} \cdot [\boldsymbol{\omega}(t) \times], \quad (2.8)$$

де

$$(\boldsymbol{\omega}(t) \times) = \begin{bmatrix} 0 & -\omega_3(t) & \omega_2(t) \\ \omega_3(t) & 0 & -\omega_1(t) \\ -\omega_2(t) & \omega_1(t) & 0 \end{bmatrix}. \quad (2.9)$$

Наведемо текст відповідної процедури:

```
function z=Pois(t,y)
% Процедура обчислення правих частин матричного рівняння Пуассона
% t – значення поточного моменту часу
% y – матриця напрямних косинусів
% z - матриця-похідна за часом від матриці напрямних косинусів
% Лазарев Ю. Ф. 15-09-2015
[psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(t);
OMx =[0 -omZ omY; omZ 0 -omX; -omY omX 0];
z = y*OMx;
```

2.3. Векторні кінематичні рівняння орієнтації

2.3.1. Векторне рівняння орієнтації Ейлера

Позначимо вектор (матрицю-стовпець з його проекцій) Ейлера

$$\boldsymbol{\varphi} = \boldsymbol{\delta} \cdot \mathbf{e}, \quad (2.10)$$

де \mathbf{e} – позначення вектора-стовпця

$$\mathbf{e} = [e_1 \ e_2 \ e_3]^T.$$

Кінематичне рівняння через вектор Ейлера (формула Борца) у матричній формі має вигляд (доведення його на основі статті Борца [10] див. у розділі 6 цього посібника);

$$\frac{d\boldsymbol{\varphi}}{dt} = \boldsymbol{\omega}_P^{PS} + \frac{1}{2}(\boldsymbol{\varphi} \times) \boldsymbol{\omega}_P^{PS} + \frac{1 - \frac{\delta}{2} \operatorname{ctg} \frac{\delta}{2}}{\delta^2} (\boldsymbol{\varphi} \times)(\boldsymbol{\varphi} \times) \boldsymbol{\omega}_P^{PS}, \quad (2.11)$$

де позначено матриці

$$\boldsymbol{\omega}_P^{PS} = [\omega_{1P}^{PS} \ \omega_{2P}^{PS} \ \omega_{3P}^{PS}]^T; \quad (\boldsymbol{\varphi} \times) = \begin{bmatrix} 0 & -\varphi_3 & \varphi_2 \\ \varphi_3 & 0 & -\varphi_1 \\ -\varphi_2 & \varphi_1 & 0 \end{bmatrix}. \quad (2.12)$$

Якщо розуміти під фазовими змінними вектор

$$\mathbf{y} = [y_1, y_2, y_3]^T = [\varphi_1, \varphi_2, \varphi_3]^T, \quad (2.13)$$

то рівняння (2.11) можна записати у вигляді

$$\mathbf{Z} = \boldsymbol{\omega}(t) + \frac{1}{2}(\mathbf{y} \times) \boldsymbol{\omega}(t) + \frac{1 - \frac{y}{2} \operatorname{ctg} \frac{y}{2}}{y^2} (\mathbf{y} \times)(\mathbf{y} \times) \boldsymbol{\omega}(t), \quad (2.14)$$

де

$$(\mathbf{y} \times) = \begin{bmatrix} 0 & -y_3 & y_2 \\ y_3 & 0 & -y_1 \\ -y_2 & y_1 & 0 \end{bmatrix}; \quad y = \sqrt{y_1^2 + y_2^2 + y_3^2};$$

$$\boldsymbol{\omega}(t) = [\omega_1(t), \omega_2(t), \omega_3(t)]^T. \quad (2.15)$$

Тоді рівняння (2.11) набудуть форми Коши (2.1).

Недоліком рівняння (2.11) є невизначеність значення коефіцієнта при його третій складовій за умови $\delta = 0$. Враховуючи, що за малих значень δ цей коефіцієнт можна подати у вигляді ряду

$$\frac{1 - \frac{\delta}{2} \operatorname{ctg} \frac{\delta}{2}}{\delta^2} = \frac{1}{12} \left(1 + \frac{\delta^2}{60} + \frac{\delta^4}{2520} + \frac{\delta^6}{10800} + \dots \right),$$

краще користуватися рівнянням виду

$$\frac{d\boldsymbol{\varphi}}{dt} = \boldsymbol{\omega}_P^{PS} + \frac{1}{2} (\boldsymbol{\varphi} \times) \boldsymbol{\omega}_P^{PS} + \frac{1}{12} \left(1 + \frac{\delta^2}{60} + \frac{\delta^4}{2520} + \frac{\delta^6}{10800} + \dots \right) (\boldsymbol{\varphi} \times) (\boldsymbol{\varphi} \times) \boldsymbol{\omega}_P^{PS}.$$

С точністю до членів шостого порядку мализни можна використовувати таке спрощене рівняння:

$$\frac{d\boldsymbol{\varphi}}{dt} \approx \boldsymbol{\omega}_P^{PS} + \frac{1}{2} (\boldsymbol{\varphi} \times) \boldsymbol{\omega}_P^{PS} + \frac{1}{12} (\boldsymbol{\varphi} \times) (\boldsymbol{\varphi} \times) \boldsymbol{\omega}_P^{PS}. \quad (2.16)$$

Подамо процедуру `v_Eul` правих частин векторного рівняння Ейлера:

```
function z=v_Eul(t,y)
% Процедура обчислення правих частин векторного рівняння Ейлера
% t – значення поточного моменту часу
% y – значення поточного вектора Ейлера
% z - значення поточного вектора похідної за часом від вектора Ейлера
% Лазарев Ю. Ф.      15-09-2015
[psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(t);
OM =[omX omY omZ]';
d = norm(y); e=y/mVE; yX=[0 -y(3) y(2); y(3) 0 y(1); -y(2) y(1) 0];
z = OM + yX*OM/2 + yX*yX*OM/12;
```

2.3.2. Кватерніонне рівняння орієнтації

Кватерніонне рівняння орієнтації має вигляд [1, с. 45]:

$$\frac{d\mathbf{Q}^{SP}}{dt} = \frac{1}{2} \mathbf{Q}^{SP} \circ \boldsymbol{\Omega}_P^{PS},$$

де \mathbf{Q}^{SP} – кватерніон повороту від вихідної системи координат S до рухомої системи координат P ; $\mathbf{\Omega}_P^{PS}$ – вектор-кватерніон, скалярна частина якого дорівнює нулю, а векторна складається з проекцій $\mathbf{\omega}_P^{PS}$ на осі системи P вектора $\mathbf{\omega}^{PS}$ кутової швидкості повороту системи P відносно системи S ; \circ позначає кватерніонний добуток.

Якщо використовувати матричну форму (1.8) позначення кватерніону повороту, то кватерніонне рівняння орієнтації складатиметься з двох рівнянь

$$\begin{cases} \frac{dq_0}{dt} = -\frac{1}{2} \mathbf{q}^T \cdot \mathbf{\omega}_P^{PS} \\ \frac{d\mathbf{q}}{dt} = \frac{1}{2} [q_0 \cdot \mathbf{\omega}_P^{PS} + (\mathbf{q} \times) \cdot \mathbf{\omega}_P^{PS}] \end{cases}, \quad (2.16)$$

де використані позначення матриць (1.6) і (1.11).

Нехай

$$\mathbf{y} = [y_1, \mathbf{y}_2] = [q_0, q_1, q_2, q_3], \quad (2.17)$$

$$\mathbf{Z} = [Z_1, \mathbf{Z}_2] = \left[-\frac{1}{2} \mathbf{y}_2^T \cdot \mathbf{\omega}(t); \quad \frac{1}{2} [y_1 \cdot \mathbf{\omega}(t) + (\mathbf{y}_2 \times) \cdot \mathbf{\omega}(t)] \right]. \quad (2.18)$$

Тоді система рівнянь (2.16) перетворюється на рівняння вигляду (2.1).

Відповідна процедура обчислення правих частин рівнянь може виглядати так:

```
function z=Kwatern(t,y)
% Процедура обчислення правих частин кватерніонного рівняння орієнтації
% t – значення поточного моменту часу
% y – значення поточного кватерніону повороту
% z - значення поточної похідної за часом від кватерніону повороту

% Лазарев Ю. Ф.      15-09-2015
[psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(t);
OM =[omX omY omZ];
q0=y(1);  q=y(2:4);
z(1) = -q*OM'/2;
z(2:4) = (q0*OM + cross(q,OM))/2;
```

2.3.3. Рівняння через вектор Гіббса

Рівняння орієнтації, подане через вектор Гіббса і виражене у матричній формі, можна подати у формі

$$\frac{d\mathbf{g}^{SP}}{dt} = \frac{1}{2} \left[\boldsymbol{\omega}_P^{PS} + (\mathbf{g}^{SP} \times) \boldsymbol{\omega}_P^{PS} + \left((\mathbf{g}^{SP})^T \cdot \boldsymbol{\omega}_P^{PS} \right) \mathbf{g}^{SP} \right], \quad (2.19)$$

де

$$\mathbf{g}^{SP} = [g_{1p}^{SP} \ g_{2p}^{SP} \ g_{3p}^{SP}] ; \quad (\mathbf{g}^{SP} \times) = \begin{bmatrix} 0 & -g_{3p}^{SP} & g_{2p}^{SP} \\ g_{3p}^{SP} & 0 & -g_{1p}^{SP} \\ -g_{2p}^{SP} & g_{1p}^{SP} & 0 \end{bmatrix}. \quad (2.20)$$

У позначеннях

$$\mathbf{y} = \mathbf{g}^{SP}, \quad (2.21)$$

та

$$\mathbf{Z} = \frac{1}{2} \left[\boldsymbol{\omega}(t) + (\mathbf{y} \times) \boldsymbol{\omega}(t) + (\mathbf{y}^T \cdot \boldsymbol{\omega}(t)) \mathbf{y} \right]. \quad (2.22)$$

рівняння (2.19) набуває форми Коші (2.1).

Наведемо відповідну процедуру обчислення правої частини цього рівняння

```
function z=Gibbs(t,y)
% Процедура обчислення правих частин векторного рівняння Гіббса
% t – значення поточного моменту часу
% y – поточне значення вектора Гіббса
% z - поточне значення вектора похідної за часом від вектора Гіббса

% Лазарев Ю. Ф.      15-09-2015
[psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(t);
OM =[omX omY omZ]';
yX=[0 -y(3) y(2); y(3) 0 y(1); -y(2) y(1) 0];
z = (OM + yX*OM + (y'*OM)*y)/2;
```

2.3.4. Рівняння через вектор Родріга

Рівняння орієнтації через вектор Родріга у матричній формі багато в чому аналогічні рівнянню через вектор Гіббса

$$\frac{d\mathbf{p}^{SP}}{dt} = \boldsymbol{\omega}_P^{PS} + \frac{1}{2}(\mathbf{p}^{SP} \times) \boldsymbol{\omega}_P^{PS} + \frac{1}{4}((\mathbf{p}^{SP})^T \cdot \boldsymbol{\omega}_P^{PS}) \mathbf{p}^{SP}, \quad (2.23)$$

де використані позначення

$$\mathbf{p}^{SP} = [p_{1p}^{SP} \ p_{2p}^{SP} \ p_{3p}^{SP}]^T; \quad (2.24)$$

$$(\mathbf{p}^{SP} \times) = \begin{bmatrix} 0 & -p_{3p}^{SP} & p_{2p}^{SP} \\ p_{3p}^{SP} & 0 & -p_{1p}^{SP} \\ -p_{2p}^{SP} & p_{1p}^{SP} & 0 \end{bmatrix}; \quad (\mathbf{p}^{SP})^T = [p_{1p}^{SP} \ p_{2p}^{SP} \ p_{3p}^{SP}]^T = \begin{bmatrix} p_{1p}^{SP} \\ p_{2p}^{SP} \\ p_{3p}^{SP} \end{bmatrix}.$$

Форму (2.1) це рівняння набуває, якщо використати позначення

$$\mathbf{y} = \mathbf{p}; \quad \mathbf{Z} = \boldsymbol{\omega}(t) + \frac{1}{2}(\mathbf{y} \times) \boldsymbol{\omega}(t) + \frac{1}{4}(\mathbf{p}^T \cdot \boldsymbol{\omega}(t)) \mathbf{y}. \quad (2.25)$$

Текст відповідної процедури може бути таким:

```
function z=Rodrigues(t,y)
% Процедура обчислення правих частин векторного рівняння Родріга
% t – значення поточного моменту часу
% y – поточне значення вектора Родріга
% z - поточне значення вектора похідної за часом від вектора Родріга
% Лазарев Ю. Ф. 15-09-2015
[psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(t);
OM=[omX omY omZ]';
yX=[0 -y(3) y(2); y(3) 0 y(1); -y(2) y(1) 0];
z = OM + yX*OM/2 + (y'*OM)*y/4;
```

3. Аналіз точності алгоритмів через комп'ютерне моделювання

3.1. Структура комп'ютерної моделі БІСО

Розглянемо задачу дослідження похибок БІСО, обумовлених виключно похибками алгоритму чисельного інтегрування, закладеного у обчислювачі БІСО. Цю задачу розв'язуватимемо шляхом комп'ютерного моделювання.

Відповідна комп'ютерна модель, вочевидь, має складатися з таких програмних блоків:

- 1) блоку імітатора поточного кутового руху основи;
- 2) блоку імітатора обчислювального пристрою БІСО з вбудованим в нього досліджуваним алгоритмом;
- 3) блоку порівнювання параметрів орієнтації, отриманих блоком імітатора обчислювача БІСО, з "дійсними" значеннями цих параметрів, отриманих у блоці імітатора руху основи; отримання масиву значень похибок алгоритму як функцій часу.

У якості кінцевих наочних параметрів поточної орієнтації основи зручніше за все обрати послідовність кутів Ейлера-Крилова.

Наведемо текст програми **UGDvObj_Laz** прикладу процедури, яка реалізує імітатор кутового руху основи:

```
function [psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(t)
% Процедура обчислення поточних параметрів руху основи
% для стандартного розташування осей координат та кутів основи
% Лазарєв Ю. Ф.      01-04-2014      05-09-2015
global psit psim tetm gam omp omt omg ep et eg
```

```

% Кути основи
psi = psit*t + psim*sin(omp*t+ep);    tet = tetm*sin(omt*t+et);
ga=gam*sin(omg*t+eg);
% Швидкості змінювання кутів
pst=psim*omp*cos(omp*t+ep); tett=tetm*omt*cos(omt*t+et);
gat=gam*omg*cos(omg*t+eg);
% Синуси та косинуси
cg=cos(ga); ct=cos(tet);  st=sin(tet);  sg=sin(ga);
% Проекції кутової швидкості основи на її осі
omX = gat+pst*st;  omY = pst*ct*cg + tett*sg;  omZ = - pst.*ct.*sg + tett.*cg;

```

Перевіримо слушність роботи цієї функції за допомогою такої допоміжної програми:

```

% Upr_Obj
% Програма перевірки роботи функції UgDvObj_laz
clear all, clc
global psit psim tetm gam omp omt omg ep et eg
% Встановлення значень параметрів руху основи
psit = 0.01;      Tt = 0;      Gt = 0;
psim = 0.08;      tetm = 0.1;   gam = 0.15;
omp = 3;          omt = 1;      omg = 2;
ep = 0*pi/180;    et = 90*pi/180;  eg = -90*pi/180;
h=0.1; t=0:h:30; n=length(t);
for k=1:n
    T=t(k);
    [psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(T);
    Psi(k)=psi;  Tet(k)=tet;  Gm(k)=ga;
end
plot(t,Psi,'o-',t,Tet,'*-',t,Gm,'.-'), grid
set(gca,'FontSize',14)
title('Кутовий рух основи (точний)')
xlabel('Час, с'), ylabel('Кути, радіани')
legend('\psi', '\theta', '\phi', 2)

```

Запуск цієї програми приводить до появи у графічному вікні монітора картини, наведеної на рисунку 3.1.

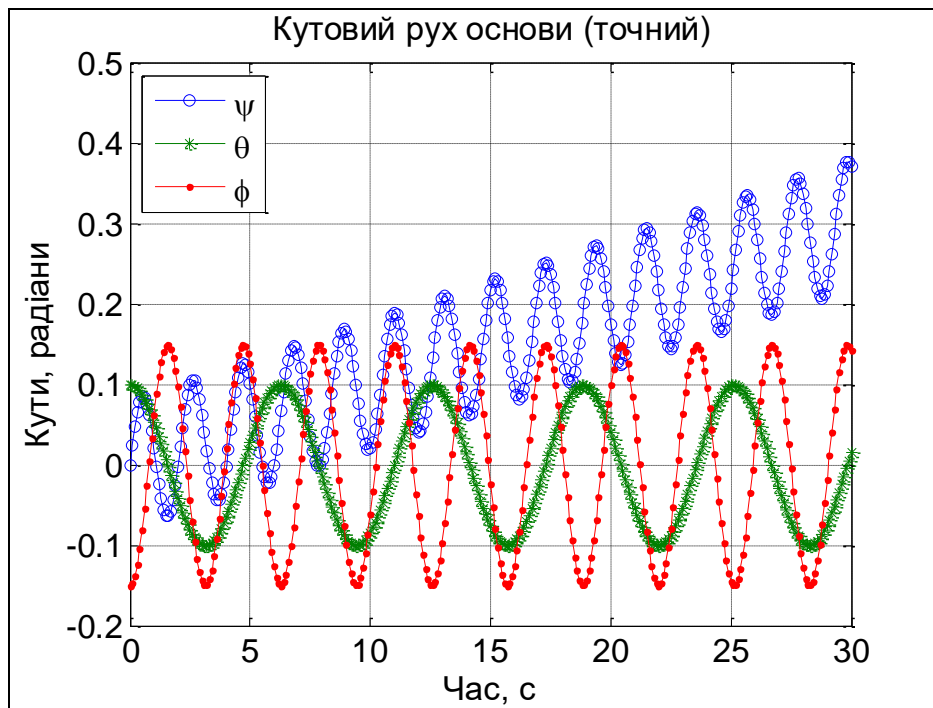


Рис. 3.1. Результат роботи блоку імітатора руху основи

Як бачимо, графіки змінювання у часі кутів повороту основи відповідають заданим законам руху основи.

3.2. Розробка керувальної програми

Мета керувальної програми – на основі попередніх створених процедур організувати виконання наступної послідовності дій:

- 1) задати значення постійних параметрів, що визначають кутовий рух основи;
- 2) задати значення кроку інтегрування, кроку опитування і тривалості модельного часу, протягом якого виявлятимуться алгоритмічні похибки заданого алгоритму БІСО;
- 3) обчислити значення параметрів кутового положення у початковий момент часу;
- 4) організувати циклічне звернення до процедури алгоритму чисельного інтегрування заданого диференціального рівняння орієнтації і

формування масиву похибок визначення кутів Ейлера на кожному циклі (кроці) інтегрування;

- 5) викликати процедуру обчислення середньої швидкості змінювання похибок визначення кутів орієнтації (дрейфу похибок) за відомими (обчисленими) масивами алгоритмічних похибок;
- б) оформити графічне вікно монітора комп'ютеру, де були б наведені:
 - а) графіки залежності алгоритмічних похибок від часу; б) текстова інформація про усі використані задані параметри руху основи і параметри інтегрування; в) обчислені дрейфи похибок з усіх кутів.

Розглянемо приклад такої програми.

Дослідимо похибки чисельного інтегрування матричного рівняння Пуассона шістьма методами Рунге-Кутти, алгоритми яких були наведені раніше.

Задля виявлення дрейфів похибок задамо такий рух основи:

$$\psi(t) \equiv 0; \quad \vartheta(t) = a_m \sin(\omega t + \varepsilon); \quad \varphi(t) = a_m \sin(\omega t),$$

за якого основа здійснює синхронні коливання з однаковою амплітудою навколо двох ортогональних осей (з кутів ϑ та φ) і не повертається навколо третьої осі.

Нижче приведено текст відповідної керувальної програми за ім'ям **Pois_RuKu_UgSk_upr**.

```
% Pois_RuKu_UgSk_upr
% Керувальна програма дослідження алгоритмічних похибок чисельного
% інтегрування матричного рівняння Пуассона методами Рунге-Кутти
% Використовувані процедури
% UgDvObj_laz - обчислення поточних точних значень кутів повороту основи
% і проєкцій його кутової швидкості на осі зв'язаної системи координат
% Ru_Ku_21 - перший метод Рунге-Кутти 2-го порядку (однокроковий)
% Ru_Ku_22 - другий метод Рунге-Кутти 2-го порядку (двокроковий)
% Ru_Ku_31 - перший метод Рунге-Кутти 3-го порядку (двокроковий)
% Ru_Ku_32 - другий метод Рунге-Кутти 3-го порядку (трикроковий)
% Ru_Ku_41 - перший метод Рунге-Кутти 4-го порядку (двокроковий)
% Ru_Ku_42 - другий метод Рунге-Кутти 4-го порядку (трикроковий)
% Ur_Poisson - обчислення правої частини рівняння Пуассона
% vect2ksm - перетворення вектора в косиметричну матрицю
```

```

% mnke      - побудова матриці напрямних косинусів,
%            відповідну одному повороту навколо заданої осі
% Ug2Mnk    - перетворення послідовності в матрицю напрямних косинусів
% Mnk2Ug1   - перетворення матриці напрямних косинусів у кути Ейлера
% Drift_L   - обчислення величини дрейфу за масивом даних
% GRAF_Ru_Ku_Laz1 - побудова графіків 18 масивів даних від часу
% Лазарев Ю. Ф.      14-11-2014    15-09-2015
clc, clear all
global psim tetm gam omp omt omg ep et eg
global h sname sprogram DR
sprogram='Pois-RuKu-UgSk-upr'; sname='Лазарев Ю. Ф.';
% Завдання значень постійних параметрів кутового руху основи
psim=0.; tetm = 0.1; gam=0.1;
omp=0; omt=1; omg=1;
ep=0*pi/180; et=0*pi/180; eg=0*pi/180;
% Завдання параметрів інтегрування
t=0; % Початковий момент часу
h=0.01; % Крок опитування вимірювачів
tfinal=100; % Кінцевий час інтегрування
% Розрахунок матриці напрямних косинусів у початковий момент часу
[psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(t); %Розрахунок кутів орієнтації
ugV=[psi,tet,ga]; kV=[2,3,1]; C0=ug2mnk(ugV,kV); % МНК
%% Метод Рунге-Кутти 21 другого порядку (однокроковий)
H=h; C2=C0;
% Початкові значення обчислювальних масивів похибок
t=0; imas = 1; tt2(imas)=t;
dpsi2(imas)=0; dtet2(imas)=0; dga2(imas)=0;
while (imas-1)*H<tfinal
    t=(imas-1)*H; % формування значення поточного часу
    [tout,yout]=Ru_Ku_21('Ur_Poisson',H,t,C2); % Інтегрування
    C2=yout; t=tout; % Формування МНК і моменту часу через крок
    imas=imas+1; % Змінювання значення лічильника
    tt2(imas)=tout; % Формування масиву моментів часу
    [ps2,te2,ga2]=Mnk2Ug1(C2); % Визначення кутів за новою МНК
    % Розрахунок точних значень кутів
    [psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(t);
    % Розрахунок похибок визначення кутів
    dpsi2(imas)=psi-ps2; dtet2(imas)=tet-te2; dga2(imas)=ga-ga2;
end
%% Метод Рунге-Кутти 22 другого порядку (двокроковий)

```



```

H=2*h; C2=C0; t=0; imas=1; tt22(imas)=t;
dpsi22(imas)=0; dtet22(imas)=0; dga22(imas)=0;
while (imas-1)*H<tfinal
    t=(imas-1)*H;
    [tout,yout]=Ru_Ku_22('Ur_Poisson',H,t,C2);
    C2=yout; t=tout;
    imas=imas+1;
    tt22(imas)=tout; % Масив моментів часу
    [ps2,te2,ga2]=Mnk2Ug1(C2);
    [psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(t);
    dpsi22(imas)=psi-ps2; dtet22(imas)=tet-te2; dga22(imas)=ga-ga2;
end

%%% Метод Рунге-Кутти 31 третього порядку (двокроковий)
H=2*h; C3=C0; t=0; imas=1; tt31(imas)=t;
dpsi31(imas)=0; dtet31(imas)=0; dga31(imas)=0;
while (imas-1)*H<tfinal
    t=(imas-1)*H;
    [tout,yout]=Ru_Ku_31('Ur_Poisson',H,t,C3);
    C3=yout; t=tout; imas=imas+1;
    tt31(imas)=tout; % Масив моментів часу
    [ps3,te3,ga3]=Mnk2Ug1(C3);
    [psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(t);
    dpsi31(imas)=psi-ps3; dtet31(imas)=tet-te3; dga31(imas)=ga-ga3;
end

%%% Метод Рунге-Кутти 32 третього порядку (трикроковий)
H=3*h; C3=C0; t=0; imas=1; tt3(imas)=t;
dpsi3(imas)=0; dtet3(imas)=0; dga3(imas)=0;
while (imas-1)*H<tfinal
    t=(imas-1)*H;
    [tout,yout]=Ru_Ku_32('Ur_Poisson',H,t,C3);
    C3=yout; t=tout; imas=imas+1;
    tt3(imas)=tout; % Масив моментів часу
    [ps3,te3,ga3]=Mnk2Ug1(C3);
    [psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(t);
    dpsi3(imas)=psi-ps3; dtet3(imas)=tet-te3; dga3(imas)=ga-ga3;
end

%%% Метод Рунге-Кутти 41 четвертого порядку (двокроковий)
H=2*h; C4=C0; t=0; imas=1; tt4(imas)=t;
dpsi4(imas)=0; dtet4(imas)=0; dga4(imas)=0;
while (imas-1)*H<tfinal

```

```

t=(imas-1)*H;
[tout,yout]=Ru_Ku_41('Ur_Poisson',H,t,C4);
C4=yout; t=tout;      imas=imas+1;
tt4(imas)=tout;      % Масив моментів часу
[ps4,te4,ga4]=Mnk2Ug1(C4);
[psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(t);
dpsi4(imas)=psi-ps4; dtet4(imas)=tet-te4; dga4(imas)=ga-ga4;
end

%%% Метод Рунге-Кутти 42 четвертого порядку (трикроковий)
H=3*h; C4=C0; t=0; imas=1; tt42(imas)=t;
dpsi42(imas)=0; dtet42(imas)=0; dga42(imas)=0;
while (imas-1)*H<tfinal
    t=(imas-1)*H;
    [tout,yout]=Ru_Ku_42('Ur_Poisson',H,t,C4);
    C4=yout; t=tout;      imas=imas+1;
    tt42(imas)=tout;      % Масив моментів часу
    [ps4,te4,ga4]=Mnk2Ug1(C4);
    [psi,tet,ga,omX,omY,omZ]=UgDvObj_laz(t);
    dpsi42(imas)=psi-ps4; dtet42(imas)=tet-te4; dga42(imas)=ga-ga4;
end

% Обчислення дрейфів
[Const,Dr12]=drift_L(tt2,dpsi2); [Const,Dr122]=drift_L(tt22,dpsi22);
[Const,Dr131]=drift_L(tt31,dpsi31); [Const,Dr13]=drift_L(tt3,dpsi3);
[Const,Dr14]=drift_L(tt4,dpsi4); [Const,Dr142]=drift_L(tt42,dpsi42);
[Const,Dr222]=drift_L(tt22,dtet22); [Const,Dr322]=drift_L(tt22,dga22);
[Const,Dr231]=drift_L(tt31,dtet31); [Const,Dr331]=drift_L(tt31,dga31);
[Const,Dr22]=drift_L(tt2,dtet2); [Const,Dr23]=drift_L(tt3,dtet3);
[Const,Dr24]=drift_L(tt4,dtet4); [Const,Dr32]=drift_L(tt2,dga2);
[Const,Dr33]=drift_L(tt3,dga3); [Const,Dr34]=drift_L(tt4,dga4);
[Const,Dr242]=drift_L(tt42,dtet42); [Const,Dr342]=drift_L(tt42,dga42);
DR=[Dr12,Dr13,Dr14,Dr22,Dr23,Dr24,Dr32,Dr33,Dr34,...
    Dr122,Dr222,Dr322,Dr131,Dr231,Dr331,Dr142,Dr242,Dr342];

% Виведення графічної і текстової інформації у графічне вікно монітору
GRAF_Ru_Ku_Laz1(tt2(1:end-1),dpsi2(1:end-1),dtet2(1:end-1),dga2(1:end-1),...
    tt22(1:end-1),dpsi22(1:end-1),dtet22(1:end-1),dga22(1:end-1),...
    tt31(1:end-1),dpsi31(1:end-1), dtet31(1:end-1),dga31(1:end-1),...
    tt3(1:end-1),dpsi3(1:end-1), dtet3(1:end-1),dga3(1:end-1),...
    tt4(1:end-1),dpsi4(1:end-1),dtet4(1:end-1),dga4(1:end-1),...
    tt42(1:end-1),dpsi42(1:end-1),dtet42(1:end-1),dga42(1:end-1))

```

Спеціальна функція **drift_L**, текст якої наведено нижче, виконуватиме операцію апроксимування масиву даних X при заданому масиві t відповідних аргументів поліномом першого степеня (прямою на графіку $X(t)$):

```
function [Const,Dr]=drift_L(t,X)
% Процедура обчислення дрейфу функції X часу
% студент Цибульник 04-10-2012
P=polyfit(t,X,1); Dr=P(1); Const=P(2);
```

При цьому виходять значення двох величин: Dr – кутового коефіцієнту прямої і $Const$ – ординати точки перетинання прямої з віссю ординат. Саме величина Dr і є значенням шуканого дрейфу похибок у розглядуваному випадку.

3.3. Дослідження похибок алгоритмів

На рисунках 3.2...3.4 показані результати роботи цієї програми за таких значень параметрів руху

$$a_m = 0,1 \text{ радіана}; \quad \omega = 1 \text{ радіан / с.}$$

Рисунок 3.2 подає результати за кроку опитування вимірювачів $h = 1$ с і при двох значеннях зсуву фаз між коливаннями основи $\varepsilon = 0^\circ$ и $\varepsilon = 90^\circ$. Аналогічно, рисунки 3.3 и 3.4 демонструють результати при $h = 0,1$ та $h = 0,01$ с.

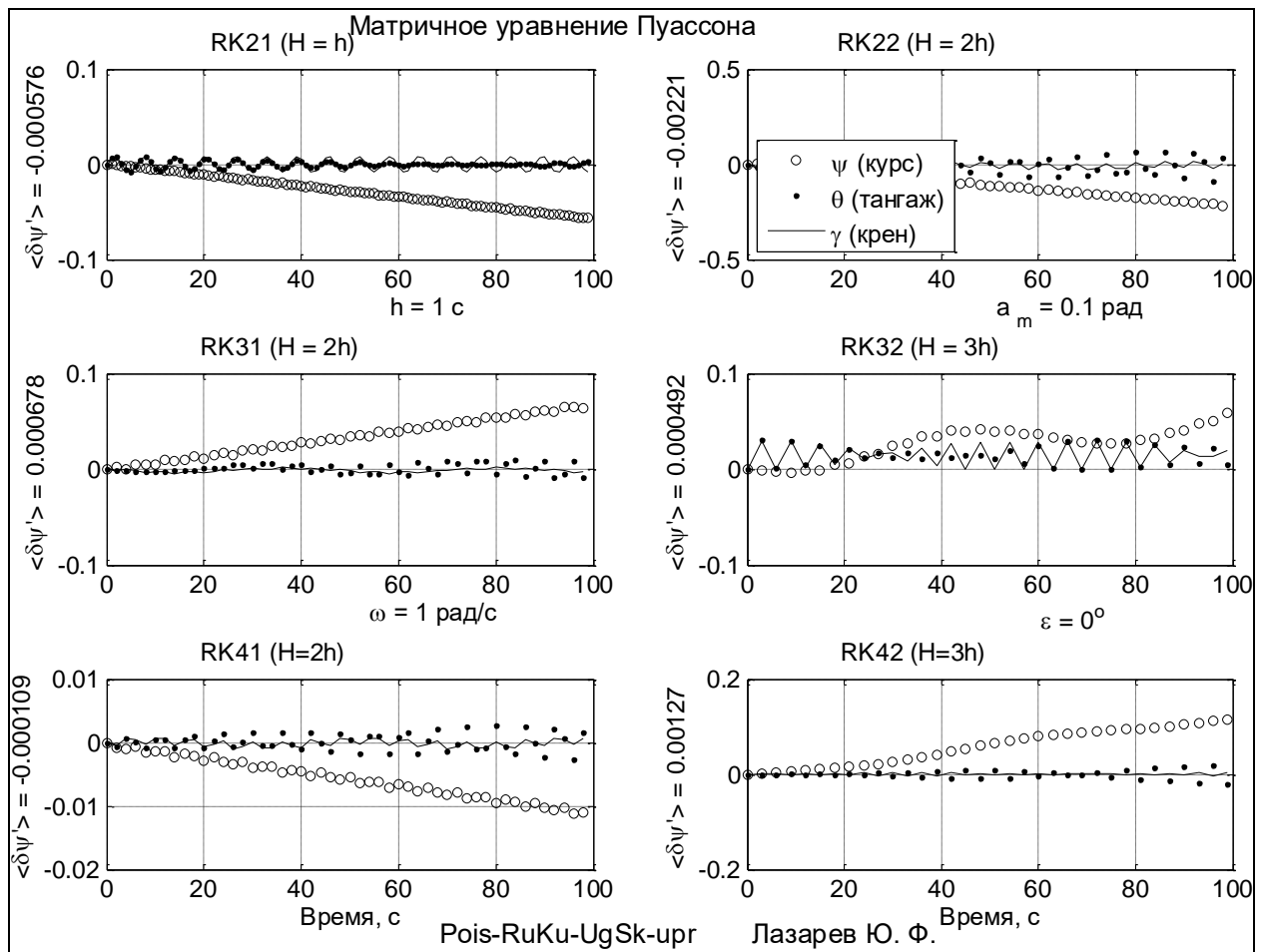
З наведених графіків випливає:

1) дрейфи похибок алгоритмів суттєво залежать від зсуву фаз між коливаннями основи навколо двох ортогональних осей; ця залежність близька до гармонічної: за одних зсувів дрейф є додатним, за інших – від'ємним, за деяких – відсутній;

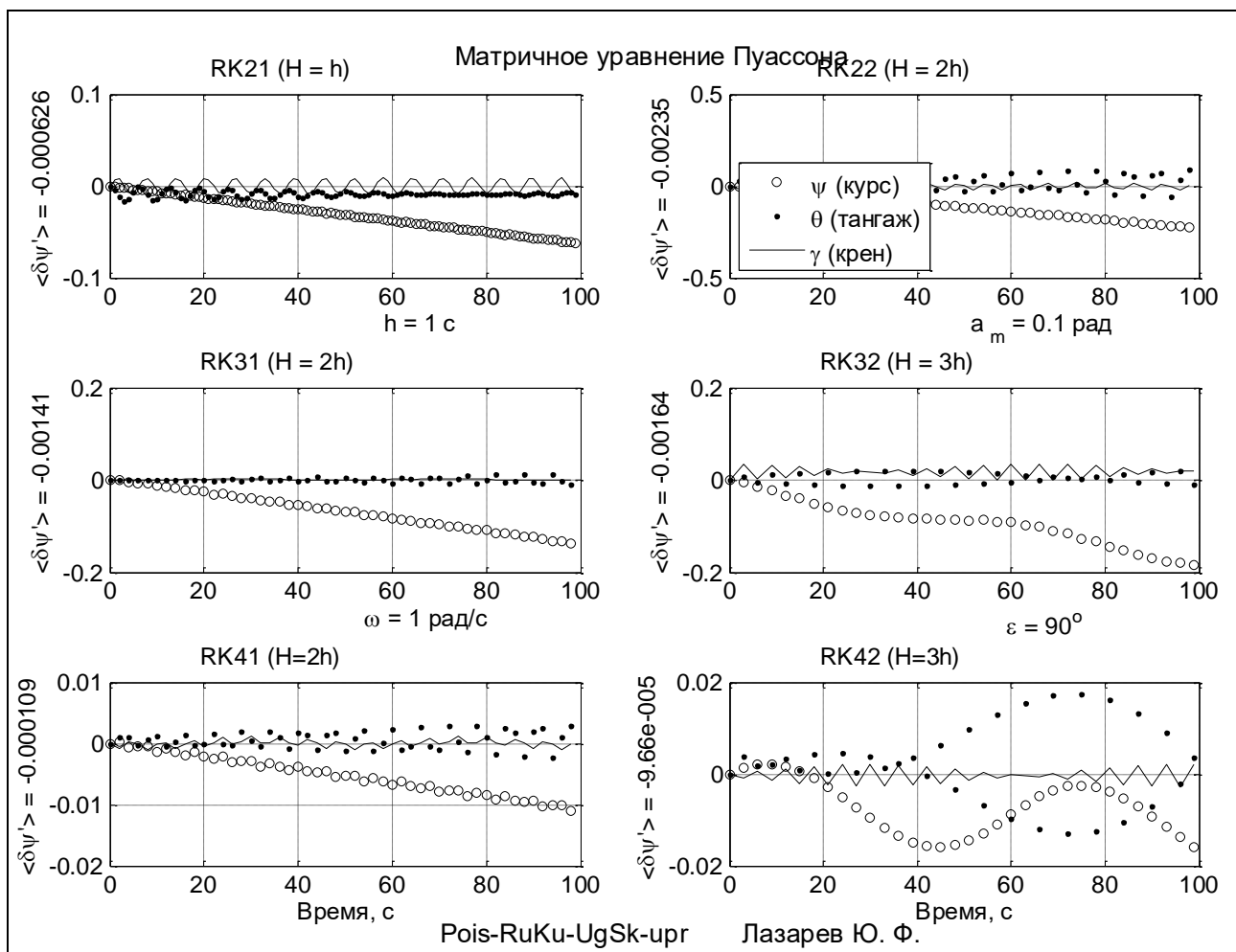
2) існують значення зсувів фаз, за яких дрейф сягає деякої максимальної за модулем величини; саме ці випадки є найбільш небезпечними для практики застосування алгоритмів; тому дрейфи похибок за цих обставин мають бути прийняті за характеристики точності алгоритмів БІСО; з рис. 3.3 і 3.4 випливає,

що алгоритми методів RK21, RK22, RK41 і RK42 мають максимальний за величиною дрейф при зсуві фаз $\varepsilon = 90^\circ$, а алгоритми RK31 і RK32 – при $\varepsilon = 0^\circ$;

3) максимальні дрейфи суттєво визначаються величиною кроку опитування; зазвичай ця залежність є степеневою, показник степеня якої характеризує порядок точності застосованого алгоритму; з поданих рисунків, наприклад, випливає, що алгоритми RK21 і RK22 мають другий порядок точності, RK31 і RK32 – третій, а RK41 і RK42 – четвертий порядок точності.



а) зсув фаз 0°



б) зсув фаз 90°

Рис. 3.2. Крок опитування 1 с

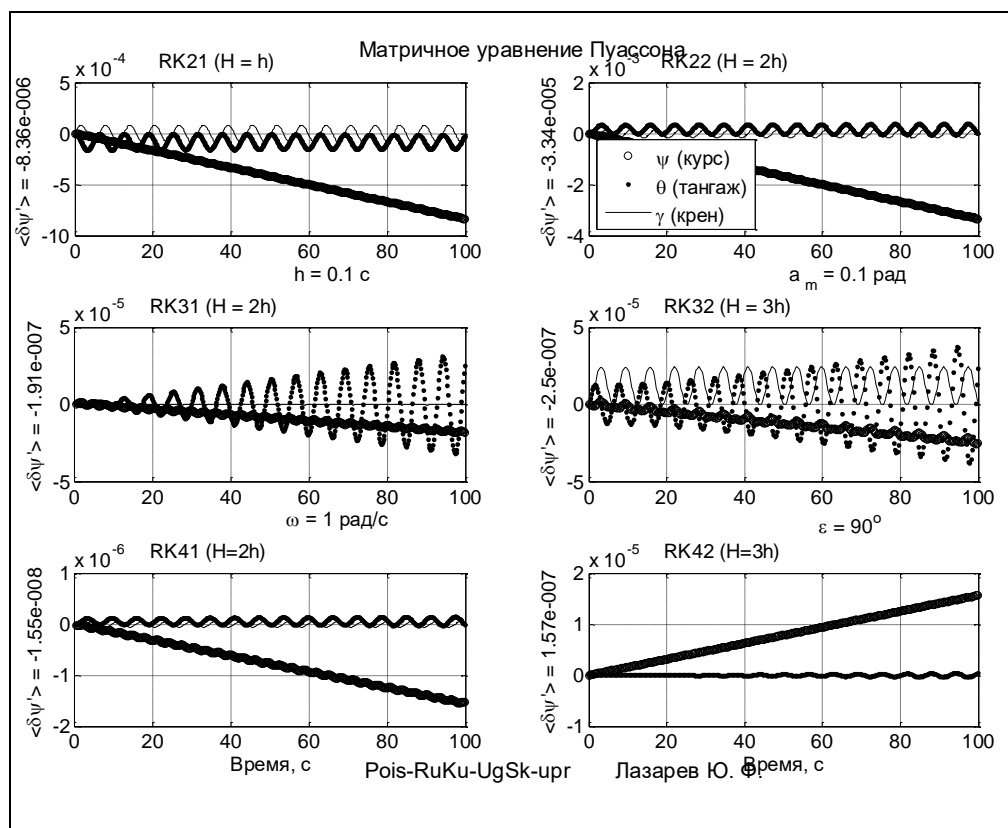
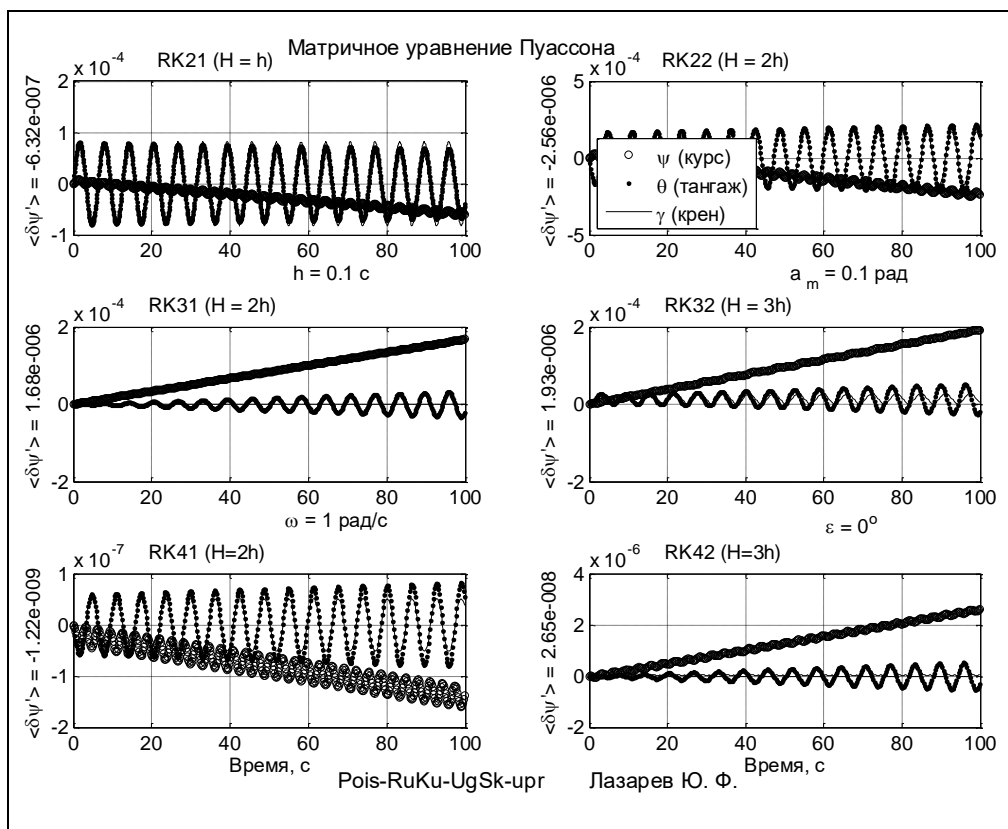


Рис. 3.3. Крок опитування 0,1 с

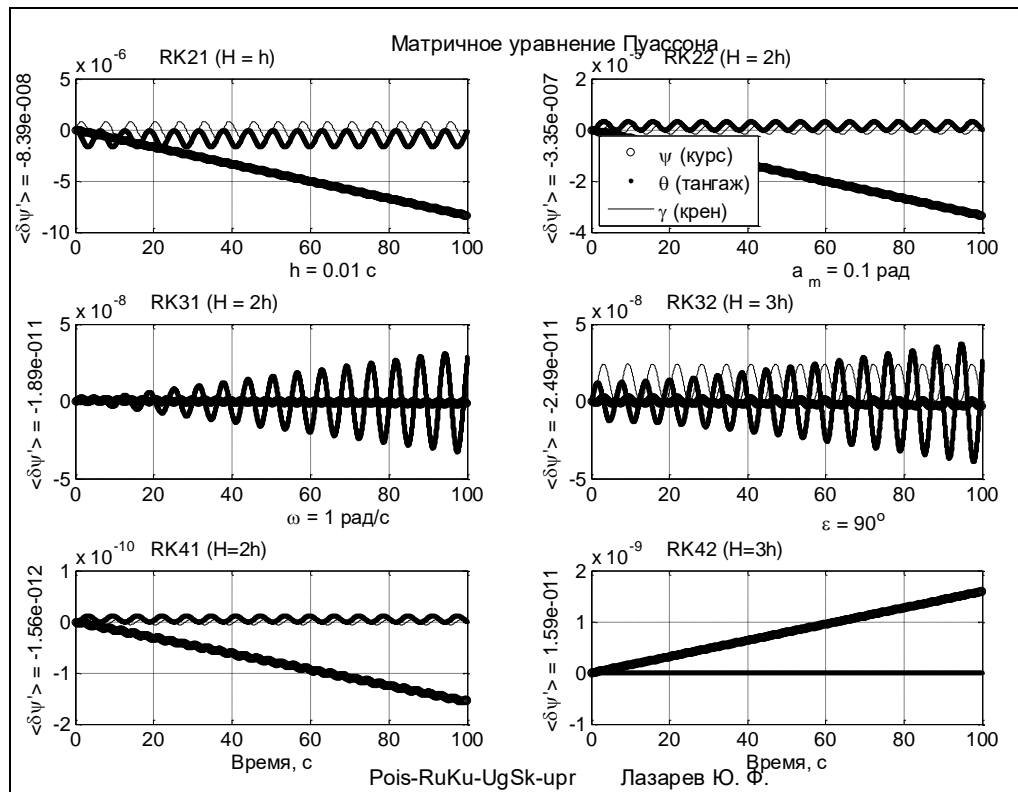
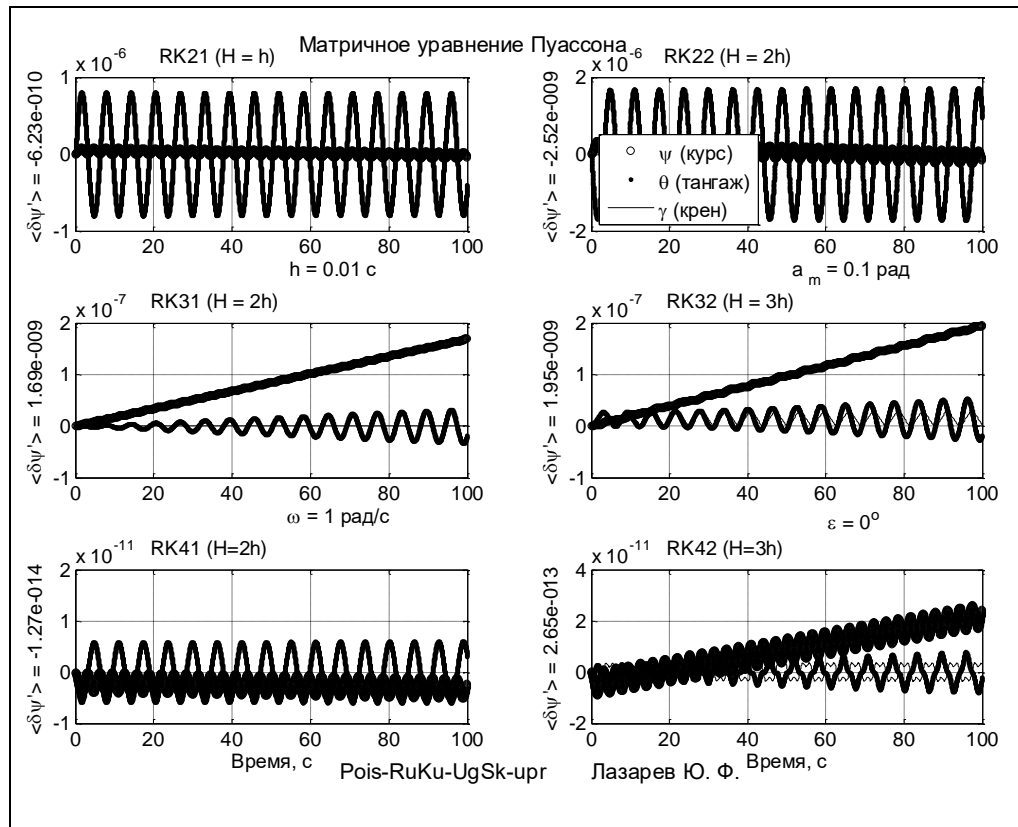


Рис. 3.4. Крок опитування 0,01 с

3.4. Про форму подання характеристик точності алгоритмів

Побудуємо у подвійному логарифмічному масштабі графіки залежності максимальних дрейфів досліджених шести алгоритмів від кроку опитування (рис. 3.5).

Дані про максимальні дрейфи наведені у Таблиці 3.1

Таблиця 3.1

Максимальні дрейфи різних методів (радіани у секунду)

Алгоритм	h, c		
	1	0,1	0,01
RK21	6.26e-4	8.36e-6	8.39e-8
RK22	2.35e-3	3.34e-5	3.35e-7
RK31	6.78e-4	1.68e-6	1.69e-9
RK32	4.92e-4	1.93e-6	1.95e-9
RK41	1.09e-4	1.55e-8	1.56e-12
RK42	9.66e-5	1.57e-7	1.59e-11

Для побудови відповідних графіків скористаємося програмою **Grafiki_NP1**, текст якої наведений нижче

```
% Grafiki_NP1
clear all, clc
h=[1 0.1 0.01];
D21=[6.26e-4 8.36e-6 8.39e-8];
D22=[2.35e-3 3.34e-5 3.35e-7];
D31=[6.78e-4 1.68e-6 1.69e-9];
D32=[4.92e-4 1.93e-6 1.95e-9];
D41=[1.09e-4 1.55e-8 1.56e-12];
D42=[9.66e-5 1.57e-7 1.59e-11];
loglog(h,D21,'o:',h,D22,'x-',h,D31,'o:',h,D32,'x-',h,D41,'o:',h,D42,'x-',...
'LineWidth',3), grid
```



```

set(gca,'FontSize',13)
title('Залежність дрейфів від кроку')
xlabel('Крок опитування h, c')
ylabel('Максимальний дрейф, c^{-1}')
legend('RK21','RK22','RK31','RK32','RK41','RK42',4)

```

Результат показано на рисунку 3.5.

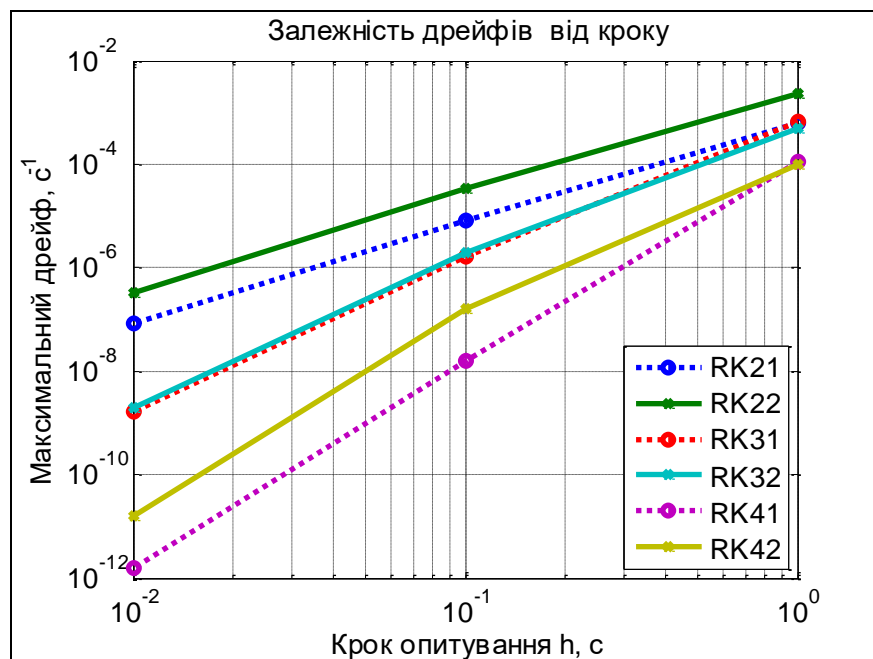


Рис. 3.5. Залежності дрейфу від кроку

Додаткові дослідження за створеною програмною моделлю показують, що максимальні дрейфи $\left\langle \frac{d(\Delta\psi)}{dt} \right\rangle_{\max}$ будь-якого алгоритму мають такі особливості:

- 1) вони пропорційні добутку амплітуд ϑ_m, φ_m коливання основи;
- 2) вони пропорційні частоті ω коливань основи у степені на одиницю вищій за порядок точності N відповідного алгоритму.

Все сказане дозволяє описати усі залежності дрейфів від кроку опитування загальною формулою

$$\left\langle \frac{d(\Delta\psi)}{dt} \right\rangle_{\max} = k \cdot \omega^{N+1} \cdot \vartheta_m \cdot \varphi_m \cdot h^N. \quad (3.1)$$

Однак, набагато зручніше користуватися безрозмірними величинами. В якості першої оберемо величину δ відносного (по відношенню до величини $\omega \vartheta_m \varphi_m$) максимального дрейфу

$$\delta = \frac{1}{\omega \vartheta_m \varphi_m} \left\langle \left| \frac{d(\Delta\psi)}{dt} \right| \right\rangle_{\max}. \quad (3.2)$$

Другою безрозмірною величиною приймемо так званий "частотний параметр":

$$\mu = \omega \cdot h. \quad (3.3)$$

Тоді формулу (3.1) можна подати у спрощеному вигляді

$$\delta = k \cdot \mu^N, \quad (3.4)$$

і характеризувати точність будь якого алгоритму лише двома параметрами:

- порядком точності N ;
- величиною коефіцієнта k .

У цих нових змінних графіки рис. 3.5 набудуть вигляду, наведеному на рисунку 3.6.

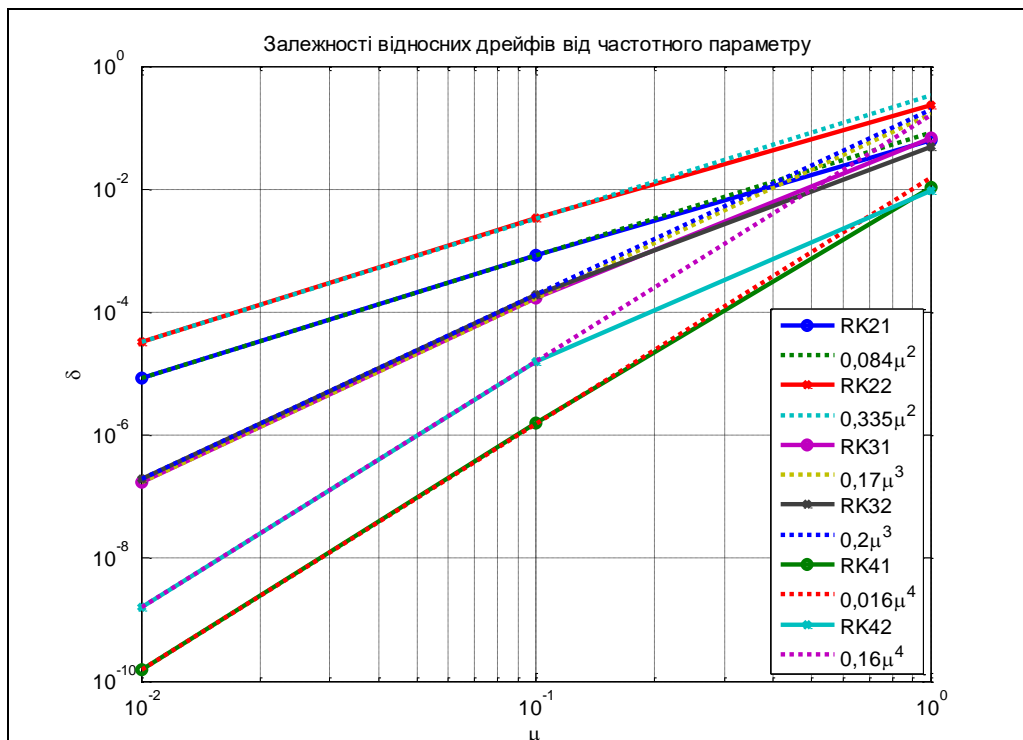


Рис. 3.6. Залежності відносних дрейфів від частотного параметра

Там же наведені (пунктиром) апроксимувальні прямі для кожної з побудованих залежностей, які достатньо добре збігаються з експериментальними у діапазоні $\mu \leq 0,1$.

Резюмуючи, досліджені алгоритми можна охарактеризувати параметрами, що наведені у Таблиці 3.2:

Таблиця 3.2. Параметри точності різних алгоритмів

Алгоритм	k	N
RK21	0,084	2
RK22	0,335	2
RK31	0,169	3
RK32	0,195	3
RK41	0,016	4
RK42	0.16	4

Отже, для виявлення параметрів точності алгоритмів БІСО слід на основі комп'ютерного моделювання будувати графіки залежності відносних максимальних дрейфів похибок цих алгоритмів при синхронних коливаннях основи навколо двох ортогональних осей від частотного параметра, а на цьому ґрунті знайти двійку параметрів, що визначають знайдену степеневу залежність.

3.5. Про точність деяких відомих алгоритмів

Вивчимо за викладеним сценарієм характеристики точності відомих з літератури алгоритмів БІСО. Перш за все це алгоритми з широко відомої книги А. П. Панова [5]. Йменуватимемо їх номерами відповідних формул у книзі.

Розглянемо низку чотирикрокових безрозгінних алгоритмів (3.3.19), (3.3.20), (3.3.22), (3.3.24), (3.3.26), (3.3.28), (3.3.30). Їх розрахункові формули наведені у Таблиці 3.3.

Таблиця 3.3. Формули розрахунків за алгоритмами

Алгоритм	$\Phi_{N+4} = \mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3 + \mathbf{q}_4 + \mathbf{F}$ Формула \mathbf{F}
(3.3.19)	$\frac{2}{3}[(\mathbf{q}_1 \times) + (\mathbf{q}_2 \times)](\mathbf{q}_3 + \mathbf{q}_4)$
(3.3.20)	$\frac{4}{3}[(\mathbf{q}_1 \times)\mathbf{q}_3 + (\mathbf{q}_2 \times)\mathbf{q}_4]$
(3.3.22)	$\frac{1}{3}[(\mathbf{q}_1 \times)\mathbf{q}_3 + (\mathbf{q}_2 \times)\mathbf{q}_4] - \frac{2}{15}[(\mathbf{q}_1 \times) - (\mathbf{q}_2 \times)](\mathbf{q}_3 - \mathbf{q}_4)$
(3.3.24)	$\frac{2}{3}[(\mathbf{q}_1 \times) + (\mathbf{q}_2 \times)](\mathbf{q}_3 + \mathbf{q}_4) + \frac{32}{45}[(\mathbf{q}_1 \times) - (\mathbf{q}_2 \times)](\mathbf{q}_3 - \mathbf{q}_4)$
(3.3.26)	$\frac{2}{3}[(\mathbf{q}_1 \times)\mathbf{q}_3 + (\mathbf{q}_2 \times)\mathbf{q}_4] + \frac{1}{2}[(\mathbf{q}_1 \times) + (\mathbf{q}_2 \times)](\mathbf{q}_3 + \mathbf{q}_4)$
(3.3.28)	$\frac{2}{3}[(\mathbf{q}_1 \times)\mathbf{q}_3 + (\mathbf{q}_2 \times)\mathbf{q}_4] + \frac{1}{2}[(\mathbf{q}_1 \times) + (\mathbf{q}_2 \times)](\mathbf{q}_3 + \mathbf{q}_4) + \frac{1}{30}[(\mathbf{q}_1 \times) - (\mathbf{q}_2 \times)](\mathbf{q}_3 - \mathbf{q}_4)$
(3.3.30)	$\frac{2}{3}[(\mathbf{q}_1 \times) + (\mathbf{q}_2 \times)](\mathbf{q}_3 + \mathbf{q}_4) + \frac{8}{15}[(\mathbf{q}_1 \times) - (\mathbf{q}_2 \times)](\mathbf{q}_3 - \mathbf{q}_4)$

Тут використані позначення:

Φ_{N+4} – приріст вектора Ейлера за крок інтегрування, починаючи з моменту часу t_N ;

\mathbf{q}_m – вектор-стовпець з проекцій квазікоординат, виміряних за m -ний крок опитування з початку інтервалу кроку інтегрування;

$(\mathbf{q}_m \times)$ – кососиметрична матриця, побудована на основі елементів вектора \mathbf{q}_m .

Результати досліджень дрейфів цих алгоритмів у графічній формі наведені на рисунку 3.7.

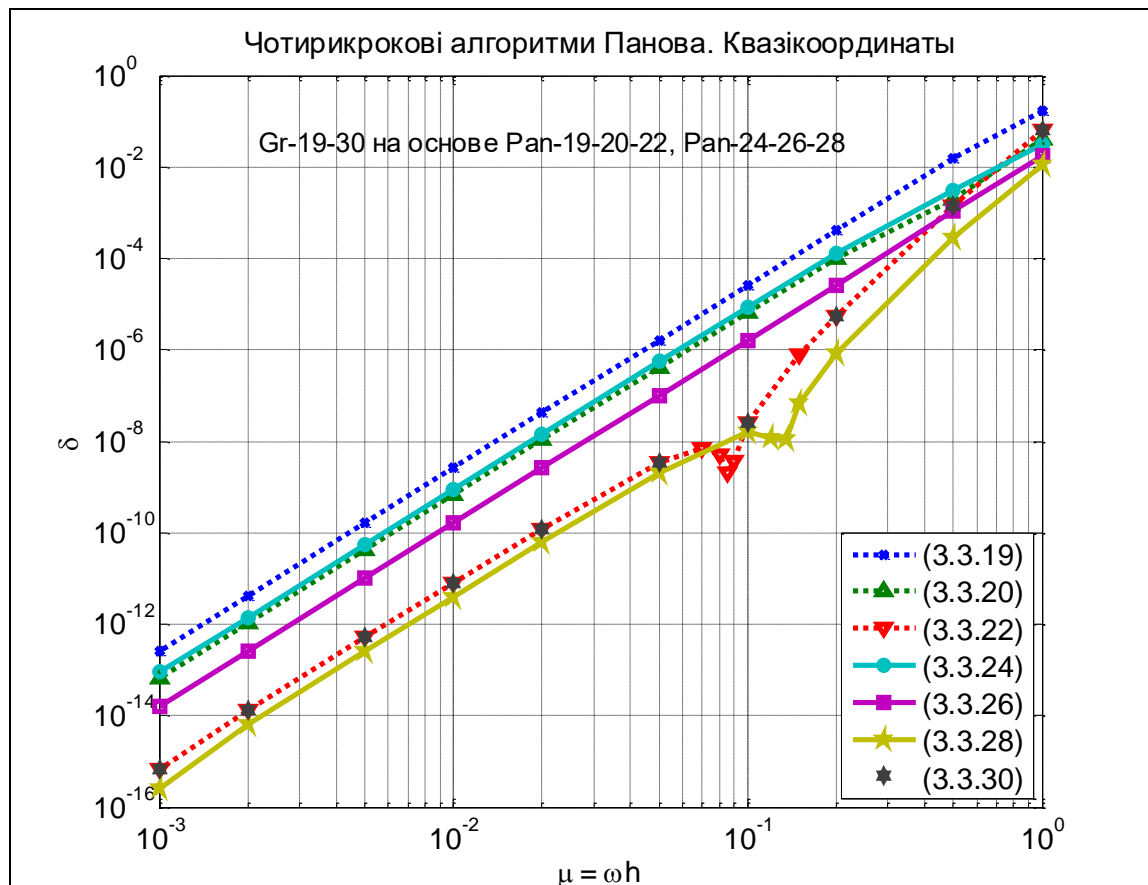


Рис. 3.7. Точнісні характеристики чотирикрокових алгоритмів з книги А.П. Панова [5]

На цій основі знайдені параметри точності, зведені у Таблицю 3.4.

Таблиця 3.4. Параметри точності алгоритмів

Алгоритм	Кроковість	N	Коефіцієнт k
(3.3.19)	4	4	0,26
(3.3.20)	4	4	0,067
(3.3.22)	4	4	0,00066
(3.3.24)	4	4	0,089
(3.3.26)	4	4	0,016
(3.3.28)	4	4	0,00026
(3.3.30)	4	4	0,00066

Як можна впевнитися, найбільш точним виявляється алгоритм (3.3.28). Алгоритми (3.3.22) і (3.3.30) мають однакову точність у всьому діапазоні змінювання частотного параметру, гіршу у 2,5 рази за попередню.

Усі ці три найточніші алгоритми в діапазоні великих значень частотного параметру ($1 \geq \mu \geq 0,1$) мають шостий порядок точності.

4. Синтез алгоритмів на основі методів Рунге-Кутти

Методи Рунге-Кутти можна пристосувати для побудови алгоритмів БІСО, дещо їх трансформували.

У прикладній математиці [2, с 166-176] методи Рунге-Кутти вважаються однокроковими, бо розраховують значення вектора фазових змінних на наступному кроці інтегрування. При цьому вони використовують обчислення правих частин диференціальних рівнянь у формі Коши у кількох моментах часу всередині кроку інтегрування. Останнє неможливо в алгоритмах БІСО через те, що значення вектору кутової швидкості досяжні лише в окремі моменти часу, розділені інтервалом опитування вимірювачів.

Тому алгоритми методів Рунге-Кутти слід переформатувати у багатокрокові, тобто такі, в яких крок інтегрування складається з кількох кроків опитування в залежності від кількості використовуваних проміжних моментів часу.

Іншою особливістю алгоритмів БІСО є те, що обчислювання вектора фазових змінних наприкінці кроку інтегрування здійснюється тоді, коли крок інтегрування вже завершився, тобто вже відоме нове значення вектора кутової швидкості основи. Тому вже однокроковий алгоритм БІСО може спиратися на знання значення правих частин диференціальних рівнянь у два моменти часу – на початку і наприкінці кроку опитування. У цьому випадку однокроковий алгоритм БІСО може бути побудований на основі модифікованого метода Ейлера.

Для багатокрокових алгоритмів БІСО введемо окрім кроку опитування h ще поняття кроку інтегрування H . Крок інтегрування може складатися з одного кроку опитування $H = h$. Це характерно для так званих розгінних багатокрокових

методів, алгоритми яких спираються на дані про кутову швидкість у кількох попередніх моментах часу, тобто потребують попереднього "розгону".

Алгоритми ж, основані на методах Рунге-Кутти, є безрозгінними (самостартувальними). Вони використовують лише ті значення вимірюваної кутової швидкості, які виходять на наступних кількох кроках опитування. В них здійснюється визначення фазових змінних через декілька кроків опитування, тобто $H = m \cdot h$, де m – кількість кроків опитування у одному кроці інтегрування.

4.1. Головні формули

Загальна формула чисельного інтегрування на основі методів Рунге-Кутти така

$$\mathbf{y}_{n+m} = \mathbf{y}_n + H \cdot \mathbf{F}(t_n; \mathbf{y}_n), \quad (4.1)$$

де \mathbf{y}_n – вектор значень фазових змінних у момент часу t_n ; n – номер початкового кроку опитування; $\mathbf{F}(t_n; \mathbf{y}_n)$ – вектор, що є певною функцією значень правої частини $\mathbf{Z}(t_j; \mathbf{y}_j)$; ($j = n, n+1, \dots, n+m$) диференціальних рівнянь у формі Коши:

$$\frac{d\mathbf{y}}{dt} = \mathbf{Z}(t; \mathbf{y}). \quad (4.2)$$

Значення функції $\mathbf{F}(t_n; \mathbf{y}_n)$ визначаються формулами, наведеними у Таблиці 4.1.

Формули у такому поданні не відрізняються від звичайних. Відмінність лише у тому, що крок інтегрування має складатися з кількох кроків опитування. Алгоритм *RK-21* є однокроковим другого порядку. Двокрокові алгоритми *RK-22*, *RK-32* і *RK-42* мають відповідно другий, третій і четвертий порядок точності. Трикрокові алгоритми *RK-33* і *RK-43* відносяться до третього і четвертого порядків відповідно.

Таблиця 4.1. Алгоритми БІСО на основі методів Рунге-Кутти

$$\mathbf{y}_{n+m} = \mathbf{y}_n + H \cdot \mathbf{F}(t_n; \mathbf{y}_n); \quad \mathbf{k}_1 = \mathbf{Z}(t_n; \mathbf{y}_n)$$

Назва	Формула	Допоміжні величини
RK-21 ($H = h$)	$\mathbf{F} = (\mathbf{k}_1 + \mathbf{k}_2) / 2$	$\mathbf{k}_2 = \mathbf{Z}(t_n + H; \mathbf{y}_n + H\mathbf{k}_1)$
RK-22 ($H = 2h$)	$\mathbf{F} = \mathbf{k}_2$	$\mathbf{k}_2 = \mathbf{Z}(t_n + H / 2; \mathbf{y}_n + H\mathbf{k}_1 / 2)$
RK-32 ($H = 2h$)	$\mathbf{F} = (\mathbf{k}_1 + 4\mathbf{k}_2 + \mathbf{k}_3) / 2$	$\mathbf{k}_2 = \mathbf{Z}(t_n + H / 2; \mathbf{y}_n + H\mathbf{k}_1 / 2);$ $\mathbf{k}_3 = \mathbf{Z}(t_n + H; \mathbf{y}_n + H(2\mathbf{k}_2 - \mathbf{k}_1))$
RK-33 ($H = 3h$)	$\mathbf{F} = (\mathbf{k}_1 + 3\mathbf{k}_3) / 4$	$\mathbf{k}_2 = \mathbf{Z}(t_n + H / 3; \mathbf{y}_n + H\mathbf{k}_1 / 3);$ $\mathbf{k}_3 = \mathbf{Z}(t_n + 2H / 3; \mathbf{y}_n + 2H\mathbf{k}_2 / 3)$
RK-42 ($H = 2h$)	$\mathbf{F} = (\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) / 6$	$\mathbf{k}_2 = \mathbf{Z}(t_n + H / 2; \mathbf{y}_n + H\mathbf{k}_1 / 2);$ $\mathbf{k}_3 = \mathbf{Z}(t_n + H / 2; \mathbf{y}_n + H\mathbf{k}_2 / 2);$ $\mathbf{k}_4 = \mathbf{Z}(t_n + H; \mathbf{y}_n + H\mathbf{k}_3)$
RK-43 ($H = 3h$)	$\mathbf{F} = (\mathbf{k}_1 + 3\mathbf{k}_2 + 3\mathbf{k}_3 + \mathbf{k}_4) / 8$	$\mathbf{k}_2 = \mathbf{Z}(t_n + H / 3; \mathbf{y}_n + H\mathbf{k}_1 / 3);$ $\mathbf{k}_3 = \mathbf{Z}(t_n + 2H / 3; \mathbf{y}_n + H(\mathbf{k}_2 - \mathbf{k}_1 / 3));$ $\mathbf{k}_4 = \mathbf{Z}(t_n + H; \mathbf{y}_n + H(\mathbf{k}_1 + \mathbf{k}_3 - \mathbf{k}_2))$

Наведемо текст можливого варіанту процедури інтегрування трикроковим алгоритмом четвертого порядку:

```
function [tout,yout]=RK_43(Zpfun,h,t,y)
% Процедура трикрокового (чотириточкового) алгоритму
% метода Рунге-Кутти четвертого порядку
% Zpfun - символічна змінна, що містить ім'я використовуваної процедури
% обчислення значення правої частини інтегрованої системи
% диференціальних рівнянь, що подана у формі Коши
% h - крок інтегрування
% t - момент часу чергового кроку інтегрування
% y - значення вектора фазових змінних у момент часу t
```

```

% tout - момент кінця чергового кроку інтегрування
% yout - вектор фазових змінних у кінці кроку інтегрування
% Лазарєв Ю. Ф.          06-09-2015
k1=feval(Zpfun,t,y);
k2=feval(Zpfun,t+h/3,y+h*k1/3);
k3=feval(Zpfun,t+2*h/3,y+h*(k2-k1/3));
k4=feval(Zpfun,t+h,y+h*(k1-k2+k3));
F=(k1+3*k2+3*k3+k4)/8;
yout=y+h*F;   tout=t+h;

```

При використанні цього алгоритму потрібно попередньо сформувати значення кроку інтегрування за відомим значенням кроку h опитування **$H=3*h$** ; і звертатися до процедури метода у такий спосіб

```

[tout,yout]=RK_43('ім'я процедури правих частин',H,t,y);

```

4.2. Завдання на проведення досліджень

Кожному студенту дається окреме завдання (див. Таблицю 4.2) на дослідження, в якому вказується:

- 1) вид рівняння орієнтації, яке інтегрується в обчислювачі БІСО;
- 2) три види алгоритмів, що досліджуватимуться на точність.

Таблиця 4.2. Варіанти завдань

Варіант	Вид рівняння орієнтації	Види алгоритмів Рунге-Кутти
1	Кінематичні Ейлера	21; 31; 41
2	Пуассона	21; 31; 41
3	Кватерніонне	21; 31; 41
4	Векторне Ейлера	21; 31; 41
5	Векторне Гіббса	21; 31; 41
6	Векторне Родріга	21; 31; 41
7	Кінематичні Ейлера	22; 32; 42
8	Пуассона	22; 32; 42
9	Кватерніонне	22; 32; 42
10	Векторне Ейлера	22; 32; 42
11	Векторне Гіббса	22; 32; 42
12	Векторне Родріга	22; 32; 42

Завдання полягає в наступному:

- 1) встановити залежність дрейфу похибки визначення куту ψ від зсуву фаз для кожного з заданих алгоритмів при трьох значеннях кроку опитування вимірювачів $h = 1; 0,1; 0,01$ с; побудувати графіки цих залежностей в діапазоні зсуву фаз від -180° до $+180^\circ$ через 30° ;

2) встановити залежність максимального (амплітудного при зсуві фаз) значення дрейфу від амплітуди коливань основи;

3) встановити залежність максимального (амплітудного при зсуві фаз) значення дрейфу від частоти коливань основи;

4) встановити залежність максимального (амплітудного при зсуві фаз) значення дрейфу від кроку опитування вимірювачів для кожного з заданих алгоритмів; побудувати графіки цих залежностей в логарифмічному масштабі при

$$h = 1; 0,4; 0,1; 0,04; 0,01; 0,004; 0,001 \text{ с};$$

5) зробити висновки щодо встановлених залежностей і властивостей дрейфів алгоритмів.

5. Метод Пікара синтезу алгоритмів

Йдитиметься про синтез (знаходженні формул обчислень) алгоритмів безплатформової інерціальної системи орієнтації (БІСО), що є частиною БІНС, шляхом застосування до кінематичного рівняння орієнтації так званого методу послідовних наближень Пікара. Метод Пікара полягає у попередньому знаходженні *наближеного аналітичного* розв'язку диференціального рівняння орієнтації на інтервалі кроку інтегрування та подальшого обчислення за ним параметрів орієнтації наприкінці кроку інтегрування. Аналітичний розв'язок відшукується *послідовними наближеннями*.

5.1. Поліноміальна інтерполяція кутової швидкості

Диференціальні рівняння орієнтації у формі Коші будь-якого виду містять у правій частині невідому функцію часу – вектор кутової швидкості основи. У БІСО ця функція невідома. Відомими є лише значення цієї функції у деякі моменти часу, розділені кроком опитування вимірювачів (якщо для цього використовуються гіротахметри, встановлені на рухомій основі), або значення інтегралів на інтервалах кроків опитування (за використання інтегровальних гіротахметрів). Щоб обчислити за цих значень інтеграли від правих частин рівнянь орієнтації, потрібно апроксимувати вектор $\boldsymbol{\omega} = \mathbf{i}\omega_1 + \mathbf{j}\omega_2 + \mathbf{k}\omega_3$ проекцій кутової швидкості на потрібному інтервалі часу деякою явною функцією часу.

5.1.1. Апроксимація кутової швидкості за вимірювання її проекцій

Нехай безпосередньо вимірюються значення проекцій кутової швидкості основи (тобто вектор ω). Позначимо ω_0 значення цього вектора на початку кроку інтегрування H ($t = t_n$, $\tau = 0$):

$$\omega(0) = \omega_0.$$

1. Якщо синтезується **однокроковий** алгоритм, слід зважити, що у цьому випадку є відомими два значення вектора кутової швидкості – отримані на початку і у кінці єдиного кроку опитування, тому апроксимувати кутову швидкість можна лінійною функцією вигляду:

$$\omega = \omega_0 + e \cdot \tau = \omega_0 + E \cdot \left(\frac{\tau}{h} \right), \quad (5.1)$$

де позначено

$$E = eh. \quad (5.2)$$

У кінці кроку опитування ($\tau = h$) у відповідності до (5.1) матимемо:

$$\omega_1 = \omega(h) = \omega_0 + E.$$

Звідси визначаємо векторний коефіцієнт E :

$$E = \omega_1 - \omega_0.$$

Тому шукана функція часу визначається виразом

$$\boxed{\omega = \omega_0 + (\omega_1 - \omega_0) \cdot \left(\frac{\tau}{h} \right)}, \quad (5.3)$$

2. Якщо розглядати **двокроковий** алгоритм, то у цьому випадку відомими є три значення вектора кутової швидкості на кінцях цих двох кроків опитування. Тому потрібно апроксимувати кутову швидкість квадратним поліномом

$$\omega = \omega_0 + e\tau + f\tau^2 = \omega_0 + E \frac{\tau}{h} + F \left(\frac{\tau}{h} \right)^2, \quad (5.4)$$

де використане позначення

$$\mathbf{F} = \mathbf{f}h^2. \quad (5.5)$$

У відповідності з цим у кінці першого кроку опитування ($\tau = h$) значення кутової швидкості визначиться співвідношенням

$$\boldsymbol{\omega}_1 = \boldsymbol{\omega}_0 + \mathbf{E} + \mathbf{F}. \quad (5.6)$$

Третє значення кутової швидкості на кінці другого кроку опитування ($\tau = 2h$) визначиться виразом:

$$\boldsymbol{\omega}_2 = \boldsymbol{\omega}_0 + 2\mathbf{E}h + 4\mathbf{f}h^2 = \boldsymbol{\omega}_0 + 2\mathbf{E} + 4\mathbf{F}.$$

Звідси впливають вирази для невідомих коефіцієнтів \mathbf{E} та \mathbf{F} :

$$\mathbf{E} = \frac{1}{2}(-3\boldsymbol{\omega}_0 + 4\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2); \quad \mathbf{F} = \frac{1}{2}(\boldsymbol{\omega}_0 - 2\boldsymbol{\omega}_1 + \boldsymbol{\omega}_2).$$

Тепер шукана функція часу визначиться у такий спосіб

$$\boxed{\boldsymbol{\omega} = \boldsymbol{\omega}_0 + \frac{1}{2}(-3\boldsymbol{\omega}_0 + 4\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2)\frac{\tau}{h} + \frac{1}{2}(\boldsymbol{\omega}_0 - 2\boldsymbol{\omega}_1 + \boldsymbol{\omega}_2)\left(\frac{\tau}{h}\right)^2}. \quad (5.7)$$

3. У випадку **трикрокового** алгоритму вимірюваними на кожному кроці інтегрування є чотири значення кутової швидкості, тому вектор кутової швидкості апроксимується кубічним поліномом

$$\boldsymbol{\omega} = \boldsymbol{\omega}_0 + \mathbf{e}\tau + \mathbf{f}\tau^2 + \mathbf{g}\tau^3 = \boldsymbol{\omega}_0 + \mathbf{E}\frac{\tau}{h} + \mathbf{F}\left(\frac{\tau}{h}\right)^2 + \mathbf{G}\left(\frac{\tau}{h}\right)^3. \quad (5.8)$$

де

$$\mathbf{G} = \mathbf{g}h^3. \quad (5.9)$$

Щоб визначити три невідомі коефіцієнти \mathbf{E} , \mathbf{F} , \mathbf{G} необхідно знати значення вектора $\boldsymbol{\omega}$ у три моменти часу: $\boldsymbol{\omega}_1$, $\boldsymbol{\omega}_2$ – у кінці другого кроку та $\boldsymbol{\omega}_3$ – у кінці третього кроку опитування.

Підставимо у вираз (5.8) значення часу $\tau = h, 2h, 3h$. Отримаємо три рівняння

$$\begin{cases} \mathbf{E} + \mathbf{F} + \mathbf{G} = \boldsymbol{\omega}_1 - \boldsymbol{\omega}_0 \\ 2\mathbf{E} + 4\mathbf{F} + 8\mathbf{G} = \boldsymbol{\omega}_2 - \boldsymbol{\omega}_0 \\ 3\mathbf{E} + 9\mathbf{F} + 27\mathbf{G} = \boldsymbol{\omega}_3 - \boldsymbol{\omega}_0 \end{cases}$$

Звідси випливає

$$\begin{aligned} \mathbf{E} &= \frac{1}{6}(-11\boldsymbol{\omega}_0 + 18\boldsymbol{\omega}_1 - 9\boldsymbol{\omega}_2 + 2\boldsymbol{\omega}_3); & \mathbf{F} &= \frac{1}{2}(2\boldsymbol{\omega}_0 - 5\boldsymbol{\omega}_1 + 4\boldsymbol{\omega}_2 - \boldsymbol{\omega}_3); \\ \mathbf{G} &= \frac{1}{6}(-\boldsymbol{\omega}_0 + 3\boldsymbol{\omega}_1 - 3\boldsymbol{\omega}_2 + \boldsymbol{\omega}_3). \end{aligned} \quad (5.10)$$

Отже функція часу матиме вигляд

$$\boxed{\boldsymbol{\omega} = \boldsymbol{\omega}_0 + \frac{1}{6}(-11\boldsymbol{\omega}_0 + 18\boldsymbol{\omega}_1 - 9\boldsymbol{\omega}_2 + 2\boldsymbol{\omega}_3)\frac{\tau}{h} + \frac{1}{2}(2\boldsymbol{\omega}_0 - 5\boldsymbol{\omega}_1 + 4\boldsymbol{\omega}_2 - \boldsymbol{\omega}_3)\left(\frac{\tau}{h}\right)^2 + \frac{1}{6}(-\boldsymbol{\omega}_0 + 3\boldsymbol{\omega}_1 - 3\boldsymbol{\omega}_2 + \boldsymbol{\omega}_3)\left(\frac{\tau}{h}\right)^3}. \quad (5.11)$$

4. На **чотирьох кроках** виміряними є п'ять значень кутової швидкості, тому вектор кутової швидкості апроксимуватимемо поліномом четвертого порядку

$$\boldsymbol{\omega} = \boldsymbol{\omega}_0 + \mathbf{e}\tau + \mathbf{f}\tau^2 + \mathbf{g}\tau^3 + \mathbf{c}\tau^4 = \boldsymbol{\omega}_0 + \mathbf{E}\frac{\tau}{h} + \mathbf{F}\left(\frac{\tau}{h}\right)^2 + \mathbf{G}\left(\frac{\tau}{h}\right)^3 + \mathbf{H}\left(\frac{\tau}{h}\right)^4, \quad (5.12)$$

де

$$\mathbf{H} = \mathbf{c} \cdot h^4. \quad (5.13)$$

Для визначення чотирьох невідомих коефіцієнтів **E**, **F**, **G** та **H**, окрім значення $\boldsymbol{\omega}_0$ у початковий момент є відомими ще $\boldsymbol{\omega}_1$, $\boldsymbol{\omega}_2$, $\boldsymbol{\omega}_3$, $\boldsymbol{\omega}_4$ – на кінцях наступних чотирьох кроків опитування.

Підставимо у праву частину (5.12) значення часу $\tau = h, 2h, 3h, 4h$. Дійдемо чотирьох рівнянь

$$\begin{aligned} \mathbf{E} + \mathbf{F} + \mathbf{G} + \mathbf{H} &= \boldsymbol{\omega}_1 - \boldsymbol{\omega}_0; & 2\mathbf{E} + 4\mathbf{F} + 8\mathbf{G} + 16\mathbf{H} &= \boldsymbol{\omega}_2 - \boldsymbol{\omega}_0; \\ 3\mathbf{E} + 9\mathbf{F} + 27\mathbf{G} + 81\mathbf{H} &= \boldsymbol{\omega}_3 - \boldsymbol{\omega}_0; & 4\mathbf{E} + 16\mathbf{F} + 64\mathbf{G} + 256\mathbf{H} &= \boldsymbol{\omega}_4 - \boldsymbol{\omega}_0. \end{aligned} \quad (5.14)$$

З яких знайдемо

$$\begin{aligned}
\mathbf{E} &= \frac{1}{12}(-25\omega_0 + 48\omega_1 - 36\omega_2 + 16\omega_3 - 3\omega_4); \\
\mathbf{F} &= \frac{1}{24}(35\omega_0 - 104\omega_1 + 114\omega_2 - 56\omega_3 + 11\omega_4); \\
\mathbf{G} &= \frac{1}{12}(-5\omega_0 + 18\omega_1 - 24\omega_2 + 14\omega_3 - 3\omega_4); \\
\mathbf{H} &= \frac{1}{24}(\omega_0 - 4\omega_1 + 6\omega_2 - 4\omega_3 + \omega_4).
\end{aligned} \tag{5.15}$$

Перелік отриманих результатів зведено у Таблицю 5.1.

Таблиця 5.1. Коефіцієнти апроксимації за вимірювання кутової швидкості

m	Виміряні величини	Апроксимува- льна функція $\omega(\tau)$	Коефіцієнти $\mathbf{E} = e h$; $\mathbf{F} = f h^2$; $\mathbf{G} = g h^3$; $\mathbf{H} = c h^4$
1	ω_0, ω_1	$\omega_0 + e\tau$	$\mathbf{E} = \omega_1 - \omega_0$
2	$\omega_0, \omega_1,$ ω_2	$\omega_0 + e\tau + f\tau^2$	$\mathbf{E} = (-3\omega_0 + 4\omega_1 - \omega_2)/2$ $\mathbf{F} = (\omega_0 - 2\omega_1 + \omega_2)/2$
3	$\omega_0, \omega_1,$ ω_2, ω_3	$\omega_0 + e\tau + f\tau^2 + g\tau^3$	$\mathbf{E} = (-11\omega_0 + 18\omega_1 - 9\omega_2 + 2\omega_3)/6$ $\mathbf{F} = (2\omega_0 - 5\omega_1 + 4\omega_2 - \omega_3)/2$ $\mathbf{G} = (-\omega_0 + 3\omega_1 - 3\omega_2 + \omega_3)/6$
4	$\omega_0, \omega_1,$ $\omega_2, \omega_3,$ ω_4	$\omega_0 + e\tau + f\tau^2 +$ $+ g\tau^3 + c\tau^4$	$\mathbf{E} = (-25\omega_0 + 48\omega_1 - 36\omega_2 + 16\omega_3 - 3\omega_4)/12$ $\mathbf{F} = (35\omega_0 - 104\omega_1 + 114\omega_2 - 56\omega_3 + 11\omega_4)/24$ $\mathbf{G} = (-5\omega_0 + 18\omega_1 - 24\omega_2 + 14\omega_3 - 3\omega_4)/12$ $\mathbf{H} = (\omega_0 - 4\omega_1 + 6\omega_2 - 4\omega_3 + \omega_4)/24$

5.1.2. Апроксимація кутової швидкості за вимірювання приростів квазікоординат

Розглянемо випадок вимірювання на кожному кроці опитування величин приростів квазікоординат, тобто інтегралів від проекцій кутової швидкості основи на її координатні осі.

Прирости квазікоординат на кроці опитування h позначатимемо у такий спосіб

$$\mathbf{q}_k = \int_{(k-1)h}^{kh} \boldsymbol{\omega}(\xi) d\xi. \quad (5.16)$$

1. У випадку **однокрокового** алгоритму на кожному кроці інтегрування відомою є лише одна величина \mathbf{q}_1 . Тому на цьому кроці кутову швидкість можна апроксимувати лише постійною величиною, тобто функцією

$$\boldsymbol{\omega}(\tau) = \boldsymbol{\omega}_o = \text{const}.$$

Позначимо

$$\mathbf{A} = \boldsymbol{\omega}_o h.$$

Тому можна записати

$$\boldsymbol{\omega}(\tau) = \boldsymbol{\omega}_o = \frac{1}{h} \mathbf{A}. \quad (5.17)$$

Інтегруючи цю функцію у межах єдиного кроку опитування, отримаємо

$$\mathbf{q}_1 = \int_0^h \boldsymbol{\omega}(\xi) d\xi = \int_0^h \frac{1}{h} \mathbf{A} d\xi = \mathbf{A},$$

звідси матимемо

$$\mathbf{A} = \mathbf{q}_1. \quad (5.18)$$

Отже замість (5.17) матимемо

$$\boxed{\omega(\tau) = \frac{1}{h} \mathbf{q}_1.} \quad (5.19)$$

2. У **двокрокових** алгоритмах на кожному кроці інтегрування є відомими два значення вимірюваної величини – \mathbf{q}_1 першого кроку та \mathbf{q}_2 – другого кроку опитування.

Кутову швидкість у цьому випадку можна апроксимувати лінійною функцією часу

$$\omega(\tau) = \omega_o + 2\varepsilon\tau = \frac{\mathbf{A}}{h} + \frac{2\mathbf{B}}{h^2} \tau, \quad (5.20)$$

де позначено

$$\mathbf{B} = \varepsilon h^2.$$

У цьому випадку маємо

$$\mathbf{q}_1 = \int_0^h \left(\frac{\mathbf{A}}{h} + \frac{2\mathbf{B}}{h^2} \xi \right) d\xi = \mathbf{A} + \mathbf{B}; \quad \mathbf{q}_2 = \int_h^{2h} \left(\frac{\mathbf{A}}{h} + \frac{2\mathbf{B}}{h^2} \xi \right) d\xi = \mathbf{A} + 3\mathbf{B}.$$

Розв'язуючи ці співвідношення, дійдемо невідомих векторних коефіцієнтів:

$$\mathbf{A} = \frac{1}{2} (3\mathbf{q}_1 - \mathbf{q}_2); \quad \mathbf{B} = \frac{1}{2} (\mathbf{q}_2 - \mathbf{q}_1). \quad (5.21)$$

Примітка. Виконується співвідношення

$$2\mathbf{A} + 4\mathbf{B} = \mathbf{q}_1 + \mathbf{q}_2. \quad (5.22)$$

У підсумку матимемо

$$\boxed{\omega(\tau) = \frac{1}{2h} (3\mathbf{q}_1 - \mathbf{q}_2) + \frac{1}{h^2} (\mathbf{q}_2 - \mathbf{q}_1) \tau.} \quad (5.23)$$

3. Якщо синтезувати **трикроковий** алгоритм, виміряними виявляться три значення приростів квазікоординат. Кутову швидкість ввижається можливим апроксимувати квадратним поліномом часу

$$\omega(\tau) = \omega_o + 2\varepsilon\tau + 3a\tau^2 = \frac{\mathbf{A}}{h} + \frac{2\mathbf{B}}{h^2} \tau + \frac{3\mathbf{C}}{h^3} \tau^2. \quad (5.24)$$

де

$$\mathbf{C} = \mathbf{a}h^3.$$

Для визначення трьох невідомих коефіцієнтів можна використати відомі значення приростів квазікоординат – \mathbf{q}_1 першого, \mathbf{q}_2 другого та \mathbf{q}_3 третього кроків опитування.

Розпишемо значення приростів у моменти часу $\tau = h, 2h, 3h$:

$$\mathbf{q}_1 = \int_0^h \left(\frac{\mathbf{A}}{h} + \frac{2\mathbf{B}}{h^2} \xi + \frac{3\mathbf{C}}{h^3} \xi^2 \right) d\xi = \mathbf{A} + \mathbf{B} + \mathbf{C};$$

$$\mathbf{q}_2 = \int_h^{2h} \left(\frac{\mathbf{A}}{h} + \frac{2\mathbf{B}}{h^2} \xi + \frac{3\mathbf{C}}{h^3} \xi^2 \right) d\xi = \mathbf{A} + 3\mathbf{B} + 7\mathbf{C};$$

$$\mathbf{q}_3 = \int_{2h}^{3h} \left(\frac{\mathbf{A}}{h} + \frac{2\mathbf{B}}{h^2} \xi + \frac{3\mathbf{C}}{h^3} \xi^2 \right) d\xi = \mathbf{A} + 5\mathbf{B} + 19\mathbf{C}.$$

Звідси випливають такі значення шуканих коефіцієнтів

$$\mathbf{A} = \frac{1}{6}(2\mathbf{q}_3 - 7\mathbf{q}_2 + 11\mathbf{q}_1); \quad \mathbf{B} = \frac{1}{2}(3\mathbf{q}_2 - 2\mathbf{q}_1 - \mathbf{q}_3); \quad \mathbf{C} = \frac{1}{6}(\mathbf{q}_3 - 2\mathbf{q}_2 + \mathbf{q}_1). \quad (5.25)$$

Примітка. Виконується співвідношення

$$3\mathbf{A} + 9\mathbf{B} + 27\mathbf{C} = \mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3. \quad (5.26)$$

Отже, виходить така функція часу

$$\boxed{\omega(\tau) = \frac{1}{6h}(2\mathbf{q}_3 - 7\mathbf{q}_2 + 11\mathbf{q}_1) + \frac{1}{h^2}(3\mathbf{q}_2 - 2\mathbf{q}_1 - \mathbf{q}_3)\tau + \frac{\mathbf{C}}{2h^3}(\mathbf{q}_3 - 2\mathbf{q}_2 + \mathbf{q}_1)\tau^2}. \quad (5.27)$$

4. Якщо обирається **чотирикроковий** алгоритм, кутову швидкість виявляється можливим апроксимувати кубічним поліномом часу

$$\omega(\tau) = \omega_o + 2\mathbf{e}\tau + 3\mathbf{a}\tau + 4\mathbf{b}\tau^3 = \frac{\mathbf{A}}{h} + \frac{2\mathbf{B}}{h^2}\tau + \frac{3\mathbf{C}}{h^3}\tau^2 + \frac{4\mathbf{D}}{h^4}\tau^3. \quad (5.28)$$

де

$$\mathbf{D} = \mathbf{b}h^4.$$

Для визначення чотирьох невідомих коефіцієнтів використаємо чотири відомі значення приростів квазікоординат $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4$ за чотири кроки опитування.

Знайдемо вирази приростів квазікоординат

$$\mathbf{q}_1 = \int_0^h \left(\frac{\mathbf{A}}{h} + \frac{2\mathbf{B}}{h^2} \xi + \frac{3\mathbf{C}}{h^3} \xi^2 + \frac{4\mathbf{D}}{h^4} \xi^3 \right) d\xi = \mathbf{A} + \mathbf{B} + \mathbf{C} + \mathbf{D};$$

$$\mathbf{q}_2 = \int_h^{2h} \left(\frac{\mathbf{A}}{h} + \frac{2\mathbf{B}}{h^2} \xi + \frac{3\mathbf{C}}{h^3} \xi^2 + \frac{4\mathbf{D}}{h^4} \xi^3 \right) d\xi = \mathbf{A} + 3\mathbf{B} + 7\mathbf{C} + 15\mathbf{D};$$

$$\mathbf{q}_3 = \int_{2h}^{3h} \left(\frac{\mathbf{A}}{h} + \frac{2\mathbf{B}}{h^2} \xi + \frac{3\mathbf{C}}{h^3} \xi^2 + \frac{4\mathbf{D}}{h^4} \xi^3 \right) d\xi = \mathbf{A} + 5\mathbf{B} + 19\mathbf{C} + 65\mathbf{D};$$

$$\mathbf{q}_4 = \int_{3h}^{4h} \left(\frac{\mathbf{A}}{h} + \frac{2\mathbf{B}}{h^2} \xi + \frac{3\mathbf{C}}{h^3} \xi^2 + \frac{4\mathbf{D}}{h^4} \xi^3 \right) d\xi = \mathbf{A} + 7\mathbf{B} + 37\mathbf{C} + 175\mathbf{D}.$$

Розв'язки цієї системи рівнянь зручніше за все шукати за допомогою наступної програми Matlab

```
% K_tu_UgSk
clear all, clc, M=[1 1 1 1; 1 3 7 15; 1 5 19 65; 1 7 37 175]
Q=[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1]
format rational,
X=inv(M)*Q, format short g
```

Результат подано нижче:

```
M =
     1     1     1     1
     1     3     7    15
     1     5    19    65
     1     7    37   175

Q =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

$$X = \begin{array}{cccc} 25/12 & -23/12 & 13/12 & -1/4 \\ -35/24 & 23/8 & -15/8 & 11/24 \\ 5/12 & -13/12 & 11/12 & -1/4 \\ -1/24 & 1/8 & -1/8 & 1/24 \end{array}$$

Отже, значення коефіцієнтів такі:

$$\begin{aligned} \mathbf{A} &= \frac{1}{12}(25\mathbf{q}_1 - 23\mathbf{q}_2 + 13\mathbf{q}_3 - 3\mathbf{q}_4); \quad \mathbf{B} = \frac{1}{24}(-35\mathbf{q}_1 + 69\mathbf{q}_2 - 45\mathbf{q}_3 + 11\mathbf{q}_4); \\ \mathbf{C} &= \frac{1}{12}(5\mathbf{q}_1 - 13\mathbf{q}_2 + 11\mathbf{q}_3 - 3\mathbf{q}_4); \quad \mathbf{D} = \frac{1}{24}(-\mathbf{q}_1 + 3\mathbf{q}_2 - 3\mathbf{q}_3 + \mathbf{q}_4). \end{aligned} \quad (5.29)$$

Примітка. Виконується співвідношення

$$4\mathbf{A} + 16\mathbf{B} + 64\mathbf{C} + 256\mathbf{D} = \mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3 + \mathbf{q}_4. \quad (5.30)$$

Неважко помітити, порівнюючи співвідношення (5.19), (5.22), (5.26) і (5.30), що вони можуть бути узагальнені єдиним виразом

$$\zeta\mathbf{A} + \zeta^2\mathbf{B} + \zeta^3\mathbf{C} + \zeta^4\mathbf{D} = \sum_{k=1}^{\zeta} \mathbf{q}_k, \quad (5.31)$$

де ζ – "кроковість" алгоритму (тобто кількість кроків опитування у кроці інтегрування).

Результати апроксимацій зведені у таблицю 5.2.

Таблиця 5.2. Коефіцієнти апроксимації кутової швидкості за вимірювань квазікоординат

ζ	Виміряні величини	$\omega(\tau)$	Коефіцієнти $A = \omega_o h$; $B = \varepsilon h^2$; $C = ah^3$; $D = bh^4$
1	q_1	ω_o	$A = q_1$
2	q_1 , q_2	$\omega_o + 2\varepsilon\tau$	$A = (3q_1 - q_2)/2$ $B = (q_2 - q_1)/2$
3	q_1 , q_2 , q_3	$\omega_o + 2\varepsilon\tau + 3a\tau^2$	$A = (2q_3 - 7q_2 + 11q_1)/6$ $B = (3q_2 - 2q_1 - q_3)/2$ $C = (q_1 - 2q_2 + q_3)/6$
4	q_1 , q_2 , q_3 , q_4	$\omega_o + 2\varepsilon\tau + 3a\tau^2 + 4b\tau^3$	$A = (25q_1 - 23q_2 + 13q_3 - 3q_4)/12$ $B = (-35q_1 + 69q_2 - 45q_3 + 11q_4)/24$ $C = (5q_1 - 13q_2 + 11q_3 - 3q_4)/12$ $D = (-q_1 + 3q_2 - 3q_3 + q_4)/24$

5.2. Алгоритми на ґрунті векторного рівняння Ейлера

5.2.1. Випадок вимірювання приростів квазікоординат

Обмежимося інтегруванням спрощеного рівняння Борца, нехтуючи останньою складовою правої частини рівняння Борца, тобто членами з подвійними векторними добутками:

$$\frac{d\boldsymbol{\varphi}}{dt} \approx \boldsymbol{\omega} + \frac{1}{2}(\boldsymbol{\varphi} \times) \cdot \boldsymbol{\omega}. \quad (5.32)$$

Покладаючи у рівнянні (5.32) величину $\boldsymbol{\varphi}(\tau)$ у нульовому початковому наближенні рівною нулю і використовуючи загальне подання (5.28) вектора кутової швидкості для чотирикрокового алгоритму, отримаємо рівняння першого наближення:

$$\frac{d\boldsymbol{\varphi}_{(1)}}{d\tau} \approx \boldsymbol{\omega} = \frac{1}{h}(\mathbf{A} + 2\mathbf{B}\zeta + 3\mathbf{C}\zeta^2 + 4\mathbf{D}\zeta^3),$$

розв'язок якого на інтервалі кроку інтегрування має вигляд

$$\boldsymbol{\varphi}_{(1)} = \int_0^\tau \boldsymbol{\omega}(\tau) d\tau = h \int_0^{\frac{\tau}{h}} \boldsymbol{\omega}(\zeta) d\zeta = (\mathbf{A}\zeta + \mathbf{B}\zeta^2 + \mathbf{C}\zeta^3 + \mathbf{D}\zeta^4); \quad \left(\zeta = \frac{\tau}{h}\right). \quad (5.33)$$

Тепер цей розв'язок можна використати для побудови другого наближення рівняння (5.32)

$$\frac{d\boldsymbol{\varphi}_{(2)}}{d\tau} \approx \boldsymbol{\omega} + \frac{1}{2}(\boldsymbol{\varphi}_{(1)} \times) \cdot \boldsymbol{\omega}.$$

Розв'язок останнього рівняння можна подати у вигляді

$$\boldsymbol{\varphi}_{(2)} = \boldsymbol{\varphi}_{(1)} + \Delta\boldsymbol{\varphi}_{(2)}, \quad (5.34)$$

де позначено

$$\Delta\boldsymbol{\varphi}_{(2)} = \frac{h}{2} \int_0^{\frac{\tau}{h}} (\boldsymbol{\varphi}_{(1)} \times) \cdot \boldsymbol{\omega} d\zeta. \quad (5.35)$$

Використовуючи вираз (5.33), маємо

$$\Delta\Phi_{(2)} \approx \frac{1}{2} \int_0^{\frac{\tau}{h}} \left((\mathbf{A}\zeta + \mathbf{B}\zeta^2 + \mathbf{C}\zeta^3 + \mathbf{D}\zeta^4) \times (\mathbf{A} + 2\mathbf{B}\zeta + 3\mathbf{C}\zeta^2 + 4\mathbf{D}\zeta^3) \right) d\zeta. \quad (5.36)$$

З врахування властивостей векторного множення

$$\mathbf{a} \times \mathbf{a} = 0; \quad \mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a},$$

підінтегральний вираз у (5.36) приводиться до вигляду

$$\begin{aligned} & ((\mathbf{A}\zeta + \mathbf{B}\zeta^2 + \mathbf{C}\zeta^3 + \mathbf{D}\zeta^4) \times (\mathbf{A} + 2\mathbf{B}\zeta + 3\mathbf{C}\zeta^2 + 4\mathbf{D}\zeta^3)) = \\ & = 2(\mathbf{A} \times \mathbf{B})\zeta^2 + 3(\mathbf{A} \times \mathbf{C})\zeta^3 + 4(\mathbf{A} \times \mathbf{D})\zeta^4 - (\mathbf{A} \times \mathbf{B})\zeta^2 + 3(\mathbf{B} \times \mathbf{C})\zeta^4 + 4(\mathbf{B} \times \mathbf{D})\zeta^5 - \\ & - (\mathbf{A} \times \mathbf{C})\zeta^3 - 2(\mathbf{B} \times \mathbf{C})\zeta^4 + 4(\mathbf{C} \times \mathbf{D})\zeta^6 - (\mathbf{A} \times \mathbf{D})\zeta^4 - 2(\mathbf{B} \times \mathbf{D})\zeta^5 - 3(\mathbf{C} \times \mathbf{D})\zeta^6 = \\ & = (\mathbf{A} \times \mathbf{B})\zeta^2 + 2(\mathbf{A} \times \mathbf{C})\zeta^3 + [3(\mathbf{A} \times \mathbf{D}) + (\mathbf{B} \times \mathbf{C})]\zeta^4 + 2(\mathbf{B} \times \mathbf{D})\zeta^5 + (\mathbf{C} \times \mathbf{D})\zeta^6 \end{aligned}$$

Тепер можна записати

$$\begin{aligned} \Delta\Phi_{(2)} & \approx \frac{1}{2} \int_0^{\frac{\tau}{h}} \left\{ (\mathbf{A} \times \mathbf{B})\zeta^2 + 2(\mathbf{A} \times \mathbf{C})\zeta^3 + \right. \\ & \left. + [3(\mathbf{A} \times \mathbf{D}) + (\mathbf{B} \times \mathbf{C})]\zeta^4 + 2(\mathbf{B} \times \mathbf{D})\zeta^5 + (\mathbf{C} \times \mathbf{D})\zeta^6 \right\} d\zeta = \\ & = \frac{1}{2} \left[(\mathbf{A} \times \mathbf{B})\frac{\zeta^3}{3} + (\mathbf{A} \times \mathbf{C})\frac{\zeta^4}{2} + [3(\mathbf{A} \times \mathbf{D}) + (\mathbf{B} \times \mathbf{C})]\frac{\zeta^5}{5} + (\mathbf{B} \times \mathbf{D})\frac{\zeta^6}{3} + (\mathbf{C} \times \mathbf{D})\frac{\zeta^7}{7} \right]. \end{aligned} \quad (5.37)$$

Рівняння третього наближення

$$\frac{d\Phi_{(3)}}{d\tau} \approx \boldsymbol{\omega} + \frac{1}{2} (\boldsymbol{\Phi}_{(2)} \times) \cdot \boldsymbol{\omega} = \boldsymbol{\omega} + \frac{1}{2} [(\boldsymbol{\Phi}_{(1)} \times) + (\Delta\boldsymbol{\Phi}_{(2)} \times)] \cdot \boldsymbol{\omega}$$

приводить до інтеграла

$$\boldsymbol{\Phi}_{(3)} = \boldsymbol{\Phi}_{(2)} + \Delta\boldsymbol{\Phi}_{(3)}, \quad (5.38)$$

де використано позначення

$$\Delta\boldsymbol{\Phi}_{(3)} = \frac{1}{2} \int_0^{\frac{\tau}{h}} (\Delta\boldsymbol{\Phi}_{(2)} \times) \boldsymbol{\omega} d\zeta. \quad (5.39)$$

Врахуємо, що інтеграл (5.39) приводить до суми подвійних добутків векторів, тому ним належить в нашій постановці задачі знехтувати, як раніше знехтували третьою складовою у правій частині рівняння Борца:

$$\Delta\boldsymbol{\Phi}_{(3)} = 0.$$

В результаті, вводячи позначення

$$\mathbf{O} = \frac{\zeta^3}{60} (\mathbf{A} \times) (10\mathbf{B} + 15\zeta\mathbf{C} + 18\zeta^2\mathbf{D}); \quad \mathbf{P} = \frac{\zeta^5}{30} (\mathbf{B} \times) (3\mathbf{C} + 5\zeta\mathbf{D});$$

$$\mathbf{R} = \frac{\zeta^7}{14}(\mathbf{C} \times) \mathbf{D}, \quad \left(\zeta = \frac{H}{h} \right), \quad (5.40)$$

и враховуючи співвідношення (5.31), отримаємо остаточно

$$\boxed{\varphi(\zeta) = \sum_{k=1}^{\zeta} \mathbf{q}_k + \mathbf{O} + \mathbf{P} + \mathbf{R}.} \quad (5.41)$$

Однокроковий алгоритм

Якщо розглядати однокроковий алгоритм, то у виразі (5.41) слід покласти $\zeta = 1$, $\mathbf{A} = \mathbf{q}_1$, $\mathbf{B} = \mathbf{C} = \mathbf{D} = 0$, в результаті чого отримаємо

$$\boxed{\varphi(h) = \mathbf{q}_1}. \quad (5.42)$$

Двокроковий алгоритм

У випадку двокрокового алгоритму маємо

$$\zeta = 2; \quad \mathbf{A} = (3\mathbf{q}_1 - \mathbf{q}_2)/2; \quad \mathbf{B} = (\mathbf{q}_2 - \mathbf{q}_1)/2; \quad \mathbf{C} = \mathbf{D} = 0.$$

Тому, у відповідності до виразів (5.40), отримаємо

$$\mathbf{O} = \frac{4}{3}(\mathbf{A} \times) \mathbf{B} = \frac{1}{3}(3\mathbf{q}_1 - \mathbf{q}_2) \times (\mathbf{q}_2 - \mathbf{q}_1) = +\frac{2}{3}(\mathbf{q}_1 \times) \mathbf{q}_2; \quad \mathbf{P} = \mathbf{R} = 0,$$

і з формули (5.41) випливає

$$\varphi(2h) = (2\mathbf{A} + 4\mathbf{B}) + \frac{4}{3}(\mathbf{A} \times) \mathbf{B},$$

або, переходячи до приростів квазікоординат

$$\boxed{\varphi(2h) = \mathbf{q}_1 + \mathbf{q}_2 + \frac{2}{3}(\mathbf{q}_1 \times) \mathbf{q}_2}. \quad (5.43)$$

Отримано алгоритм (3.3.15) з книги Панова А. П. [5, с. 150].

Трикроковий алгоритм

У цьому випадку маємо

$$\begin{aligned} \zeta = 3; \quad \mathbf{A} &= (2\mathbf{q}_3 - 7\mathbf{q}_2 + 11\mathbf{q}_1)/6; \quad \mathbf{B} = (3\mathbf{q}_2 - 2\mathbf{q}_1 - \mathbf{q}_3)/2; \\ \mathbf{C} &= (\mathbf{q}_1 - 2\mathbf{q}_2 + \mathbf{q}_3)/6; \quad \mathbf{D} = 0. \end{aligned}$$

Співвідношення (5.40) приймають форму

$$(2\mathbf{B} + 9\mathbf{C}) = (3\mathbf{q}_2 - 2\mathbf{q}_1 - \mathbf{q}_3) + \frac{3}{2}(\mathbf{q}_1 - 2\mathbf{q}_2 + \mathbf{q}_3) = \frac{1}{2}(\mathbf{q}_3 - \mathbf{q}_1);$$

$$\begin{aligned}\mathbf{O} &= \frac{9}{4}(\mathbf{A} \times)(2\mathbf{B} + 9\mathbf{C}) = \frac{3}{16}(2\mathbf{q}_3 - 7\mathbf{q}_2 + 11\mathbf{q}_1) \times (\mathbf{q}_3 - \mathbf{q}_1) = \\ &= \frac{3}{16}(-7\mathbf{q}_1 \times \mathbf{q}_2 + 13\mathbf{q}_1 \times \mathbf{q}_3 - 7\mathbf{q}_2 \times \mathbf{q}_3); \end{aligned}$$

$$\mathbf{P} = \frac{243}{10}(\mathbf{B} \times)\mathbf{C} = \frac{81}{40}(3\mathbf{q}_2 - 2\mathbf{q}_1 - \mathbf{q}_3) \times (\mathbf{q}_1 - 2\mathbf{q}_2 + \mathbf{q}_3) = \frac{81}{40}(\mathbf{q}_1 \times \mathbf{q}_2 - \mathbf{q}_1 \times \mathbf{q}_3 + \mathbf{q}_2 \times \mathbf{q}_3);$$

$$\mathbf{R} = 0.$$

Отже в результаті співвідношення (5.41) виглядатиме так:

$$\boxed{\varphi(3h) = \mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3 + \frac{57}{80}\mathbf{q}_1 \times \mathbf{q}_2 + \frac{33}{80}\mathbf{q}_1 \times \mathbf{q}_3 + \frac{57}{80}\mathbf{q}_2 \times \mathbf{q}_3.} \quad (5.44)$$

Це збігається з формулою (3.3.17) з книги А. П. Панова [5, с. 151].

Чотирикроковий алгоритм

Щоб отримати формулу чотирикрокового алгоритму, необхідно у виразах (5.40) покласти $\zeta = 4$. В результаті дійдемо

$$\mathbf{O} = \frac{64}{60}(\mathbf{A} \times)(10\mathbf{B} + 60\mathbf{C} + 28\mathbf{D}); \quad \mathbf{P} = \frac{\zeta^5}{30}(\mathbf{B} \times)(3\mathbf{C} + 5\zeta\mathbf{D}); \quad \mathbf{R} = \frac{\zeta^7}{14}(\mathbf{C} \times)\mathbf{D}.$$

Для отримання явного виразу через виміряні прирости квазікоординат використаємо програму Matlab. При цьому слід врахувати значення (5.29) векторних коефіцієнтів, а також те, що результат прийме форму

$$\begin{aligned}\varphi(4h) &= \mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3 + \mathbf{q}_4 + S(1) \cdot (\mathbf{q}_1 \times)\mathbf{q}_2 + S(2) \cdot (\mathbf{q}_1 \times)\mathbf{q}_3 + S(3) \cdot (\mathbf{q}_1 \times)\mathbf{q}_4 + \\ &+ S(4) \cdot (\mathbf{q}_2 \times)\mathbf{q}_3 + S(5) \cdot (\mathbf{q}_2 \times)\mathbf{q}_4 + S(6) \cdot (\mathbf{q}_3 \times)\mathbf{q}_4\end{aligned}, \quad (5.45)$$

де S – вектор (матриця-строка з шести елементів) значень коефіцієнтів при векторних коефіцієнтах.

Нижче наведений текст відповідної програми **ABCD4h**:

```
% ABCD4h
% Програма расчётов векторных коэффициентов для 4-шагового
% алгоритма БИСО
% Лазарев Ю. Ф.      28-08-2016
clc, clear all, format rational
dz=4;
```

```

A=[25 -23 13 -3]/12;   B=[-35 69 -45 11]/24;
C=[5 -13 11 -3]/12;   D=[-1 3 -3 1]/24;
%=====
vO=10*B+15*dz*C+18*dz^2*D;
maO=dz^3/60*A'*vO;   k=0;
for i=1:4
    for j=i+1:4
        k=k+1;   O(k)=maO(i,j)-maO(j,i);
    end
end, O;
%=====
vP=3*C+5*dz*D;
maP=dz^5/30*B'*vP;   k=0;
for i=1:4
    for j=i+1:4
        k=k+1;   P(k)=maP(i,j)-maP(j,i);
    end
end, P;
%=====
maR=dz^7/14*C'*D;   k=0;
for i=1:4
    for j=i+1:4
        k=k+1;   R(k)=maR(i,j)-maR(j,i);
    end
end, R;
h4X=O+P+R

```

Результатом роботи цієї програми є вектор $h4X$ коефіцієнтів при векторних добутках вимірних векторів приростів квазікоординат основи

$h4X = \begin{bmatrix} 736/945 & 334/945 & 526/945 & 218/315 & 334/945 \\ 736/945 \end{bmatrix}$.

Отже, формула алгоритму через прирости квазікоординат набуває вигляду

$$\begin{aligned} \varphi(4h) = & \mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3 + \mathbf{q}_4 + \frac{736}{945}(\mathbf{q}_1 \times) \mathbf{q}_2 + \frac{334}{945}(\mathbf{q}_1 \times) \mathbf{q}_3 + \frac{526}{945}(\mathbf{q}_1 \times) \mathbf{q}_4 + \\ & + \frac{218}{315}(\mathbf{q}_2 \times) \mathbf{q}_3 + \frac{334}{945}(\mathbf{q}_2 \times) \mathbf{q}_4 + \frac{736}{945}(\mathbf{q}_3 \times) \mathbf{q}_4 \end{aligned} \quad (5.46)$$

Ця формула також є відомою, вона подає алгоритм 5 з статті [7, с. 79].

Дослідження похибок алгоритмів

Дослідження похибок синтезованих алгоритмів здійснимо за допомогою комп'ютерного моделювання процесу обробки вимірювальної інформації цими алгоритмами.

Для вивчення похибок синтезованих алгоритмів утворено програму **OPR4h_upr**, текст якої наведений нижче:

```
% OPR4h_upr
% Лазарев Ю. Ф., 30-08-2016
clc, clear all
global sprogram sname Dr
global psit psim omp ep Tt Tm omT eT Gt Gm omG eG
global D2

% Установка параметров движения основания
psit = 0;      Tt = 0;      Gt = 0;
psim = 0;      Tm = 0.1;    Gm = 0.1;
omp = 0;      omT = 1;      omG = 1;
ep = 0*pi/180; eT = 90*pi/180; eG = 0*pi/180;
D2=[psit Tt Gt psim Tm Gm omp omT omG ep eT eG];

% Установка параметров интегрирования
h = 0.1;
TK = 100; tfinal=TK; % Конечный момент интегрирования
sprogram = ' OPR4h-upr'; sname = 'Лазарев Ю. Ф.';
disp(' Вы вошли в программу OPR4h_upr')
disp(' Интегрируется векторное уравнение ориентации Эйлера')
disp(' при следующих значениях параметров: ')
disp(' '),disp('      движения ОСНОВАНИЯ ')
disp([sprintf(' psit = %g ',psit), sprintf(' psim = %g ',psim), ...
      sprintf(' omp = %g ',omp), sprintf(' ep = %g ',ep)])
disp([sprintf(' Tt = %g ',Tt) sprintf(' Tm = %g ',Tm) ...
      sprintf(' omT = %g ',omT) sprintf(' eT = %g ',eT)])
disp([sprintf(' Gt = %g ',Gt) sprintf(' Gm = %g ',Gm) ...
      sprintf(' omG = %g ',omG) sprintf(' eG = %g ',eG)])
disp(' '), disp('      интегрирования')
disp([sprintf(' h = %g ',h) sprintf(' TK = %g ',TK)]);

% БЛОК ИНТЕГРУВАННЯ
t=0; % Задание начального момента времени
% Задание вектора начальных условий
```

```

[Ug,q0] = UG_Dv_Obj_Psdo(t,h); q04=q0;
kv=[2 3 1];          % Вектор номеров осей поворота
% Начальные кватернионы поворота
KW01 = Ug2quat(Ug,kv); KW02 = KW01; KW03 = KW02; KW04 = KW02;
% Циклы интегрирования
disp('Подождите (идут вычисления)')
% Одношаговый алгоритм
H=h;          % Шаг интегрирования
t=0;  imas = 1;  tt1(imas)=t;
while (imas-1)*H<tfinal
    t=(imas-1)*H;
    fi1= OPR1h(h,t);      qw1 = fi2qw(fi1); KW1 = mult_Kwat(KW01,qw1); KW01 = KW1;
    Yg1=Quat2Ug(KW1,kv);  [z,Psd]=UG_Dv_Obj_Psdo(t+H,h);
    dpsi1(imas)=Yg1(1)-z(1); dteta1(imas)=Yg1(2)-z(2);
    dfi1(imas)=Yg1(3)-z(3); tt1(imas)=t;  imas=imas+1;
end
% Двухшаговый алгоритм
H=2*h;          % Шаг интегрирования
t=0;  imas = 1;  tt2(imas)=t;
while (imas-1)*H<tfinal
    t=(imas-1)*H;
    fi2= OPR2h(h,t);
    qw2 = fi2qw(fi2); KW2 = mult_Kwat(KW02,qw2); KW02 = KW2;
    Yg2=Quat2Ug(KW2,kv);  [z,Psd]=UG_Dv_Obj_Psdo(t+H,h);
    dpsi2(imas)=Yg2(1)-z(1); dteta2(imas)=Yg2(2)-z(2);
    dfi2(imas)=Yg2(3)-z(3); tt2(imas)=t;  imas=imas+1;
end
% Трехшаговый алгоритм
H=3*h;          % Шаг интегрирования
t=0;  imas = 1;  tt3(imas)=t;
while (imas-1)*H<tfinal
    t=(imas-1)*H;
    fi3= OPR3h(h,t);
    qw3 = fi2qw(fi3); KW3 = mult_Kwat(KW03,qw3); KW03 = KW3;
    Yg3=Quat2Ug(KW3,kv);  [z,Psd]=UG_Dv_Obj_Psdo(t+H,h);
    dpsi3(imas)=Yg3(1)-z(1); dteta3(imas)=Yg3(2)-z(2);
    dfi3(imas)=Yg3(3)-z(3); tt3(imas)=t;  imas=imas+1;
end
% Четырехшаговый алгоритм
H=4*h;          % Шаг интегрирования

```

```

t=0;  imas = 1;      tt4(imas)=t;
while (imas-1)*H<tfinal
    t=(imas-1)*H;
    fi4= OPR4h(h,t);
    qw4 = fi2qw(fi4); KW4 = mult_Kwat(KW04,qw4);  KW04 = KW4;
    Yg4=Quat2Ug(KW4,kv);    [z,Psd]=UG_Dv_Obj_Psdo(t+H,h);
    dps4(imas)=Yg4(1)-z(1);  dteta4(imas)=Yg4(2)-z(2);
    dfi4(imas)=Yg4(3)-z(3);  tt4(imas)=t;  imas=imas+1;
end

% ПОСТРОЕНИЕ ГРАФИКОВ
stit1='Погрешности алгоритмов БИСО (радианы). Метод Пикара ';
sy1ab1='OPR1h';  sx1ab1=' ';
sy1ab2='OPR2h';  sx1ab2=' ';  stit2=' ';
sy1ab3='OPR3h';  sx1ab3=' ';  stit3=' ';
sy1ab4='OPR4h';  sx1ab4='Время, с';  stit4=' ';
grafik12_4(tt1,dpsi1,dfi1,dteta1,stit1,sy1ab1,sx1ab1,...
    tt2,dpsi2,dfi2,dteta2,stit2,sy1ab2,sx1ab2,...
    tt3,dpsi3,dfi3,dteta3,stit3,sy1ab3,sx1ab3, ...
    tt4,dpsi4,dfi4,dteta4,stit4,sy1ab4,sx1ab4);
[Const,Dr1]=Drift_L(tt1,dpsi1); [Const,Dr2]=Drift_L(tt2,dpsi2);
[Const,Dr3]=Drift_L(tt3,dpsi3); [Const,Dr4]=Drift_L(tt4,dpsi4);
Dr=[Dr1,Dr2,Dr3,Dr4];
GrTxtVEulBr14

```

Далі наведені тексти процедур, що реалізують обчислення у відповідності до формул (5.42), (5.43), (5.44) та (5.46) синтезованих алгоритмів.

```

function FI = OPR1h(h,t)
% Одношаговый безразгонный алгоритм вычисления методом Пикара
% приращения вектора Эйлера по измерениям квазиординат
%   t - момент времени начала шага интегрирования
%   h - шаг опроса измерителей
%   FI - приращение вектора Эйлера за время h
%   q1 - приращение квазиординат на интервале от t до t+h
% Лазарев Ю. Ф.      29-08-2016
z1=UgDvObPseudo(t+h,h);      q1=[z1(4) z1(5) z1(6)];
FI=q1;
=====
function FI = OPR2h(h,t)

```

```
% Двухшаговый безразгонный алгоритм метода Пикара
% вычисления вектора Эйлера по измерению квазиординат
% t - момент времени начала шага интегрирования
% h - шаг опроса измерителей
% FI - приращение вектора Эйлера за время h
% q1 - приращение квазиординат на интервале от t до t+h
% q2 - приращение квазиординат на интервале от t+h до t+2h
% Лазарев Ю. Ф. 29-08-2016
z1=UgDvObPseudo(t+h,h); q1=[z1(4) z1(5) z1(6)];
z2=UgDvObPseudo(t+2*h,h); q2=[z2(4) z2(5) z2(6)];
q12=q1+q2; q1Xq2=cross(q1,q2);
FI=q12+2*q1Xq2/3;
```

=====

```
function FI = OPR3h(h,t)
% Трехшаговый безразгонный алгоритм метода Пикара
% вычисления вектора Эйлера по измерению квазиординат
% t - момент времени начала шага интегрирования
% h - шаг опроса измерителей
% FI - приращение вектора Эйлера за время h
% q1 - приращение квазиординат на интервале от t до t+h
% q2 - приращение квазиординат на интервале от t+h до t+2h
% q3 - приращение квазиординат на интервале от t+2h до t+3h
% Лазарев Ю. Ф. 29-08-2016
z1=UgDvObPseudo(t+h,h); q1=[z1(4) z1(5) z1(6)];
z2=UgDvObPseudo(t+2*h,h); q2=[z2(4) z2(5) z2(6)];
z3=UgDvObPseudo(t+3*h,h); q3=[z3(4) z3(5) z3(6)];
q123=q1+q2+q3;
q1Xq2=cross(q1,q2); q1Xq3=cross(q1,q3); q2Xq3=cross(q2,q3);
FI=q123+(57*(q1Xq2+q2Xq3)+33*q1Xq3)/80;
```

=====

```
function FI = OPR4h(h,t)
% Четырехшаговый безразгонный алгоритм метода Пикара
% вычисления вектора Эйлера по измерению квазиординат
% t - момент времени начала шага интегрирования
% h - шаг опроса измерителей
% FI - приращение вектора Эйлера за время h
% q1 - приращение квазиординат на интервале от t до t+h
% q2 - приращение квазиординат на интервале от t+h до t+2h
% q3 - приращение квазиординат на интервале от t+2h до t+3h
% q4 - приращение квазиординат на интервале от t+3h до t+4h
```



```

z1=UgDvObPseudo(t+h,h);      q1=[z1(4) z1(5) z1(6)];
z2=UgDvObPseudo(t+2*h,h);      q2=[z2(4) z2(5) z2(6)];
z3=UgDvObPseudo(t+3*h,h);      q3=[z3(4) z3(5) z3(6)];
z4=UgDvObPseudo(t+4*h,h);      q4=[z4(4) z4(5) z4(6)];
q1234=q1+q2+q3+q4;
q1Xq2=cross(q1,q2); q1Xq3=cross(q1,q3); q1Xq4=cross(q1,q4);
q2Xq3=cross(q2,q3); q2Xq4=cross(q2,q4); q3Xq4=cross(q3,q4);
Fl=q1234+(736*(q1Xq2+q3Xq4)+334*(q1Xq3+q2Xq4)+526*q1Xq4+3*218*q2Xq3)/945;

```

Результати дослідження залежностей відносних дрейфів від величині частотного параметра зведені у тексті програми **Gr_1234h**.

```

% Gr_1234h
clear all, clc
om=1; am=0.1; k=am^2*om;
h=[1 5e-1 2e-1 1e-1 5e-2 2e-2 1e-2 5e-3 2e-3 1e-3]; mu=om*h;
O1h=[-5.87e-4 -2.05e-4 -3.31e-5 -8.3e-6 -2.08e-6 -3.32e-5...
      -8.3e-8 -2.08e-8 -3.32e-9 -8.3e-10];
O2h=[-1.45e-4 -9.82e-6 -2.57e-7 -1.61e-8 -1.01e-9 -2.58e-11...
      -1.61e-12 -1.01e-13 -2.58e-15 -1.62e-16];
O3h=[5.27e-5 6.57e-6 1.99e-7 1.27e-8 5.99e-10 2.05e-11...
      1.28e-12 8e-14 2.05e-15 1.29e-16];
O4h=[3.49e-5 3.11e-7 5.12e-9 3.58e-10 2.31e-11 5.98e-13...
      3.74e-14 2.34e-15 5.93e-17 3.16e-18];
mu1=[0.001 1];
O1=8.3e-2*mu1.^2; O2=1.62e-2*mu1.^4; O3=1.29e-2*mu1.^4;
O4=3.16e-4*mu1.^4;
loglog(mu,abs(O1h)/k,'v',mu1,O1,'.',mu,abs(O2h)/k,'o',mu1,O2,'.',...
      mu,abs(O3h)/k,'s',mu1,O3,'.',mu,abs(O4h)/k,'^',...
      mu1,O4,'.', 'LineWidth',2),grid
set(gca,'FontSize',14)
title('Безразгонные алгоритмы. Вектор Эйлера. Квазикоординаты')
xlabel(' \mu = \omegah '), ylabel(' \delta ')
legend('O1h','\delta=0,083\mu^2','O2h','\delta=0,0162\mu^4',...
      'O3h','\delta=0,0129\mu^4','O4h','\delta=0,000316\mu^4',4)
gttext('Gr-1234h на основе программы OPR4h-upr','FontSize',12)

```

Відповідні графіки подані на рисунку 5.1. Там же пунктиром показані графіки апроксимувальних степеневих залежностей.

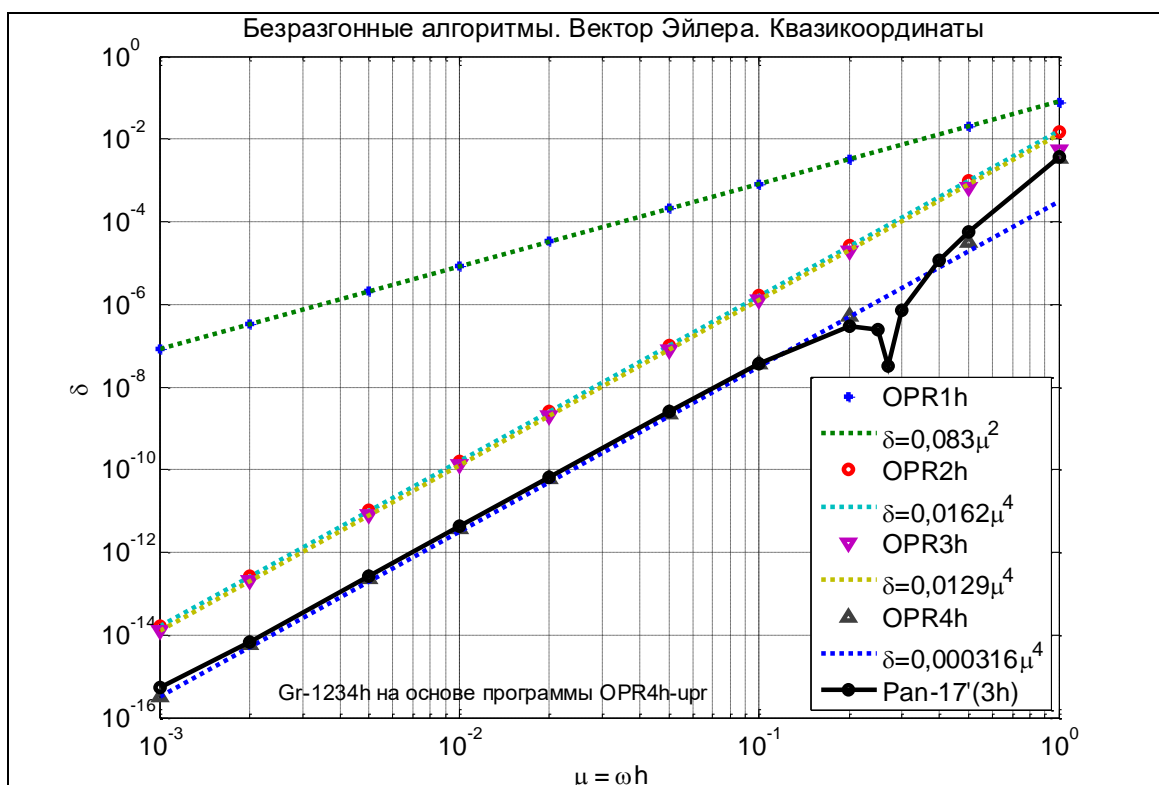


Рис. 5.1. Залежності відносних максимальних дрейфів від частотного параметра для алгоритмів методу Пікара векторного рівняння Ейлера з використанням вимірювань приростів квазікоординат

З вивчення результатів випливає такий висновок.

Досліджені залежності є близькими до простої степеневій функції вигляду $\delta = k \cdot \mu^N$, де ціле число N є порядком точності відповідного алгоритму. Параметри точності алгоритмів наведені у таблиці 5.1.

Таблиця 5.1. Параметри точності алгоритмів. вимірювання квазікоординат

Алгоритм	ζ	N	k
O1h	1	2	0,083
O2h	2	4	0,0162
O3h	3	4	0,0129
O4h	4	4	0,000316

Для порівняння додатково (у вигляді суцільної лінії) наведена точнісна характеристика трикрокового алгоритму за формулою (3.3.17') з книги А. П. Панова [5, с. 151]:

$$\varphi(3h) = \mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3 + \frac{27}{40} \mathbf{q}_1 \times \mathbf{q}_2 + \frac{9}{20} \mathbf{q}_1 \times \mathbf{q}_3 + \frac{27}{40} \mathbf{q}_2 \times \mathbf{q}_3. \quad (5.47)$$

Цей алгоритм реалізований процедурою **Pan_17_1**:

```
function fi = Pan_17_1(h,t)
% Безразгонный трехшаговый алгоритмы по формуле (3.3.17') Панова
% h- шаг опроса
% Лазарев Ю. Ф.      03-01-2015  05-08-2016
[Ug,q1] = UG_Dv_Obj_Psdo(t+h,h);
[Ug,q2] = UG_Dv_Obj_Psdo(t+2*h,h);
[Ug,q3] = UG_Dv_Obj_Psdo(t+3*h,h);
q123=q1+q2+q3;      q13=q1-q3;
fi = q123 + 9/20*cross(q1,q3) + 27/40*cross(q13,q2);
```

Неважко впевнитися, що цей трикроковий алгоритм значно (більш, ніж на порядок) точніше трикрокового алгоритму OPR3h та є близьким за точністю до чотирікрокового алгоритму OPR4h.

5.2.2. Випадок вимірювання вектора кутової швидкості

Як і раніше, вважаючи у рівнянні (5.32) величину $\varphi(\tau)$ в нульовому початковому наближенні рівною нулю і використовуючи загальне подання (5.12) вектора кутової швидкості для чотирікрокового алгоритму, отримаємо рівняння першого наближення:

$$\frac{d\varphi_{(1)}}{d\tau} \approx \boldsymbol{\omega} = \boldsymbol{\omega}_0 + \mathbf{E}\zeta + \mathbf{F}\zeta^2 + \mathbf{G}\zeta^3 + \mathbf{H}\zeta^4, \quad (5.48)$$

розв'язок якого на інтервалі кроку інтегрування має вигляд

$$\Phi_{(1)} = \int_0^{\tau} \omega(\tau) d\tau = h \int_0^{\frac{\tau}{h}} \omega(\zeta) d\zeta = h \left(\omega_0 \zeta + \mathbf{E} \frac{\zeta^2}{2} + \mathbf{F} \frac{\zeta^3}{3} + \mathbf{G} \frac{\zeta^4}{4} + \mathbf{H} \frac{\zeta^5}{5} \right); \quad \left(\zeta = \frac{\tau}{h} \right). \quad (5.49)$$

Тепер цей розв'язок можна використати для побудови другого наближення рівняння (5.32), розв'язок матиме вигляд (5.34). З врахуванням виразу (5.49), знайдемо

$$\Delta\Phi_{(2)} \approx \frac{h^2}{2} \int_0^{\frac{\tau}{h}} \left(\left(\omega_0 \zeta + \mathbf{E} \frac{\zeta^2}{2} + \mathbf{F} \frac{\zeta^3}{3} + \mathbf{G} \frac{\zeta^4}{4} + \mathbf{H} \frac{\zeta^5}{5} \right) \times \left(\omega_0 + \mathbf{E} \zeta + \mathbf{F} \zeta^2 + \mathbf{G} \zeta^3 + \mathbf{H} \zeta^4 \right) \right) d\zeta. \quad (5.50)$$

Внаслідок властивостей векторного добутку підінтегральний вираз у (5.50) приводиться до такого вигляду

$$\begin{aligned} & \left(\left(\omega_0 \zeta + \mathbf{E} \frac{\zeta^2}{2} + \mathbf{F} \frac{\zeta^3}{3} + \mathbf{G} \frac{\zeta^4}{4} + \mathbf{H} \frac{\zeta^5}{5} \right) \times \left(\omega_0 + \mathbf{E} \zeta + \mathbf{F} \zeta^2 + \mathbf{G} \zeta^3 + \mathbf{H} \zeta^4 \right) \right) = \\ & = (\omega_0 \times \mathbf{E}) \left(1 - \frac{1}{2}\right) \zeta^2 + (\omega_0 \times \mathbf{F}) \left(1 - \frac{1}{3}\right) \zeta^3 + (\omega_0 \times \mathbf{G}) \left(1 - \frac{1}{4}\right) \zeta^4 + (\omega_0 \times \mathbf{H}) \left(1 - \frac{1}{5}\right) \zeta^5 + \\ & + (\mathbf{E} \times \mathbf{F}) \left(\frac{1}{2} - \frac{1}{3}\right) \zeta^4 + (\mathbf{E} \times \mathbf{G}) \left(\frac{1}{2} - \frac{1}{4}\right) \zeta^5 + (\mathbf{E} \times \mathbf{H}) \left(\frac{1}{2} - \frac{1}{5}\right) \zeta^6 + (\mathbf{F} \times \mathbf{G}) \left(\frac{1}{3} - \frac{1}{4}\right) \zeta^6 + \\ & + (\mathbf{F} \times \mathbf{H}) \left(\frac{1}{3} - \frac{1}{5}\right) \zeta^7 + (\mathbf{G} \times \mathbf{H}) \left(\frac{1}{4} - \frac{1}{5}\right) \zeta^8 = \\ & = (\omega_0 \times \mathbf{E}) \frac{1}{2} \zeta^2 + (\omega_0 \times \mathbf{F}) \frac{2}{3} \zeta^3 + (\omega_0 \times \mathbf{G}) \frac{3}{4} \zeta^4 + (\omega_0 \times \mathbf{H}) \frac{4}{5} \zeta^5 + (\mathbf{E} \times \mathbf{F}) \frac{1}{6} \zeta^4 + \\ & + (\mathbf{E} \times \mathbf{G}) \frac{1}{4} \zeta^5 + (\mathbf{E} \times \mathbf{H}) \frac{3}{10} \zeta^6 + (\mathbf{F} \times \mathbf{G}) \frac{1}{12} \zeta^6 + (\mathbf{F} \times \mathbf{H}) \frac{2}{15} \zeta^7 + (\mathbf{G} \times \mathbf{H}) \frac{1}{20} \zeta^8 \end{aligned}$$

Отже, здійснивши інтегрування, вираз (5.50) можна записати у такий спосіб:

$$\Delta\Phi_{(2)} \approx \frac{h^2}{2} \left\{ \begin{aligned} & (\omega_0 \times \mathbf{E}) \frac{1}{6} \zeta^3 + (\omega_0 \times \mathbf{F}) \frac{1}{6} \zeta^4 + (\omega_0 \times \mathbf{G}) \frac{3}{20} \zeta^5 + (\omega_0 \times \mathbf{H}) \frac{2}{15} \zeta^6 + (\mathbf{E} \times \mathbf{F}) \frac{1}{30} \zeta^5 + \\ & + (\mathbf{E} \times \mathbf{G}) \frac{1}{24} \zeta^6 + (\mathbf{E} \times \mathbf{H}) \frac{3}{70} \zeta^7 + (\mathbf{F} \times \mathbf{G}) \frac{1}{84} \zeta^7 + (\mathbf{F} \times \mathbf{H}) \frac{1}{60} \zeta^8 + (\mathbf{G} \times \mathbf{H}) \frac{1}{180} \zeta^9 \end{aligned} \right\}. \quad (5.51)$$

Запишемо праву частину рівності (5.51) в більш зручній для користування формі:

$$\Delta\Phi_{(2)} \approx \frac{h^2}{2} \zeta^3 \{ \mathbf{K} + \mathbf{M} + \mathbf{N} + \mathbf{U} \}, \quad (5.52)$$

де позначено

$$\begin{aligned}
\mathbf{K} &= \boldsymbol{\omega}_0 \times \left(\frac{1}{6} \mathbf{E} + \frac{\zeta}{6} \mathbf{F} + \frac{3\zeta^2}{20} \mathbf{G} + \frac{2\zeta^3}{15} \mathbf{H} \right); & \mathbf{M} &= \mathbf{E} \times \zeta^2 \left(\frac{1}{30} \mathbf{F} + \frac{\zeta}{24} \mathbf{G} + \frac{3\zeta^2}{70} \mathbf{H} \right); \\
\mathbf{N} &= \mathbf{F} \times \zeta^4 \left(\frac{1}{84} \mathbf{G} + \frac{\zeta}{60} \mathbf{H} \right); & \mathbf{U} &= \mathbf{G} \times \frac{\zeta^6}{180} \mathbf{H}.
\end{aligned} \tag{5.53}$$

Рівняння третього наближення (5.39) приводить до суми подвійних векторних добутків векторів. Тому ним слід в нашій постановці задачі знехтувати: $\Delta\boldsymbol{\varphi}_{(3)} = 0$. В результаті отримаємо остаточно

$$\boldsymbol{\varphi}(\zeta h) = h \left\{ \boldsymbol{\omega}_0 \zeta + \mathbf{E} \frac{\zeta^2}{2} + \mathbf{F} \frac{\zeta^3}{3} + \mathbf{G} \frac{\zeta^4}{4} + \mathbf{H} \frac{\zeta^5}{5} \right\} + \frac{h^2}{2} \zeta^3 \{ \mathbf{K} + \mathbf{M} + \mathbf{N} + \mathbf{U} \}. \tag{5.54}$$

Такою є загальна формула для усіх багатокрокових алгоритмів аж до чотирьокрових.

Однокроковий алгоритм

У випадку однокрокового алгоритму $\zeta = 1$, $\mathbf{E} = \boldsymbol{\omega}_1 - \boldsymbol{\omega}_0$, $\mathbf{F} = \mathbf{G} = \mathbf{H} = 0$. Тому з (5.54) випливає

$$\boldsymbol{\varphi}(h) = h \left(\boldsymbol{\omega}_0 + \mathbf{E} \frac{1}{2} + h(\boldsymbol{\omega}_0 \times \mathbf{E}) \frac{1}{12} \right) = h \left(\boldsymbol{\omega}_0 + \frac{1}{2} (\boldsymbol{\omega}_1 - \boldsymbol{\omega}_0) + \frac{h}{12} (\boldsymbol{\omega}_0 \times) (\boldsymbol{\omega}_1 - \boldsymbol{\omega}_0) \right).$$

Або, остаточно

$$\boldsymbol{\varphi}(h) = h \frac{\boldsymbol{\omega}_0 + \boldsymbol{\omega}_1}{2} + \frac{h^2}{12} (\boldsymbol{\omega}_0 \times) \boldsymbol{\omega}_1. \tag{5.55}$$

Двокроковий алгоритм

При синтезі двокрокового алгоритму слід використати такі значення параметрів:

$$\zeta = 2, \quad \mathbf{E} = \frac{1}{2} (-3\boldsymbol{\omega}_0 + 4\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2), \quad \mathbf{F} = \frac{1}{2} (\boldsymbol{\omega}_0 - 2\boldsymbol{\omega}_1 + \boldsymbol{\omega}_2), \quad \mathbf{G} = \mathbf{H} = 0.$$

Загальна формула (5.54) перетворюється на таку

$$\Phi(2h) = h \left\{ 2\omega_0 + 2\mathbf{E} + \frac{8}{3}\mathbf{F} \right\} + \frac{h^2}{2} \{ \mathbf{K}(2h) + \mathbf{M}(2h) \},$$

де

$$\mathbf{K}(2h) = \omega_0 \times \left(\frac{1}{6}\mathbf{E} + \frac{1}{3}\mathbf{F} \right); \quad \mathbf{M}(2h) = \mathbf{E} \times \left(\frac{2}{15}\mathbf{F} \right). \quad (5.56)$$

Задля обчислення коефіцієнтів використаємо програму

```
% VSP2nn          % Ю.Ф. Лазарев   15-03-2018
clear all,        clc
disp('Коэффициенты при векторных произведениях ')
disp('    для двухшагового алгоритма')
disp('    1) исходные данные')
dz=2, om0=[1 0 0]
E=[-3 4 -1]/2, F=[1 -2 1]/2
disp('    2) алгоритм в первом приближении')
fi2hdh = om0*dz + E*dz^2/2 + F*dz^3/3
disp('    3) к-ты при векторных произведениях')
dK=(E+F*dz)/6, dM=dz^2/840*28*F
K=v1Xv2(om0,dK), M=v1Xv2(E,dM)
dFi2hdh2=(K+M)*dz^3/2
```

У цій програмі і надалі використовується наступна процедура-функція **v1Xv2**:

```
function Vout=v1Xv2(v1,v2)
% Процедура v1Xv2
% 1) образует квадратную матрицу на основе произведения двух заданных
%    векторов
% 2) создает соответствующую кососимметричную матрицу
% 3) строит вектор из элементов КСМ-матрицы выше главной диагонали
% Ю. Ф. Лазарев   13-03-2018
Mat=v1'*v2; [n,k]=size(Mat);
ksm=Mat-Mat'; m=0;
for i=1:n-1
    for j=i+1:k
        m=m+1;
```

```

Vout(m)=ksm(i,j);
end
end

```

У результаті роботи програми **VSP2nn** отримаємо

*Коэффициенты при векторных произведениях
для двухшагового алгоритма*

1) исходные данные

$$\begin{aligned}
 dz &= 2 \\
 om0 &= 1 \quad 0 \quad 0 \\
 E &= -3/2 \quad 2 \quad -1/2 \\
 F &= 1/2 \quad -1 \quad 1/2
 \end{aligned}$$

2) алгоритм в первом приближении

$$fi2hdh = 1/3 \quad 4/3 \quad 1/3$$

3) к-ты при векторных произведениях

$$\begin{aligned}
 dK &= -1/12 \quad 0 \quad 1/12 \\
 dM &= 1/15 \quad -2/15 \quad 1/15 \\
 dM &= 1/15 \quad -2/15 \quad 1/15 \\
 K &= 0 \quad 1/12 \quad 0 \\
 M &= 1/15 \quad -1/15 \quad 1/15 \\
 dFi2hdh2 &= 4/15 \quad 1/15 \quad 4/15
 \end{aligned}$$

Це дозволяє записати формулу двокрокового алгоритму у вигляді

$$\boxed{\varphi(2h) = \frac{h}{3}(\omega_0 + 4\omega_1 + \omega_2) + \frac{h^2}{15}\{4\omega_0 \times \omega_1 + \omega_0 \times \omega_2 + 4\omega_1 \times \omega_2\}}. \quad (5.57)$$

Ця формула збігається з формулою алгоритму (3.3.126) [5, с. 179], якщо останню скорегувати, змінивши знак на протилежний перед останньою складовою.

Трикроковий алгоритм

В цьому випадку маємо

$$\begin{aligned}\zeta = 3; \quad \mathbf{E} &= \frac{1}{6}(-11\omega_0 + 18\omega_1 - 9\omega_2 + 2\omega_3); \quad \mathbf{F} = \frac{1}{2}(2\omega_0 - 5\omega_1 + 4\omega_2 - \omega_3); \\ \mathbf{G} &= \frac{1}{6}(-\omega_0 + 3\omega_1 - 3\omega_2 + \omega_3); \quad \mathbf{H} = 0.\end{aligned}\quad (5.58)$$

Програма **VSP3nn** обчислює коефіцієнти при векторах кутової швидкості та їх добутків:

```
% VSP3nn          % Ю.Ф. Лазарев   15-03-2018
clear all, clc
disp('Коеффициенты при векторных произведениях ')
disp('   для трехшагового алгоритма')
disp('   1) исходные данные')
dz=3, om0=[1 0 0 0]
E=[-11 18 -9 2]/6, F=[2 -5 4 -1]/2, G=[-1 3 -3 1]/6
disp('   2) алгоритм в первом приближении')
fi3hhdh=dz*(om0 + dz*E/2 + dz^2*F/3 + dz^3*G/4)
disp('   3) к-ты при векторных произведениях')
dK=(10*E+10*F*dz+9*G*dz^2)/60, dM=dz^2/840*(28*F+35*dz*G),
A=dz^4/420*5, dN=A*G
K=v1Xv2(om0,dK), M=v1Xv2(E,dM), N=v1Xv2(F,dN)
dFi3hhdh2=(K+M+N)*dz^3/2
```

В результаті виходить

Коеффициенты при векторных произведениях

для трехшагового алгоритма

1) исходные данные

$$\begin{aligned}dz &= 3 \\ om0 &= \begin{matrix} 1 & 0 & 0 & 0 \end{matrix} \\ E &= \begin{matrix} -11/6 & 3 & -3/2 & 1/3 \end{matrix} \\ F &= \begin{matrix} 1 & -5/2 & 2 & -1/2 \end{matrix}\end{aligned}$$

$$G = \begin{matrix} & -1/6 & 1/2 & -1/2 & 1/6 \end{matrix}$$

2) алгоритм в первом приближении

$$fi3hdh = \begin{matrix} & 3/8 & 9/8 & 9/8 & 3/8 \end{matrix}$$

3) к-ты при векторных произведениях

$$dK = \begin{matrix} & -11/360 & -3/40 & 3/40 & 11/360 \end{matrix}$$

$$dM = \begin{matrix} & 9/80 & -3/16 & 3/80 & 3/80 \end{matrix}$$

$$dN = \begin{matrix} & -9/56 & 27/56 & -27/56 & 9/56 \end{matrix}$$

$$K = \begin{matrix} & -3/40 & 3/40 & 11/360 & 0 & 0 & 0 \end{matrix}$$

$$M = \begin{matrix} & 1/160 & 1/10 & -17/160 & -27/160 & 7/40 & -11/160 \end{matrix}$$

$$N = \begin{matrix} & 9/112 & -9/56 & 9/112 & 27/112 & -9/56 & 9/112 \end{matrix}$$

$$dFi3hdh2 =$$

$$\begin{matrix} 351/2240 & 27/140 & 141/2240 & 784/803 & 27/140 & 351/2240 \end{matrix}$$

У підсумку формула трикрокового алгоритму набуває такої форми:

$$\Phi(3h) = \frac{3}{8}h(\omega_0 + 3\omega_1 + 3\omega_2 + \omega_3) + h^2 \left[\begin{aligned} & \frac{351}{2240}(\omega_0 \times)\omega_1 + \frac{27}{140}(\omega_0 \times)\omega_2 + \\ & \frac{141}{2240}(\omega_0 \times)\omega_3 + \frac{784}{803}(\omega_1 \times)\omega_2 + \\ & \frac{27}{140}(\omega_1 \times)\omega_3 + \frac{351}{2240}(\omega_2 \times)\omega_3 \end{aligned} \right]. \quad (5.57)$$

Чотирикроковий алгоритм

Для синтезу чотирикрокового алгоритму потрібно врахувати наступні значення параметрів:

$$\zeta = 4; \quad \mathbf{E} = \frac{1}{12}(-25\omega_0 + 48\omega_1 - 36\omega_2 + 16\omega_3 - 3\omega_4);$$

$$\mathbf{F} = \frac{1}{24}(35\omega_0 - 104\omega_1 + 114\omega_2 - 56\omega_3 + 11\omega_4);$$

$$\mathbf{G} = \frac{1}{12}(-5\omega_0 + 18\omega_1 - 24\omega_2 + 14\omega_3 - 3\omega_4);$$

$$\mathbf{H} = \frac{1}{24}(\omega_0 - 4\omega_1 + 6\omega_2 - 4\omega_3 + \omega_4). \quad (5.59)$$

Користуючись аналогічною програмою **VSP4nn**, знайдемо значення коефіцієнтів за формою (5.53).

```
% VSP4nn          % Ю.Ф. Лазарев   15-03-2018
clear all,        clc
disp('Кoeffициенты при векторных произведениях ')
disp('   для четырехшагового алгоритма')
disp('   1) исходные данные')
dz=4;  om0=[1 0 0 0 0];
E=[-25 48 -36 16 -3]/12;  F=[35 -104 114 -56 11]/24;
G=[-5 18 -24 14 -3]/12;  H=[1 -4 6 -4 1]/24;
disp('   2) первое приближение')
fi4hdh= dz*om0 + dz^2/2*E + dz^3/3*F + dz^4/4*G + dz^5/5*H
disp('   3) вычисление коэффициентов')
dK=(10*E+10*dz*F+9*dz^2*G+8*dz^3*H)/60
dM=dz^2/840*(28*F+35*dz*G+36*dz^2*H)
dN=dz^4*(G/84+dz*H/60); dU=dz^6/180*H
K=v1Xv2(om0,dK), M=v1Xv2(E,dM), N=v1Xv2(F,dN), U=v1Xv2(G,dU)
disp('   4) второе приближение')
fi4hdh=(K+M+N+U)*dz^3/2
```

Результатом виконання дій цією програмою є такі значення матриць коефіцієнтів

Кoeffициенты при векторных произведениях

для четырехшагового алгоритма

1) исходные данные

2) первое приближение

$fi4hdh = \begin{matrix} 14/45 & 64/45 & 8/15 & 64/45 & 14/45 \end{matrix}$

3) вычисление коэффициентов

$dK = \begin{matrix} -7/360 & -2/45 & 0 & 2/45 & 7/360 \end{matrix}$

$dM = \begin{matrix} 13/105 & -44/315 & -2/35 & 4/105 & 11/315 \end{matrix}$

$$\begin{aligned}
dU &= \begin{matrix} 128/135 & -512/135 & 256/45 & -512/135 & 128/135 \\ K = & -2/45 & 0 & 2/45 & 7/360 & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ M = & -193/945 & 103/210 & -11/45 & -79/1890 & -68/105 \\ & 64/189 & 11/105 & -4/105 & -5/42 & 53/945 \\ N = & 92/945 & -4/315 & -4/15 & 172/945 & -88/315 \\ & 128/135 & -4/7 & -8/9 & 188/315 & -28/135 \\ U = & 64/405 & -64/135 & 64/135 & -64/405 & 128/135 \\ & -512/405 & 64/135 & 128/135 & -64/135 & 64/405 \end{matrix}
\end{aligned}$$

4) второе приближение

$$fi4h2dh2 =$$

$$\begin{aligned}
& \begin{matrix} 608/2835 & 16/135 & 32/135 & 148/2835 & 128/189 \\ 2048/2835 & 32/135 & 128/189 & 16/135 & 608/2835 \end{matrix}
\end{aligned}$$

Це означає, що формула чотирикрокового алгоритму матиме такий вигляд

$$\begin{aligned}
\varphi(4h) &= \frac{2}{45} h(7\omega_0 + 32\omega_1 + 12\omega_2 + 32\omega_3 + 7\omega_4) + \\
&+ h^2 \left(\frac{608}{2835} \omega_0 \times \omega_1 + \frac{16}{135} \omega_0 \times \omega_2 + \frac{32}{135} \omega_0 \times \omega_3 + \frac{148}{2835} \omega_0 \times \omega_4 + \frac{128}{189} \omega_1 \times \omega_2 + \right. \\
&\left. + \frac{2048}{2835} \omega_1 \times \omega_3 + \frac{32}{135} \omega_1 \times \omega_4 + \frac{128}{189} \omega_2 \times \omega_3 + \frac{16}{135} \omega_2 \times \omega_4 + \frac{608}{2835} \omega_3 \times \omega_4 \right) \cdot \quad (5.60)
\end{aligned}$$

Дослідження дрейфів алгоритмів

Комп'ютерне моделювання згідно отриманих формул (5.54), (5.56), (5.58) та (5.60) приводить до безрозмірних залежностей дрейфів від кроку опитування, які графічно подані на рисунку 5.2.

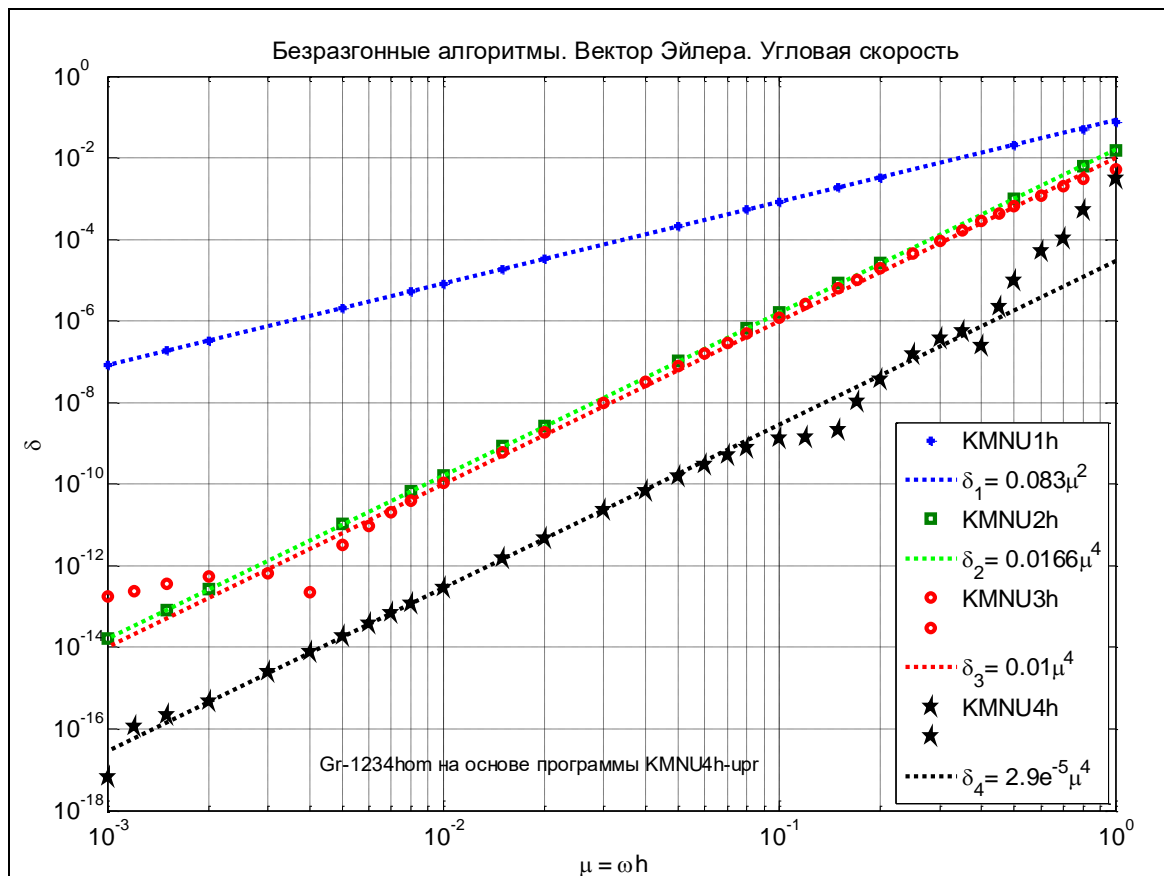


Рис. 5.2. Залежності відносних максимальних дрейфів від частотного параметра для алгоритмів методу Пікара векторного рівняння Ейлера з використанням вимірювання кутової швидкості

На основі отриманих графіків можна дійти висновків:

1) параметри точності створених і досліджених алгоритмів наведені у Таблиці 5.2:

Таблиця 5.2. Параметри точності алгоритмів при вимірюванні кутової швидкості

Алгоритм	Кроковість	N	k
KMNU1h	1	2	0,083
KMNU2h	2	4	0,0166
KMNU3h	3	4	0,01
KMNU4h	4	4	2,9e-5

2) показники степеня точності залишаються незмінними у всьому діапазоні частотного параметра лише для алгоритмів KMNU1h та KMNU2h; спостерігаються деякі незначні відхилення від вказаної залежності за великих і малих значення частотного параметра в алгоритмі KMNU3h;

3) чотирикроковий алгоритм KMNU4h має вказані параметри точності лише в діапазоні 0,001...0,1 частотного параметру; він набагато гірший за точністю алгоритмів KMNU2h та KMNU3h;

4) найбільш точним виявляється двокроковий алгоритм KMNU2h.

5.3. Алгоритми на ґрунті рівняння Пуассона

(На основі магістерської дисертації С. В. Льчука [9])

Для другого прикладу у якості рівняння орієнтації використаємо рівняння Пуассона.

У цьому випадку задача полягає в отриманні алгоритму чисельного розв'язку матричного диференціального рівняння:

$$\frac{dC}{dt} = C(\omega \times), \quad (5.61)$$

де C – матриця напрямних косинусів,

$$(\omega \times) = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \quad (5.62)$$

При цьому інформація про кутову швидкість надається у дискретні моменти часу $t_i (i = 1, 2, \dots)$ з інтервалами h кроку опитування вимірювачів (тобто $t_i - t_{i-1} = h$) або у вигляді миттєвих значень кутової швидкості $\omega_j(t_i)$, ($j = 1, 2, 3$), або у вигляді *приростів квазікоординат* $q_j(t_i)$ (при використанні інтегровальних гіротахometrів):

$$q_j(t_i) = \int_{t_{i-1}}^{t_i} \omega_j(\tau) d\tau.$$

Інтегруючи (5.61) в межах від t_0 до t , отримаємо

$$C(t) = C(t_0) + \int_{t_0}^t C(\xi) \cdot [\omega(\xi) \times] d\xi. \quad (5.63)$$

Розв'язок рівняння (5.63) шукатимемо у вигляді добутку початкового значення матриці напрямних косинусів на деяку невідому матрицю $P(t)$ – функцію часу, вираз для якої й потрібно знайти:

$$C(t) = C(t_0) \cdot P(t). \quad (5.64)$$

Підставимо (5.64) у (5.63):

$$C(t_0) \cdot P(t) = C(t_0) + \int_{t_0}^t C(t_0) \cdot P(\xi) \cdot [\omega(\xi) \times] d\xi,$$

звідки випливає

$$P(t) = E + \int_{t_0}^t P(\xi) \cdot [\omega(\xi) \times] d\xi, \quad (5.65)$$

де E – одинична матриця розміром (3×3) . Тепер рівняння (2.3) можна подати у вигляді (5.64).

Розглянемо алгоритм знаходження значення матриці напрямних косинусів $C_{n+1} = C(t_{n+1})$ у момент часу $t = t_{n+1} = t_n + h$ за заданим її значенням $C_n = C(t_n)$ у попередній момент часу $t = t_n$, де h – крок інтегрування.

Застосовуючи (5.64), знайдемо

$$C_{n+1} = C_n \cdot P_{n+1}, \quad (5.66)$$

причому, у відповідності до (5.55), P_{n+1} є розв'язком рівняння

$$P(t) = E + \int_{t_n}^t P(\xi) \cdot [\omega(\xi) \times] d\xi \quad (5.67)$$

у момент часу $t = t_{n+1} = t_n + h$.

Рівняння (5.77), за методом Пікара, розв'язується послідовними наближеннями.

У нульовому наближенні приймемо $P_{(0)}(t) = E$. Підставивши це у праву частину виразу (5.67), отримаємо розв'язок цього рівняння у першому наближенні:

$$P_{(1)}(t) = E + \int_{t_n}^t P_{(0)}(\xi) \cdot [\omega(\xi) \times] d\xi = E + \Delta P_{(1)}(t),$$

де позначено

$$\Delta P_{(1)}(t) = \int_{t_n}^t [\omega(\xi) \times] d\xi. \quad (5.68)$$

Для знаходження розв'язку у другому наближенні у відповідності до методу Пікара, достатньо підставити у праву частину виразу (5.67) результат попереднього (першого) наближення матриці P :

$$\begin{aligned} P_{(2)}(t) &= E + \int_{t_n}^t P_{(1)}(\xi) \cdot [\omega(\xi) \times] d\xi = E + \int_{t_n}^t \left(E + \int_{t_n}^{\xi} [\omega(\eta) \times] d\eta \right) \cdot [\omega(\xi) \times] d\xi = \\ &= P_{(1)}(t) + \int_{t_n}^t \Delta P_{(1)}(\xi) \cdot [\omega(\xi) \times] d\xi = P_{(1)}(t) + \Delta P_{(2)}(t), \end{aligned}$$

де

$$\Delta P_{(2)}(t) = \int_{t_n}^t \Delta P_{(1)}(\xi) \cdot [\omega(\xi) \times] d\xi. \quad (5.69)$$

Використовуючи цей результат, в аналогічний спосіб знайдемо третє наближення

$$P_{(3)}(t) = E + \int_{t_n}^t P_{(2)}(\xi) \cdot [\omega(\xi) \times] d\xi = P_{(2)}(t) + \int_{t_n}^t \Delta P_{(2)}(\xi) \cdot [\omega(\xi) \times] d\xi = P_{(2)}(t) + \Delta P_{(3)}(t),$$

де

$$\Delta P_{(3)}(t) = \int_{t_n}^t \Delta P_{(2)}(\xi) \cdot [\omega(\xi) \times] d\xi. \quad (5.70)$$

У четвертому наближенні отримаємо:

$$P_{(4)}(t) = P_{(3)}(t) + \Delta P_{(4)}(t),$$

де

$$\Delta P_{(4)}(t) = \int_{t_n}^t \Delta P_{(3)}(\xi) \cdot [\omega(\xi) \times] d\xi. \quad (5.71)$$

5.3.1. Випадок вимірювань кутової швидкості

Однокрокові алгоритми

При вимірюванні кутової швидкості на одному кроці опитування є відомими дві виміряні величини вектора кутової швидкості основи: ω_0 – на початку та ω_1 – на кінці цього кроку. Тому в цьому випадку можна апроксимувати вектор кутової швидкості лінійною функцією часу:

$$\omega(\tau) = \omega_0 + 2\varepsilon\tau.$$

Щоб знайти невідомий векторний коефіцієнт ε , підставимо у цей вираз значення часу кінця кроку:

$$\omega_1 = \omega_0 + 2\varepsilon h,$$

з чого випливає:

$$\varepsilon = \frac{1}{2h}(\omega_1 - \omega_0).$$

Тому:

$$\omega(\tau) = \omega_0 + 2\frac{1}{2h}(\omega_1 - \omega_0)\tau = \omega_0 + (\omega_1 - \omega_0)\zeta = \omega_0 + \mathbf{A}\zeta, \quad (5.72)$$

де використані позначення:

$$\boxed{\mathbf{A} = \omega_1 - \omega_0}; \quad \zeta = \frac{\tau}{h}. \quad (5.73)$$

Перше наближення

З співвідношення (5.68) випливає у першому наближенні:

$$\Delta P_{(1)}(\tau) = \int_0^\tau [\omega(\xi) \times] d\xi = \int_0^\zeta [(\omega_0 \times) + (\mathbf{A} \times)\zeta] d(\zeta h) = h \int_0^\zeta [(\omega_0 \times) + (\mathbf{A} \times)\zeta] d\zeta,$$

або, остаточно:

$$\Delta P_{(1)}(\tau) = h[(\omega_0 \times)\zeta + \frac{1}{2}(\mathbf{A} \times)\zeta^2].$$

Наприкінці кроку інтегрування (тобто при $\zeta = 1$) величина матриці P набуде значення

$$P_{(1)}(h) = E + h \left[(\boldsymbol{\omega}_0 \times) + \frac{1}{2} (\mathbf{A} \times) \right] = E + \frac{h}{2} [(\boldsymbol{\omega}_0 \times) + (\boldsymbol{\omega}_1 \times)]. \quad (5.74)$$

Друге наближення

Вираз (5.69) перетворюється у такий спосіб

$$\begin{aligned} \Delta P_{(2)}(\tau) &= \int_0^\tau \Delta P_{(1)}(\xi) \cdot [(\boldsymbol{\omega}(\xi) \times)] d\xi = h^2 \int_0^\zeta [(\boldsymbol{\omega}_0 \times) \zeta + \frac{1}{2} (\mathbf{A} \times) \zeta^2] [(\boldsymbol{\omega}_0 \times) + (\mathbf{A} \times) \zeta] d\zeta = \\ &= h^2 \left\{ \frac{1}{2} (\boldsymbol{\omega}_0 \times)^2 \zeta^2 + \frac{1}{3} \left[(\boldsymbol{\omega}_0 \times)(\mathbf{A} \times) + \frac{1}{2} (\mathbf{A} \times)(\boldsymbol{\omega}_0 \times) \right] \zeta^3 + \frac{1}{8} (\mathbf{A} \times)^2 \zeta^4 \right\}. \end{aligned}$$

Задля спрощення введемо позначення:

$$\boxed{\mathbf{K}_{22} = \frac{1}{2} (\boldsymbol{\omega}_0 \times)^2}; \quad \boxed{\mathbf{K}_{23} = \frac{1}{3} \left[(\boldsymbol{\omega}_0 \times)(\mathbf{A} \times) + \frac{1}{2} (\mathbf{A} \times)(\boldsymbol{\omega}_0 \times) \right]}; \quad \boxed{\mathbf{K}_{24} = \frac{1}{8} (\mathbf{A} \times)^2}.$$

Внаслідок чого отримаємо:

$$\Delta P_{(2)}(\tau) = h^2 [\mathbf{K}_{22} \zeta^2 + \mathbf{K}_{23} \zeta^3 + \mathbf{K}_{24} \zeta^4] = h^2 \sum_{i=2}^4 \mathbf{K}_{2i} \zeta^i.$$

Тепер формулу однокрокового алгоритму у другому наближенні можна подати так ($\zeta = 1$):

$$P_{(2)}(h) = E + \frac{h}{2} [(\boldsymbol{\omega}_0 \times) + (\boldsymbol{\omega}_1 \times)] + h^2 \sum_{i=2}^4 \mathbf{K}_{2i}. \quad (5.75)$$

Третє наближення

Доданок у третьому наближенні отримаємо зі співвідношення (5.70):

$$\begin{aligned} \Delta P_{(3)}(\tau) &= \int_0^\tau \Delta P_{(2)}(\xi) \cdot [(\boldsymbol{\omega}(\xi) \times)] d\xi = h^3 \int_0^\zeta \left[\sum_{i=2}^4 \mathbf{K}_{2i} \zeta^i \right] [(\boldsymbol{\omega}_0 \times) + (\mathbf{A} \times) \zeta] d\zeta = \\ &= h^3 \left[\frac{1}{3} \mathbf{K}_{22} (\boldsymbol{\omega}_0 \times) + \frac{1}{4} [\mathbf{K}_{22} (\mathbf{A} \times) + \mathbf{K}_{23} (\boldsymbol{\omega}_0 \times)] + \frac{1}{5} [\mathbf{K}_{23} (\mathbf{A} \times) + \mathbf{K}_{24} (\boldsymbol{\omega}_0 \times)] + \frac{1}{6} \mathbf{K}_{22} (\mathbf{A} \times) \right]. \end{aligned}$$

Задля спрощення запису використаємо позначення:

$$\begin{aligned} \boxed{\mathbf{K}_{33} = \frac{1}{3} \mathbf{K}_{22} (\boldsymbol{\omega}_0 \times)}; & \quad \boxed{\mathbf{K}_{34} = \frac{1}{4} [\mathbf{K}_{22} (\mathbf{A} \times) + \mathbf{K}_{23} (\boldsymbol{\omega}_0 \times)]}; \\ \boxed{\mathbf{K}_{35} = \frac{1}{4} [\mathbf{K}_{23} (\mathbf{A} \times) + \mathbf{K}_{24} (\boldsymbol{\omega}_0 \times)]}; & \quad \boxed{\mathbf{K}_{36} = \frac{1}{6} \mathbf{K}_{22} (\mathbf{A} \times)}. \end{aligned}$$

В результаті отримаємо:

$$\Delta P_{(2)}(\tau) = h^3 [\mathbf{K}_{33}\zeta^3 + \mathbf{K}_{34}\zeta^4 + \mathbf{K}_{35}\zeta^5 + \mathbf{K}_{36}\zeta^6] = h^3 \sum_{j=3}^6 \mathbf{K}_{3j}\zeta^j.$$

Тому формула однокрокового алгоритму у третьому наближенні ($\zeta=1$) набуває такого вигляду:

$$\Delta P_{(3)}(h) = E + \frac{h}{2} [(\boldsymbol{\omega}_0 \times) + (\boldsymbol{\omega}_1 \times)] + h^2 \sum_{i=2}^4 \mathbf{K}_{2i} + h^3 \sum_{j=3}^6 \mathbf{K}_{3j}. \quad (5.76)$$

Четверте наближення

Доданок четвертого наближення отримаємо з співвідношення (5.71):

$$\Delta P_{(4)}(\tau) = \int_0^\tau \Delta P_{(3)}(\xi) \cdot [(\boldsymbol{\omega}(\xi) \times)] d\xi = h^4 \int_0^\tau \left[\sum_{j=3}^6 \mathbf{K}_{3j}\zeta^j \right] [(\boldsymbol{\omega}_0 \times) + (\mathbf{A} \times)\zeta] d\zeta.$$

Здійснивши операції добутку і інтегрування, обчислимо коефіцієнти \mathbf{K}_{4k} при кожному ζ^k :

$\mathbf{K}_{44} = \frac{1}{4} \mathbf{K}_{33}(\boldsymbol{\omega}_0 \times);$	$\mathbf{K}_{45} = \frac{1}{5} [\mathbf{K}_{33}(\mathbf{A} \times) + \mathbf{K}_{34}(\boldsymbol{\omega}_0 \times)];$	$\mathbf{K}_{46} = \frac{1}{6} [\mathbf{K}_{34}(\mathbf{A} \times) + \mathbf{K}_{35}(\boldsymbol{\omega}_0 \times)];$
$\mathbf{K}_{47} = \frac{1}{7} [\mathbf{K}_{35}(\mathbf{A} \times) + \mathbf{K}_{36}(\boldsymbol{\omega}_0 \times)];$		$\mathbf{K}_{48} = \frac{1}{8} \mathbf{K}_{37}(\boldsymbol{\omega}_0 \times).$

В результаті можна записати:

$$\Delta P_{(4)}(\tau) = h^4 \sum_{k=4}^8 \mathbf{K}_{4k}\zeta^k.$$

Наприкінці кроку інтегрування ($\zeta=1$), додаючи цей доданок до правої частини формули (5.16), приходимо до формули однокрокового алгоритму у четвертому наближенні:

$$\Delta P_{(4)}(h) = E + \frac{h}{2} [(\boldsymbol{\omega}_0 \times) + (\boldsymbol{\omega}_1 \times)] + h^2 \sum_{i=2}^4 \mathbf{K}_{2i} + h^3 \sum_{j=3}^6 \mathbf{K}_{3j} + h^4 \sum_{k=4}^8 \mathbf{K}_{4k}. \quad (5.77)$$

Двокрокові алгоритми

На двох кроках опитування, що складають один крок інтегрування, відомими є три виміряні величини вектора кутової швидкості основи: $\boldsymbol{\omega}_0$ – на початку, $\boldsymbol{\omega}_1$, $\boldsymbol{\omega}_2$ – на кінцях першого и другого кроків відповідно. Тому у цьому випадку кутову швидкість потрібно розглядати у вигляді квадратного поліному:

$$\omega(\tau) = \omega_o + 2\varepsilon\tau + 3a\tau^2.$$

Щоб знайти невідомі векторні коефіцієнти ε , a , підставимо у цей вираз час кінця кожного кроку опитування ($\omega_o = \omega(0)$, $\omega_1 = \omega(h)$, $\omega_2 = \omega(2h)$).

Прирости кутової швидкості на кожному з двох кроків опитування дорівнюватимуть:

$$\omega_1 - \omega_o = 2\varepsilon h + 3ah^2; \quad \omega_2 - \omega_o = 4\varepsilon h + 12ah^2.$$

Розв'язавши цю систему рівнянь відносно εh і ah^2 та вводячи відповідні позначення, отримаємо:

$$\mathbf{A} = \frac{1}{4}(-3\omega_o + 4\omega_1 - \omega_2); \quad \mathbf{B} = \frac{1}{6}(\omega_o - 2\omega_1 + \omega_2). \quad (5.78)$$

При цьому вираз кутової швидкості можна подати у формі:

$$\omega(\tau) = \omega_o + 2\mathbf{A}\zeta + 3\mathbf{B}\zeta^2, \quad \left(\zeta = \frac{\tau}{h}\right). \quad (5.79)$$

Перше наближення

З співвідношення (5.68) з врахуванням (5.79), виходить у першому наближенні:

$$\Delta P_{(1)}(\tau) = \int_0^\tau [\omega(\xi) \times] d\xi = \int_0^\zeta [(\omega_o \times) + 2(\mathbf{A} \times)\zeta + 3(\mathbf{B} \times)\zeta^2] d(\zeta h),$$

або

$$\Delta P_{(1)}(\tau) = h[(\omega_o \times)\zeta + (\mathbf{A} \times)\zeta^2 + (\mathbf{B} \times)\zeta^3]. \quad (5.80)$$

У кінці кроку інтегрування (тобто при $\zeta = 2$) величина матриці P набуде значення:

$$\boxed{P_{(1)}(2h) = E + h[2(\omega_o \times) + 2^2(\mathbf{A} \times) + 2^3(\mathbf{B} \times)]}. \quad (5.81)$$

Друге наближення

З виразу (5.69) зі врахуванням (5.79) виходить

$$\Delta P_{(2)}(\tau) = \int_0^\tau \Delta P_{(1)}(\xi) \cdot [\omega(\xi) \times] d\xi = h^2 \int_0^\xi [(\omega_0 \times) \zeta + (\mathbf{A} \times) \zeta^2 + (\mathbf{B} \times) \zeta^3] \times \\ \times [(\omega_0 \times) + 2(\mathbf{A} \times) \zeta + 3(\mathbf{B} \times) \zeta^2] d\zeta$$

Здійснюючи математичні дії, отримаємо:

$$\Delta P_{(2)}(\tau) = h^2 \left\{ \frac{1}{2} (\omega_0 \times)^2 \zeta^2 + \frac{1}{3} [2(\omega_0 \times)(\mathbf{A} \times) + (\mathbf{A} \times)(\omega_0 \times)] \zeta^3 + \right. \\ \left. + \frac{1}{4} [3(\omega_0 \times)(\mathbf{B} \times) + 2(\mathbf{A} \times)^2 + (\mathbf{B} \times)(\omega_0 \times)] \zeta^4 + \right. \\ \left. + \frac{1}{5} [3(\mathbf{A} \times)(\mathbf{B} \times) + 2(\mathbf{B} \times)(\mathbf{A} \times)] \zeta^5 + \frac{1}{6} [3(\mathbf{B} \times)^2] \zeta^6 \right\}.$$

Позначимо:

$$\begin{array}{c} \boxed{\mathbf{K}_{22} = \frac{1}{2} (\omega_0 \times)^2;} \quad \boxed{\mathbf{K}_{23} = \frac{1}{3} [2(\omega_0 \times)(\mathbf{A} \times) + (\mathbf{A} \times)(\omega_0 \times)];} \\ \boxed{\mathbf{K}_{24} = \frac{1}{4} [3(\omega_0 \times)(\mathbf{B} \times) + 2(\mathbf{A} \times)^2 + (\mathbf{B} \times)(\omega_0 \times)];} \quad \boxed{\mathbf{K}_{25} = \frac{1}{5} [3(\mathbf{A} \times)(\mathbf{B} \times) + 2(\mathbf{B} \times)(\mathbf{A} \times)];} \\ \boxed{\mathbf{K}_{26} = \frac{1}{6} [3(\mathbf{B} \times)^2].} \end{array}$$

Тому матимемо:

$$\Delta P_{(2)}(\tau) = h^2 \sum_{i=2}^6 \mathbf{K}_{2i} \zeta^i$$

Тепер двокроковий алгоритм у другому наближенні можна подати ($\zeta = 2$) у вигляді формули:

$$\boxed{P_{(2)}(2h) = E + h[2(\omega_0 \times) + 2^2(\mathbf{A} \times) + 2^3(\mathbf{B} \times)] + h^2 \sum_{i=2}^6 2^i \mathbf{K}_{2i}}. \quad (5.82)$$

Третє наближення

Додаткову частину, що виникає у третьому наближенні, отримаємо з співвідношення (5.70):

$$\Delta P_{(3)}(\tau) = \int_0^\tau \Delta P_{(2)}(\xi) \cdot [\omega(\xi) \times] d\xi = h^3 \int_0^\xi \left[h^2 \sum_{i=2}^6 \mathbf{K}_{2i} \zeta^i \right] \times [(\omega_0 \times) + 2(\mathbf{A} \times) \zeta + 3(\mathbf{B} \times) \zeta^2] d\zeta.$$

Здійснивши множення і інтегрування, знайдемо коефіцієнти \mathbf{K}_{3j} при кожному ζ^j :

$\mathbf{K}_{33} = \frac{1}{3} \mathbf{K}_{22}(\omega_0 \times) \zeta^3;$	$\mathbf{K}_{34} = \frac{1}{4} [2\mathbf{K}_{22}(\mathbf{A} \times) + \mathbf{K}_{23}(\omega_0 \times)];$
$\mathbf{K}_{35} = \frac{1}{5} [3\mathbf{K}_{22}(\mathbf{B} \times) + 2\mathbf{K}_{23}(\mathbf{A} \times) + \mathbf{K}_{24}(\omega_0 \times)];$	
$\mathbf{K}_{36} = \frac{1}{6} [3\mathbf{K}_{23}(\mathbf{B} \times) + 2\mathbf{K}_{24}(\mathbf{A} \times) + \mathbf{K}_{25}(\omega_0 \times)];$	$\mathbf{K}_{37} = \frac{1}{7} [3\mathbf{K}_{24}(\mathbf{B} \times) + 2\mathbf{K}_{25}(\mathbf{A} \times) + \mathbf{K}_{26}(\omega_0 \times)];$
$\mathbf{K}_{38} = \frac{1}{8} [3\mathbf{K}_{25}(\mathbf{B} \times) + 2\mathbf{K}_{26}(\mathbf{A} \times)];$	
$\mathbf{K}_{39} = \frac{1}{3} \mathbf{K}_{26}(\mathbf{B} \times).$	

Внаслідок цього отримаємо:

$$\Delta P_{(3)}(\tau) = h^3 \sum_{j=3}^9 \mathbf{K}_{3j} \zeta^j.$$

У кінці кроку інтегрування ($\zeta = 2$) ця матриця набуває значення:

$$\Delta P_{(3)}(h) = h^3 \sum_{j=3}^9 2^j \mathbf{K}_{3j}.$$

Додаючи цей доданок до правої частини формули (5.22) у кінці кроку інтегрування ($\zeta = 2$), прийдемо до формули двокрокового алгоритму у третьому наближенні:

$P_{(3)}(2h) = E + h[2(\omega_0 \times) + 2^2(\mathbf{A} \times) + 2^3(\mathbf{B} \times)] + h^2 \sum_{i=2}^6 2^i \mathbf{K}_{2i} + h^3 \sum_{j=3}^9 2^j \mathbf{K}_{3j}. \quad (5.83)$

Четверте наближення

У четвертому наближенні з співвідношення (5.71) знайдемо:

$$\begin{aligned} \Delta P_{(4)}(\tau) &= \int_0^\tau \Delta P_{(3)}(\xi) \cdot [\omega(\xi) \times] d\xi = h^4 \int_0^\xi \left[\sum_{j=3}^9 \mathbf{K}_{3j} \zeta^j \right] \times \\ &\quad \times [(\omega_0 \times) + 2(\mathbf{A} \times)\zeta + 3(\mathbf{B} \times)\zeta^2 + 4(\mathbf{C} \times)\zeta^3] d\zeta \end{aligned}$$

Звідси можна відшукати значення коефіцієнтів \mathbf{K}_{4k} при кожному ζ^k :

$\mathbf{K}_{44} = \frac{1}{4} \mathbf{K}_{33}(\omega_0 \times);$	$\mathbf{K}_{45} = \frac{1}{5} [2\mathbf{K}_{33}(\mathbf{A} \times) + \mathbf{K}_{34}(\omega_0 \times)];$
$\mathbf{K}_{46} = \frac{1}{6} [3\mathbf{K}_{33}(\mathbf{B} \times) + 2\mathbf{K}_{34}(\mathbf{A} \times) + \mathbf{K}_{35}(\omega_0 \times)];$	$\mathbf{K}_{47} = \frac{1}{7} [3\mathbf{K}_{34}(\mathbf{B} \times) + 2\mathbf{K}_{35}(\mathbf{A} \times) + \mathbf{K}_{36}(\omega_0 \times)];$

$\mathbf{K}_{48} = \frac{1}{8} [3\mathbf{K}_{35}(\mathbf{B} \times) + 2\mathbf{K}_{36}(\mathbf{A} \times) + \mathbf{K}_{37}(\boldsymbol{\omega}_0 \times)];$	$\mathbf{K}_{49} = \frac{1}{9} [3\mathbf{K}_{36}(\mathbf{B} \times) + 2\mathbf{K}_{37}(\mathbf{A} \times) + \mathbf{K}_{38}(\boldsymbol{\omega}_0 \times)];$
$\mathbf{K}_{410} = \frac{1}{10} [3\mathbf{K}_{37}(\mathbf{B} \times) + 2\mathbf{K}_{38}(\mathbf{A} \times) + \mathbf{K}_{39}(\boldsymbol{\omega}_0 \times)];$	$\mathbf{K}_{411} = \frac{1}{11} [3\mathbf{K}_{38}(\mathbf{B} \times) + 2\mathbf{K}_{39}(\mathbf{A} \times)];$
$\mathbf{K}_{412} = \frac{1}{12} [3\mathbf{K}_{39}(\mathbf{B} \times)];$	

В результаті виходить:

$$\Delta P_{(4)}(\tau) = h^4 \sum_{k=4}^{12} \mathbf{K}_{4k} \zeta^k .$$

Наприкінці кроку інтегрування ($\zeta = 2$) ця матриця набуде значення:

$$\Delta P_{(4)}(\tau) = h^4 \sum_{k=4}^{12} 2^k \mathbf{K}_{4k} .$$

Додаючи цей доданок до правої частини формули (5.74), дійдемо формули двокрокового алгоритму у четвертому наближенні:

$$\boxed{P_{(4)}(2h) = E + h[2(\boldsymbol{\omega}_0 \times) + 2^2(\mathbf{A} \times) + 2^3(\mathbf{B} \times)] + h^2 \sum_{i=2}^6 2^i \mathbf{K}_{2i} + h^3 \sum_{j=3}^9 2^j \mathbf{K}_{3j} + h^4 \sum_{k=4}^{12} 2^k \mathbf{K}_{4k}} \quad (5.84)$$

Трикрокові алгоритми

На трьох кроках опитування, що складають один крок інтегрування, відомими є чотири виміряні величини вектора кутової швидкості основи: $\boldsymbol{\omega}_0$ – на початку, $\boldsymbol{\omega}_1$, $\boldsymbol{\omega}_2$, $\boldsymbol{\omega}_3$ – на кінцях першого, другого і третього кроків опитування відповідно. У цьому випадку кутову швидкість слід шукати як кубічний поліном часу:

$$\boldsymbol{\omega}(\tau) = \boldsymbol{\omega}_o + 2\boldsymbol{\varepsilon}\tau + 3\mathbf{a}\tau^2 + 4\mathbf{b}\tau^3 .$$

Для знаходження коефіцієнтів $\boldsymbol{\varepsilon}$, \mathbf{a} , \mathbf{b} , підставимо у цей вираз час кінця кожного кроку опитування ($\boldsymbol{\omega}_0 = \boldsymbol{\omega}(0)$, $\boldsymbol{\omega}_1 = \boldsymbol{\omega}(h)$, $\boldsymbol{\omega}_2 = \boldsymbol{\omega}(2h)$, $\boldsymbol{\omega}_3 = \boldsymbol{\omega}(3h)$).

Прирости кутової швидкості на кожному з трьох кроків опитування матимуть вигляд:

$$\begin{aligned} \boldsymbol{\omega}_1 - \boldsymbol{\omega}_o &= 2\boldsymbol{\varepsilon}h + 3\mathbf{a}h^2 + 4\mathbf{b}h^3; & \boldsymbol{\omega}_2 - \boldsymbol{\omega}_o &= 4\boldsymbol{\varepsilon}h + 12\mathbf{a}h^2 + 32\mathbf{b}h^3; \\ \boldsymbol{\omega}_3 - \boldsymbol{\omega}_o &= 6\boldsymbol{\varepsilon}h + 27\mathbf{a}h^2 + 108\mathbf{b}h^3. \end{aligned}$$

Запишемо розв'язок цієї системи рівнянь у матричній формі:

$$\begin{bmatrix} \varepsilon h \\ \mathbf{a} h^2 \\ \mathbf{b} h^3 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & -\frac{3}{4} & \frac{1}{6} \\ -\frac{5}{6} & \frac{2}{3} & -\frac{1}{6} \\ \frac{1}{8} & -\frac{1}{8} & \frac{1}{24} \end{bmatrix} \times \begin{bmatrix} \boldsymbol{\omega}_1 - \boldsymbol{\omega}_o \\ \boldsymbol{\omega}_2 - \boldsymbol{\omega}_o \\ \boldsymbol{\omega}_3 - \boldsymbol{\omega}_o \end{bmatrix}. \quad (5.85)$$

Якщо ввести позначення:

$$\begin{aligned} \mathbf{A} &= \frac{1}{12}(-11\boldsymbol{\omega}_o + 18\boldsymbol{\omega}_1 - 9\boldsymbol{\omega}_2 + 2\boldsymbol{\omega}_3); & \mathbf{B} &= \frac{1}{6}(2\boldsymbol{\omega}_o - 5\boldsymbol{\omega}_1 + 4\boldsymbol{\omega}_2 - \boldsymbol{\omega}_3); \\ \mathbf{C} &= \frac{1}{24}(-\boldsymbol{\omega}_o + 3\boldsymbol{\omega}_1 - 3\boldsymbol{\omega}_2 + \boldsymbol{\omega}_3); \end{aligned} \quad (5.86)$$

то вираз кутової швидкості можна подати у вигляді:

$$\boldsymbol{\omega}(\tau) = \boldsymbol{\omega}_o + 2\mathbf{A}\zeta + 3\mathbf{B}\zeta^2 + 4\mathbf{C}\zeta^3, \quad \left(\zeta = \frac{\tau}{h} \right). \quad (5.87)$$

Перше наближення

З співвідношення (5.68) виходить у першому наближенні:

$$\Delta P_{(1)}(\tau) = \int_0^\tau [\boldsymbol{\omega}(\xi) \times] d\xi = \int_0^\zeta [(\boldsymbol{\omega}_o \times) + 2(\mathbf{A} \times)\zeta + 3(\mathbf{B} \times)\zeta^2 + 4(\mathbf{C} \times)\zeta^3] d(\zeta h),$$

або:

$$\Delta P_{(1)}(\tau) = h[(\boldsymbol{\omega}_o \times)\zeta + (\mathbf{A} \times)\zeta^2 + (\mathbf{B} \times)\zeta^3 + (\mathbf{C} \times)\zeta^4]. \quad (5.88)$$

В кінці кроку інтегрування (тобто при $\zeta = 3$) величина матриці P стає та-

кою:

$$P_{(1)}(3h) = E + h[3(\boldsymbol{\omega}_o \times) + 3^2(\mathbf{A} \times) + 3^3(\mathbf{B} \times) + 3^4(\mathbf{C} \times)]. \quad (5.89)$$

Друге наближення

Вираз (5.69) перетворюється у такий спосіб

$$\begin{aligned} \Delta P_{(2)}(\tau) &= \int_0^\tau \Delta P_{(1)}(\xi) \cdot [\boldsymbol{\omega}(\xi) \times] d\xi = h^2 \int_0^\zeta [(\boldsymbol{\omega}_o \times)\zeta + (\mathbf{A} \times)\zeta^2 + (\mathbf{B} \times)\zeta^3 + (\mathbf{C} \times)\zeta^4] \times \\ &\quad \times [(\boldsymbol{\omega}_o \times) + 2(\mathbf{A} \times)\zeta + 3(\mathbf{B} \times)\zeta^2 + 4(\mathbf{C} \times)\zeta^3] d\zeta \end{aligned}$$

Перемноживши і проінтегрувавши, отримаємо:

$$\Delta P_{(2)}(\tau) = h^2 \left\{ \begin{aligned} & \frac{1}{2}(\omega_0 \times)^2 \zeta^2 + \frac{1}{3}[2(\omega_0 \times)(A \times) + (A \times)(\omega_0 \times)]\zeta^3 + \frac{1}{4}[3(\omega_0 \times)(B \times) + 2(A \times)^2 + (B \times)(\omega_0 \times)]\zeta^4 + \\ & + \frac{1}{5}[4(\omega_0 \times)(C \times) + 3(A \times)(B \times) + 2(B \times)(A \times) + (C \times)(\omega_0 \times)]\zeta^5 + \\ & + \frac{1}{6}[4(A \times)(C \times) + 3(B \times)^2 + 2(C \times)(A \times)]\zeta^6 + \frac{1}{7}[4(B \times)(C \times) + 3(C \times)(B \times)]\zeta^7 + \frac{1}{8}[4(C \times)^2]\zeta^8 \end{aligned} \right\}$$

Введемо позначення:

$\mathbf{K}_{22} = \frac{1}{2}(\omega_0 \times)^2;$	$\mathbf{K}_{23} = \frac{1}{3}[2(\omega_0 \times)(A \times) + (A \times)(\omega_0 \times)];$
$\mathbf{K}_{24} = \frac{1}{4}[3(\omega_0 \times)(B \times) + 2(A \times)^2 + (B \times)(\omega_0 \times)];$	
$\mathbf{K}_{25} = \frac{1}{5}[4(\omega_0 \times)(C \times) + 3(A \times)(B \times) + 2(B \times)(A \times) + (C \times)(\omega_0 \times)];$	
$\mathbf{K}_{26} = \frac{1}{6}[4(A \times)(C \times) + 3(B \times)^2 + 2(C \times)(A \times)];$	$\mathbf{K}_{27} = \frac{1}{7}[4(B \times)(C \times) + 3(C \times)(B \times)];$
$\mathbf{K}_{28} = \frac{1}{2}(C \times)^2.$	

З врахуванням цього матимемо:

$$\Delta P_{(2)}(\tau) = h^2 \sum_{i=2}^8 \mathbf{K}_{2i} \zeta^i.$$

Тепер формулу трикрокового алгоритму у другому наближенні можна записати так ($\zeta = 3$):

$$P_{(2)}(3h) = E + h[3(\omega_0 \times) + 3^2(A \times) + 3^3(B \times) + 3^4(C \times)] + h^2 \sum_{i=2}^8 3^i \mathbf{K}_{2i}. \quad (5.90)$$

Третє наближення

З співвідношення (5.70) виходить:

$$\begin{aligned} \Delta P_{(3)}(\tau) &= \int_0^\tau \Delta P_{(2)}(\xi) \cdot [\omega(\xi) \times] d\xi = h^3 \int_0^\xi \left[h^2 \sum_{i=2}^8 \mathbf{K}_{2i} \zeta^i \right] \times \\ &\quad \times [(\omega_0 \times) + 2(A \times)\zeta + 3(B \times)\zeta^2 + 4(C \times)\zeta^3] d\zeta \end{aligned}$$

Здійснивши операції множення і інтегрування, знайдемо коефіцієнти \mathbf{K}_{3j} при кожному ζ^j :

$\mathbf{K}_{33} = \frac{1}{3}\mathbf{K}_{22}(\omega_0 \times);$	$\mathbf{K}_{34} = \frac{1}{4}[2\mathbf{K}_{22}(A \times) + \mathbf{K}_{23}(\omega_0 \times)];$
$\mathbf{K}_{35} = \frac{1}{5}[3\mathbf{K}_{22}(B \times) + 2\mathbf{K}_{23}(A \times) + \mathbf{K}_{24}(\omega_0 \times)];$	

$$\begin{aligned}
\mathbf{K}_{36} &= \frac{1}{6} [4\mathbf{K}_{22}(\mathbf{C} \times) + 3\mathbf{K}_{23}(\mathbf{B} \times) + 2\mathbf{K}_{24}(\mathbf{A} \times) + \mathbf{K}_{25}(\boldsymbol{\omega}_0 \times)]; \\
\mathbf{K}_{37} &= \frac{1}{7} [4\mathbf{K}_{23}(\mathbf{C} \times) + 3\mathbf{K}_{24}(\mathbf{B} \times) + 2\mathbf{K}_{25}(\mathbf{A} \times) + \mathbf{K}_{26}(\boldsymbol{\omega}_0 \times)]; \\
\mathbf{K}_{38} &= \frac{1}{8} [4\mathbf{K}_{24}(\mathbf{C} \times) + 3\mathbf{K}_{25}(\mathbf{B} \times) + 2\mathbf{K}_{26}(\mathbf{A} \times) + \mathbf{K}_{27}(\boldsymbol{\omega}_0 \times)]; \\
\mathbf{K}_{39} &= \frac{1}{9} [4\mathbf{K}_{25}(\mathbf{C} \times) + 3\mathbf{K}_{26}(\mathbf{B} \times) + 2\mathbf{K}_{27}(\mathbf{A} \times) + \mathbf{K}_{28}(\boldsymbol{\omega}_0 \times)]; \\
\mathbf{K}_{310} &= \frac{1}{10} [4\mathbf{K}_{26}(\mathbf{C} \times) + 3\mathbf{K}_{27}(\mathbf{B} \times) + 2\mathbf{K}_{28}(\mathbf{A} \times)]; & \mathbf{K}_{311} &= \frac{1}{11} [4\mathbf{K}_{27}(\mathbf{C} \times) + 3\mathbf{K}_{28}(\mathbf{B} \times)]; \\
\mathbf{K}_{312} &= \frac{1}{3} \mathbf{K}_{28}(\mathbf{C} \times).
\end{aligned}$$

В результаті виходить:

$$\Delta P_{(3)}(\tau) = h^3 \sum_{j=3}^{12} \mathbf{K}_{3j} \zeta^j.$$

У кінці кроку інтегрування ($\zeta = 3$) ця матриця стає рівною:

$$\Delta P_{(3)}(h) = h^3 \sum_{j=3}^{15} 3^j \mathbf{K}_{3j}.$$

Звідси випливає така формула трикрокового алгоритму у третьому наближенні:

$$\begin{aligned}
P_{(3)}(3h) &= E + h [3(\boldsymbol{\omega}_0 \times) + 3^2(\mathbf{A} \times) + 3^3(\mathbf{B} \times) + 3^4(\mathbf{C} \times)] + \\
&\quad + h^2 \sum_{i=2}^8 3^i \mathbf{K}_{2i} + h^3 \sum_{j=3}^{12} 3^j \mathbf{K}_{3j}.
\end{aligned} \tag{5.91}$$

Четверте наближення

З виразу (5.71) отримуємо:

$$\begin{aligned}
\Delta P_{(4)}(\tau) &= \int_0^{\tau} \Delta P_{(3)}(\xi) \cdot [\boldsymbol{\omega}(\xi) \times] d\xi = h^4 \int_0^{\xi} \left[\sum_{j=3}^{12} \mathbf{K}_{3j} \zeta^j \right] \times \\
&\quad \times [(\boldsymbol{\omega}_0 \times) + 2(\mathbf{A} \times)\zeta + 3(\mathbf{B} \times)\zeta^2 + 4(\mathbf{C} \times)\zeta^3] d\zeta.
\end{aligned}$$

Після інтегрування, використовуючи позначення коефіцієнтів \mathbf{K}_{4k} при кожному ζ^k :

$$\mathbf{K}_{44} = \frac{1}{4} \mathbf{K}_{33}(\boldsymbol{\omega}_0 \times);$$

$$\mathbf{K}_{45} = \frac{1}{5} [2\mathbf{K}_{33}(\mathbf{A} \times) + \mathbf{K}_{34}(\boldsymbol{\omega}_0 \times)];$$

$$\begin{aligned}
& \mathbf{K}_{46} = \frac{1}{6} [3\mathbf{K}_{33}(\mathbf{B} \times) + 2\mathbf{K}_{34}(\mathbf{A} \times) + \mathbf{K}_{35}(\boldsymbol{\omega}_0 \times)]; \\
& \mathbf{K}_{47} = \frac{1}{7} [4\mathbf{K}_{33}(\mathbf{C} \times) + 3\mathbf{K}_{34}(\mathbf{B} \times) + 2\mathbf{K}_{35}(\mathbf{A} \times) + \mathbf{K}_{36}(\boldsymbol{\omega}_0 \times)]; \\
& \mathbf{K}_{48} = \frac{1}{8} [4\mathbf{K}_{34}(\mathbf{C} \times) + 3\mathbf{K}_{35}(\mathbf{B} \times) + 2\mathbf{K}_{36}(\mathbf{A} \times) + \mathbf{K}_{37}(\boldsymbol{\omega}_0 \times)]; \\
& \mathbf{K}_{49} = \frac{1}{9} [4\mathbf{K}_{35}(\mathbf{C} \times) + 3\mathbf{K}_{36}(\mathbf{B} \times) + 2\mathbf{K}_{37}(\mathbf{A} \times) + \mathbf{K}_{38}(\boldsymbol{\omega}_0 \times)]; \\
& \mathbf{K}_{410} = \frac{1}{10} [4\mathbf{K}_{36}(\mathbf{C} \times) + 3\mathbf{K}_{37}(\mathbf{B} \times) + 2\mathbf{K}_{38}(\mathbf{A} \times) + \mathbf{K}_{39}(\boldsymbol{\omega}_0 \times)]; \\
& \mathbf{K}_{411} = \frac{1}{11} [4\mathbf{K}_{37}(\mathbf{C} \times) + 3\mathbf{K}_{38}(\mathbf{B} \times) + 2\mathbf{K}_{39}(\mathbf{A} \times) + \mathbf{K}_{310}(\boldsymbol{\omega}_0 \times)]; \\
& \mathbf{K}_{412} = \frac{1}{12} [4\mathbf{K}_{38}(\mathbf{C} \times) + 3\mathbf{K}_{39}(\mathbf{B} \times) + 2\mathbf{K}_{310}(\mathbf{A} \times) + \mathbf{K}_{311}(\boldsymbol{\omega}_0 \times)]; \\
& \mathbf{K}_{413} = \frac{1}{13} [4\mathbf{K}_{39}(\mathbf{C} \times) + 3\mathbf{K}_{310}(\mathbf{B} \times) + 2\mathbf{K}_{311}(\mathbf{A} \times) + \mathbf{K}_{312}(\boldsymbol{\omega}_0 \times)]; \\
& \mathbf{K}_{414} = \frac{1}{14} [4\mathbf{K}_{310}(\mathbf{C} \times) + 3\mathbf{K}_{311}(\mathbf{B} \times) + 2\mathbf{K}_{312}(\mathbf{A} \times)]; \quad \mathbf{K}_{415} = \frac{1}{15} [4\mathbf{K}_{311}(\mathbf{C} \times) + 3\mathbf{K}_{312}(\mathbf{B} \times)]; \\
& \mathbf{K}_{416} = \frac{1}{14} \mathbf{K}_{312}(\mathbf{C} \times);
\end{aligned}$$

дійдемо:

$$\Delta P_{(4)}(\tau) = h^4 \sum_{k=4}^{16} \mathbf{K}_{4k} \zeta^k$$

У кінці кроку інтегрування ($\zeta = 3$) ця матриця набуває значення:

$$\Delta P_{(4)}(\tau) = h^4 \sum_{k=4}^{16} 3^k \mathbf{K}_{4k}.$$

Додавши цей результат до правої частини формули (5.31), отримаємо формулу трикрокового алгоритму у четвертому наближенні:

$$\begin{aligned}
P_{(4)}(3h) = E + h [3(\boldsymbol{\omega}_0 \times) + 3^2(\mathbf{A} \times) + 3^3(\mathbf{B} \times) + 3^4(\mathbf{C} \times)] + \\
+ h^2 \sum_{i=2}^8 3^i \mathbf{K}_{2i} + h^3 \sum_{j=3}^{12} 3^j \mathbf{K}_{3j} + h^4 \sum_{k=4}^{16} 3^k \mathbf{K}_{4k}.
\end{aligned} \tag{5.92}$$

Чотирикрокові алгоритми

На чотирьох кроках опитування, які складають черговий крок інтегрування, стають відомими п'ять вимірних величин вектора кутової швидкості основи: $\boldsymbol{\omega}_0$ –

на початку, $\omega_1, \omega_2, \omega_3, \omega_4$ – на кінцях другого, третього та четвертого кроків. Тому кутову швидкість треба апроксимувати поліномом четвертого порядку:

$$\omega(\tau) = \omega_o + 2\varepsilon\tau + 3a\tau^2 + 4b\tau^3 + 5c\tau^4. \quad (5.93)$$

Щоб знайти векторні коефіцієнти ε, a, b, c , підставимо у цей вираз час кінця кожного кроку опитування ($\omega_0 = \omega(0)$, $\omega_1 = \omega(h)$, $\omega_2 = \omega(2h)$, $\omega_3 = \omega(3h)$, $\omega_4 = \omega(4h)$).

Прирости кутової швидкості на кожному з чотирьох кроків опитування матимуть вигляд:

$$\begin{aligned} \omega_1 - \omega_o &= 2\varepsilon h + 3ah^2 + 4bh^3 + 5ch^4; \\ \omega_2 - \omega_o &= 4\varepsilon h + 12ah^2 + 32bh^3 + 80ch^4; \\ \omega_3 - \omega_o &= 6\varepsilon h + 27ah^2 + 108bh^3 + 405ch^4; \\ \omega_4 - \omega_o &= 8\varepsilon h + 48ah^2 + 256bh^3 + 1280ch^4. \end{aligned}$$

Якщо розв'язати цю систему рівнянь відносно векторних коефіцієнтів, то кутову швидкість можна подати у формі:

$$\omega(\tau) = \omega_o + 2\mathbf{A}\zeta + 3\mathbf{B}\zeta^2 + 4\mathbf{C}\zeta^3 + 5\mathbf{D}\zeta^4, \quad \left(\zeta = \frac{\tau}{h} \right). \quad (5.94)$$

де використані позначення

$\mathbf{A} = \frac{1}{24}(-25\omega_o + 48\omega_1 - 36\omega_2 + 16\omega_3 - 3\omega_4);$	$\mathbf{C} = \frac{1}{48}(-5\omega_o + 18\omega_1 - 24\omega_2 + 14\omega_3 - 3\omega_4);$
$\mathbf{B} = \frac{1}{72}(35\omega_o - 104\omega_1 + 114\omega_2 - 56\omega_3 + 35\omega_4);$	$\mathbf{D} = \frac{1}{120}(\omega_o - 4\omega_1 + 6\omega_2 - 4\omega_3 + \omega_4);$

(5.95)

Перше наближення

У першому наближенні у відповідності до співвідношення (5.68) отримаємо:

$$\Delta P_{(1)}(\tau) = \int_0^\tau [\omega(\xi) \times] d\xi = \int_0^\zeta [(\omega_o \times) + 2(\mathbf{A} \times)\zeta + 3(\mathbf{B} \times)\zeta^2 + 4(\mathbf{C} \times)\zeta^3 + 5(\mathbf{D} \times)\zeta^4] d(\zeta h),$$

або

$$\Delta P_{(1)}(\tau) = h[(\omega_o \times)\zeta + (\mathbf{A} \times)\zeta^2 + (\mathbf{B} \times)\zeta^3 + (\mathbf{C} \times)\zeta^4 + (\mathbf{D} \times)\zeta^5]. \quad (5.96)$$

У кінці кроку інтегрування (тобто при $\zeta = 4$) величина матриці P стане такою:

$$P_{(1)}(4h) = E + h[4(\omega_0 \times) + 4^2(\mathbf{A} \times) + 4^3(\mathbf{B} \times) + 4^4(\mathbf{C} \times) + 4^5(\mathbf{D} \times)]. \quad (5.97)$$

Друге наближення

Вираз (5.69) трансформується у таке:

$$\Delta P_{(2)}(\tau) = \int_0^\tau \Delta P_{(1)}(\xi) \cdot [\omega(\xi) \times] d\xi = h^2 \int_0^\zeta [(\omega_0 \times)\zeta + (\mathbf{A} \times)\zeta^2 + (\mathbf{B} \times)\zeta^3 + (\mathbf{C} \times)\zeta^4 + (\mathbf{D} \times)\zeta^5] \times \\ \times [(\omega_0 \times) + 2(\mathbf{A} \times)\zeta + 3(\mathbf{B} \times)\zeta^2 + 4(\mathbf{C} \times)\zeta^3 + 5(\mathbf{D} \times)\zeta^4] d\zeta$$

Перемноживши і проінтегрувавши, дійдемо:

$$\Delta P_{(2)}(\tau) = h^2 \left\{ \begin{aligned} & \frac{1}{2}(\omega_0 \times)^2 \zeta^2 + \frac{1}{3}[2(\omega_0 \times)(\mathbf{A} \times) + (\mathbf{A} \times)(\omega_0 \times)]\zeta^3 + \frac{1}{4}[3(\omega_0 \times)(\mathbf{B} \times) + 2(\mathbf{A} \times)^2 + (\mathbf{B} \times)(\omega_0 \times)]\zeta^4 + \\ & + \frac{1}{5}[4(\omega_0 \times)(\mathbf{C} \times) + 3(\mathbf{A} \times)(\mathbf{B} \times) + 2(\mathbf{B} \times)(\mathbf{A} \times) + (\mathbf{C} \times)(\omega_0 \times)]\zeta^5 + \\ & + \frac{1}{6}[5(\omega_0 \times)(\mathbf{D} \times) + 4(\mathbf{A} \times)(\mathbf{C} \times) + 3(\mathbf{B} \times)^2 + 2(\mathbf{C} \times)(\mathbf{A} \times) + (\mathbf{D} \times)(\omega_0 \times)]\zeta^6 + \\ & + \frac{1}{7}[5(\mathbf{A} \times)(\mathbf{D} \times) + 4(\mathbf{B} \times)(\mathbf{C} \times) + 3(\mathbf{C} \times)(\mathbf{B} \times) + 2(\mathbf{D} \times)(\mathbf{A} \times)]\zeta^7 + \\ & + \frac{1}{8}[5(\mathbf{B} \times)(\mathbf{D} \times) + 4(\mathbf{C} \times)^2 + 3(\mathbf{D} \times)(\mathbf{B} \times)]\zeta^8 + \frac{1}{9}[5(\mathbf{C} \times)(\mathbf{D} \times) + 4(\mathbf{D} \times)(\mathbf{C} \times)]\zeta^9 + \frac{1}{10}[5(\mathbf{D} \times)^2]\zeta^{10} \end{aligned} \right\}$$

Введемо позначення:

$$\begin{aligned} \mathbf{K}_{22} &= \frac{1}{2}(\omega_0 \times)^2; & \mathbf{K}_{23} &= \frac{1}{3}[2(\omega_0 \times)(\mathbf{A} \times) + (\mathbf{A} \times)(\omega_0 \times)]; \\ \mathbf{K}_{24} &= \frac{1}{4}[3(\omega_0 \times)(\mathbf{B} \times) + 2(\mathbf{A} \times)^2 + (\mathbf{B} \times)(\omega_0 \times)]; \\ \mathbf{K}_{25} &= \frac{1}{5}[4(\omega_0 \times)(\mathbf{C} \times) + 3(\mathbf{A} \times)(\mathbf{B} \times) + 2(\mathbf{B} \times)(\mathbf{A} \times) + (\mathbf{C} \times)(\omega_0 \times)]; \\ \mathbf{K}_{26} &= \frac{1}{6}[5(\omega_0 \times)(\mathbf{D} \times) + 4(\mathbf{A} \times)(\mathbf{C} \times) + 3(\mathbf{B} \times)^2 + 2(\mathbf{C} \times)(\mathbf{A} \times) + (\mathbf{D} \times)(\omega_0 \times)]; \\ \mathbf{K}_{27} &= \frac{1}{7}[5(\mathbf{A} \times)(\mathbf{D} \times) + 4(\mathbf{B} \times)(\mathbf{C} \times) + 3(\mathbf{C} \times)(\mathbf{B} \times) + 2(\mathbf{D} \times)(\mathbf{A} \times)]; \\ \mathbf{K}_{28} &= \frac{1}{8}[5(\mathbf{B} \times)(\mathbf{D} \times) + 4(\mathbf{C} \times)^2 + 3(\mathbf{D} \times)(\mathbf{B} \times)]; & \mathbf{K}_{29} &= \frac{1}{9}[5(\mathbf{C} \times)(\mathbf{D} \times) + 4(\mathbf{D} \times)(\mathbf{C} \times)]; \\ \mathbf{K}_{210} &= \frac{1}{2}[(\mathbf{D} \times)^2] \end{aligned}$$

Після цього виходить:

$$\Delta P_{(2)}(\tau) = h^2 \sum_{i=2}^{10} \zeta^i \mathbf{K}_{2i}.$$

Тепер формула чотирикрокового алгоритму у другому наближенні можна подати так ($\zeta = 4$):

$$P_{(2)}(4h) = E + h[4(\omega_0 \times) + 4^2(\mathbf{A} \times) + 4^3(\mathbf{B} \times) + 4^4(\mathbf{C} \times) + 4^5(\mathbf{D} \times)] + h^2 \sum_{i=2}^{10} 4^i \mathbf{K}_{2i}. \quad (5.98)$$

Третє наближення

Застосовуючи співвідношення (5.70), матимемо:

$$\Delta P_{(3)}(\tau) = \int_0^\tau \Delta P_{(2)}(\xi) \cdot [\omega(\xi) \times] d\xi = h^3 \int_0^\zeta \left[h^2 \sum_{i=2}^{10} \zeta^i \mathbf{K}_{2i} \right] \times \\ \times [(\omega_0 \times) + 2(\mathbf{A} \times)\zeta + 3(\mathbf{B} \times)\zeta^2 + 4(\mathbf{C} \times)\zeta^3 + 5(\mathbf{D} \times)\zeta^4] d\zeta$$

Після перемноження та інтегрування знайдемо коефіцієнти \mathbf{K}_{3j} при кожному ζ^j :

$$\begin{aligned} \mathbf{K}_{33} &= \frac{1}{3} \mathbf{K}_{22}(\omega_0 \times); & \mathbf{K}_{34} &= \frac{1}{4} [2\mathbf{K}_{22}(\mathbf{A} \times) + \mathbf{K}_{23}(\omega_0 \times)]; \\ \mathbf{K}_{35} &= \frac{1}{5} [3\mathbf{K}_{22}(\mathbf{B} \times) + 2\mathbf{K}_{23}(\mathbf{A} \times) + \mathbf{K}_{24}(\omega_0 \times)]; \\ \mathbf{K}_{36} &= \frac{1}{6} [4\mathbf{K}_{22}(\mathbf{C} \times) + 3\mathbf{K}_{23}(\mathbf{B} \times) + 2\mathbf{K}_{24}(\mathbf{A} \times) + \mathbf{K}_{25}(\omega_0 \times)]; \\ \mathbf{K}_{37} &= \frac{1}{7} [5\mathbf{K}_{22}(\mathbf{D} \times) + 4\mathbf{K}_{23}(\mathbf{C} \times) + 3\mathbf{K}_{24}(\mathbf{B} \times) + 2\mathbf{K}_{25}(\mathbf{A} \times) + \mathbf{K}_{26}(\omega_0 \times)]; \\ \mathbf{K}_{38} &= \frac{1}{8} [5\mathbf{K}_{23}(\mathbf{D} \times) + 4\mathbf{K}_{24}(\mathbf{C} \times) + 3\mathbf{K}_{25}(\mathbf{B} \times) + 2\mathbf{K}_{26}(\mathbf{A} \times) + \mathbf{K}_{27}(\omega_0 \times)]; \\ \mathbf{K}_{39} &= \frac{1}{9} [5\mathbf{K}_{24}(\mathbf{D} \times) + 4\mathbf{K}_{25}(\mathbf{C} \times) + 3\mathbf{K}_{26}(\mathbf{B} \times) + 2\mathbf{K}_{27}(\mathbf{A} \times) + \mathbf{K}_{28}(\omega_0 \times)]; \\ \mathbf{K}_{310} &= \frac{1}{10} [5\mathbf{K}_{25}(\mathbf{D} \times) + 4\mathbf{K}_{26}(\mathbf{C} \times) + 3\mathbf{K}_{27}(\mathbf{B} \times) + 2\mathbf{K}_{28}(\mathbf{A} \times) + \mathbf{K}_{29}(\omega_0 \times)]; \\ \mathbf{K}_{311} &= \frac{1}{11} [5\mathbf{K}_{26}(\mathbf{D} \times) + 4\mathbf{K}_{27}(\mathbf{C} \times) + 3\mathbf{K}_{28}(\mathbf{B} \times) + 2\mathbf{K}_{29}(\mathbf{A} \times) + \mathbf{K}_{210}(\omega_0 \times)]; \\ \mathbf{K}_{312} &= \frac{1}{12} [5\mathbf{K}_{27}(\mathbf{D} \times) + 4\mathbf{K}_{28}(\mathbf{C} \times) + 3\mathbf{K}_{29}(\mathbf{B} \times) + 2\mathbf{K}_{210}(\mathbf{A} \times)]; \\ \mathbf{K}_{313} &= \frac{1}{13} [5\mathbf{K}_{28}(\mathbf{D} \times) + 4\mathbf{K}_{29}(\mathbf{C} \times) + 3\mathbf{K}_{210}(\mathbf{B} \times)]; & \mathbf{K}_{314} &= \frac{1}{14} [5\mathbf{K}_{29}(\mathbf{D} \times) + 4\mathbf{K}_{210}(\mathbf{C} \times)]; \\ \mathbf{K}_{315} &= \frac{1}{3} [\mathbf{K}_{210}(\mathbf{D} \times)]; \end{aligned}$$

Звідси випливає:

$$\Delta P_{(3)}(\tau) = h^3 \sum_{j=3}^{15} \zeta^j \mathbf{K}_{3j}.$$

Тому формула чотирикрокового алгоритму у третьому наближенні матиме вигляд:

$$P_{(3)}(4h) = E + h[4(\omega_0 \times) + 4^2(\mathbf{A} \times) + 4^3(\mathbf{B} \times) + 4^4(\mathbf{C} \times) + 4^5(\mathbf{D} \times)] + h^2 \sum_{i=2}^{10} 4^i \mathbf{K}_{2i} + h^3 \sum_{j=3}^{15} 4^j \mathbf{K}_{3j} \quad (5.99)$$

Четверте наближення

Доданок четвертого наближення виходить з співвідношення (5.71):

$$\Delta P_{(4)}(\tau) = \int_0^\tau \Delta P_{(3)}(\xi) \cdot [\omega(\xi) \times] d\xi = h^4 \int_0^\xi \left[\sum_{j=3}^{15} \zeta^j \mathbf{K}_{3j} \right] \times \left[(\omega_0 \times) + 2(\mathbf{A} \times)\zeta + 3(\mathbf{B} \times)\zeta^2 + 4(\mathbf{C} \times)\zeta^3 + 5(\mathbf{D} \times)\zeta^4 \right] d\zeta$$

Перемноживши та проінтегрувавши, знайдемо коефіцієнти \mathbf{K}_{4k} при кожному

ζ^k :

$$\begin{aligned} \mathbf{K}_{44} &= \frac{1}{4} \mathbf{K}_{33}(\omega_0 \times); & \mathbf{K}_{45} &= \frac{1}{5} [2\mathbf{K}_{33}(\mathbf{A} \times) + \mathbf{K}_{34}(\omega_0 \times)]; \\ \mathbf{K}_{46} &= \frac{1}{6} [3\mathbf{K}_{33}(\mathbf{B} \times) + 2\mathbf{K}_{34}(\mathbf{A} \times) + \mathbf{K}_{35}(\omega_0 \times)]; \\ \mathbf{K}_{47} &= \frac{1}{7} [4\mathbf{K}_{33}(\mathbf{C} \times) + 3\mathbf{K}_{34}(\mathbf{B} \times) + 2\mathbf{K}_{35}(\mathbf{A} \times) + \mathbf{K}_{36}(\omega_0 \times)]; \\ \mathbf{K}_{48} &= \frac{1}{8} [5\mathbf{K}_{33}(\mathbf{D} \times) + 4\mathbf{K}_{34}(\mathbf{C} \times) + 3\mathbf{K}_{35}(\mathbf{B} \times) + 2\mathbf{K}_{36}(\mathbf{A} \times) + \mathbf{K}_{37}(\omega_0 \times)]; \\ \mathbf{K}_{49} &= \frac{1}{9} [5\mathbf{K}_{34}(\mathbf{D} \times) + 4\mathbf{K}_{35}(\mathbf{C} \times) + 3\mathbf{K}_{36}(\mathbf{B} \times) + 2\mathbf{K}_{37}(\mathbf{A} \times) + \mathbf{K}_{38}(\omega_0 \times)]; \\ \mathbf{K}_{410} &= \frac{1}{10} [5\mathbf{K}_{35}(\mathbf{D} \times) + 4\mathbf{K}_{36}(\mathbf{C} \times) + 3\mathbf{K}_{37}(\mathbf{B} \times) + 2\mathbf{K}_{38}(\mathbf{A} \times) + \mathbf{K}_{39}(\omega_0 \times)]; \\ \mathbf{K}_{411} &= \frac{1}{11} [5\mathbf{K}_{36}(\mathbf{D} \times) + 4\mathbf{K}_{37}(\mathbf{C} \times) + 3\mathbf{K}_{38}(\mathbf{B} \times) + 2\mathbf{K}_{39}(\mathbf{A} \times) + \mathbf{K}_{310}(\omega_0 \times)]; \\ \mathbf{K}_{412} &= \frac{1}{12} [5\mathbf{K}_{37}(\mathbf{D} \times) + 4\mathbf{K}_{38}(\mathbf{C} \times) + 3\mathbf{K}_{39}(\mathbf{B} \times) + 2\mathbf{K}_{310}(\mathbf{A} \times) + \mathbf{K}_{311}(\omega_0 \times)]; \\ \mathbf{K}_{413} &= \frac{1}{13} [5\mathbf{K}_{38}(\mathbf{D} \times) + 4\mathbf{K}_{39}(\mathbf{C} \times) + 3\mathbf{K}_{310}(\mathbf{B} \times) + 2\mathbf{K}_{311}(\mathbf{A} \times) + \mathbf{K}_{312}(\omega_0 \times)]; \\ \mathbf{K}_{414} &= \frac{1}{14} [5\mathbf{K}_{39}(\mathbf{D} \times) + 4\mathbf{K}_{310}(\mathbf{C} \times) + 3\mathbf{K}_{311}(\mathbf{B} \times) + 2\mathbf{K}_{312}(\mathbf{A} \times) + \mathbf{K}_{313}(\omega_0 \times)]; \\ \mathbf{K}_{415} &= \frac{1}{15} [5\mathbf{K}_{310}(\mathbf{D} \times) + 4\mathbf{K}_{311}(\mathbf{C} \times) + 3\mathbf{K}_{312}(\mathbf{B} \times) + 2\mathbf{K}_{313}(\mathbf{A} \times) + \mathbf{K}_{314}(\omega_0 \times)]; \\ \mathbf{K}_{416} &= \frac{1}{16} [5\mathbf{K}_{311}(\mathbf{D} \times) + 4\mathbf{K}_{312}(\mathbf{C} \times) + 3\mathbf{K}_{313}(\mathbf{B} \times) + 2\mathbf{K}_{314}(\mathbf{A} \times) + \mathbf{K}_{315}(\omega_0 \times)]; \end{aligned}$$

$$\begin{aligned} \mathbf{K}_{417} &= \frac{1}{18} [5\mathbf{K}_{312}(\mathbf{D} \times) + 4\mathbf{K}_{313}(\mathbf{C} \times) + 3\mathbf{K}_{314}(\mathbf{B} \times) + 2\mathbf{K}_{315}(\mathbf{A} \times)]; \\ \mathbf{K}_{418} &= \frac{1}{18} [5\mathbf{K}_{313}(\mathbf{D} \times) + 4\mathbf{K}_{314}(\mathbf{C} \times) + 3\mathbf{K}_{315}(\mathbf{B} \times)]; \\ \mathbf{K}_{419} &= \frac{1}{19} [5\mathbf{K}_{314}(\mathbf{D} \times) + 4\mathbf{K}_{315}(\mathbf{C} \times)]; & \mathbf{K}_{420} &= \frac{1}{4} [\mathbf{K}_{315}(\mathbf{D} \times)] \end{aligned}$$

В результаті отримаємо:

$$\Delta P_{(4)}(\tau) = h^4 \sum_{k=4}^{20} \zeta^k \mathbf{K}_{4k}.$$

Загальна формула алгоритму четвертого порядку четвертого наближення матиме вигляд ($\zeta = 4$):

$$\begin{aligned} P_{(4)}(4h) = E + h & \left[4(\boldsymbol{\omega}_0 \times) + 4^2(\mathbf{A} \times) + 4^3(\mathbf{B} \times) + 4^4(\mathbf{C} \times) + 4^5(\mathbf{D} \times) \right] + \\ & + h^2 \sum_{i=2}^{10} 4^i \mathbf{K}_{2i} + h^3 \sum_{j=3}^{15} 4^j \mathbf{K}_{3j} + h^4 \sum_{k=4}^{20} 4^k \mathbf{K}_{4k}. \end{aligned} \quad (5.100)$$

Дослідження похибок алгоритмів

Результати модельного дослідження похибок алгоритмів, що спираються на вимірювання кутової швидкості подані у графічній формі на рисунках 5.3, 5.4, 5.5, та 5.6.

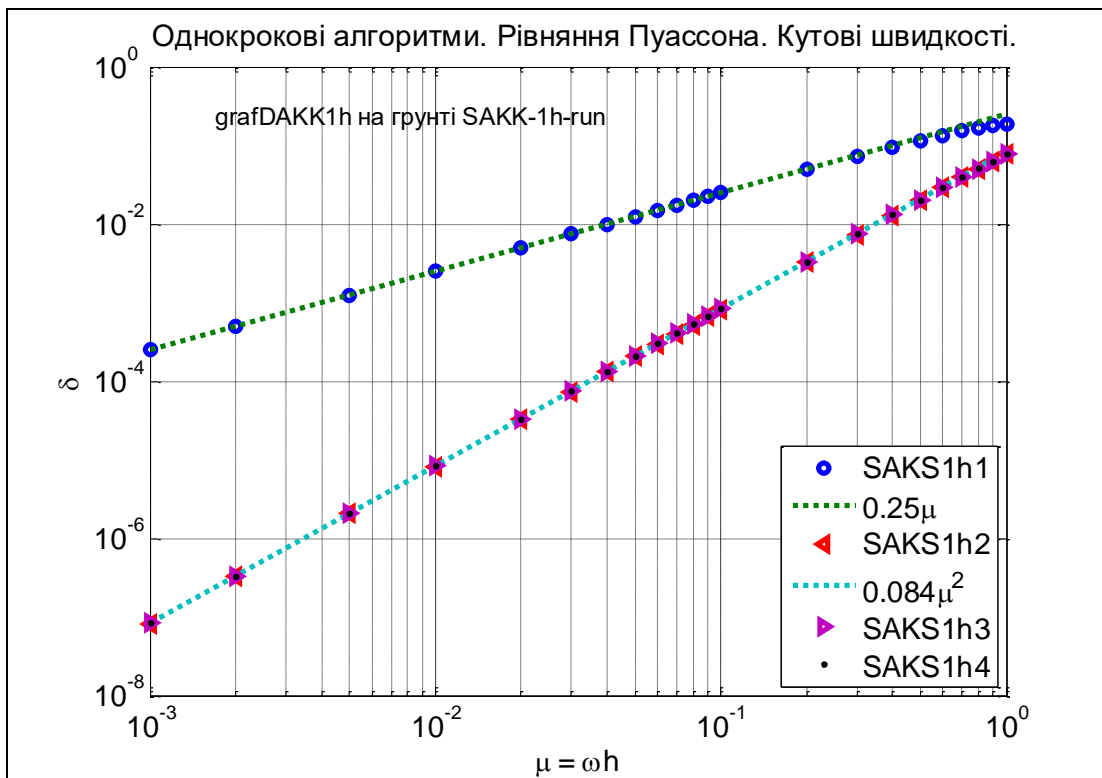


Рис. 5.3. Однокрокові алгоритми

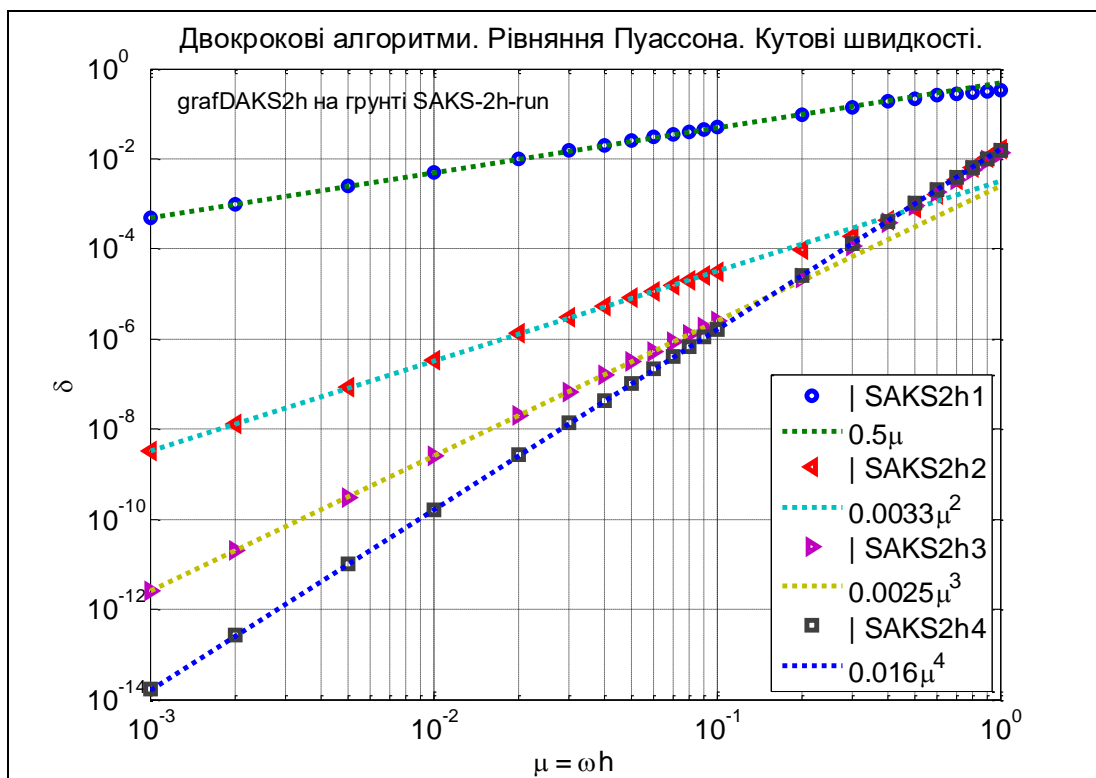


Рис. 5.4. Двокрокові алгоритми

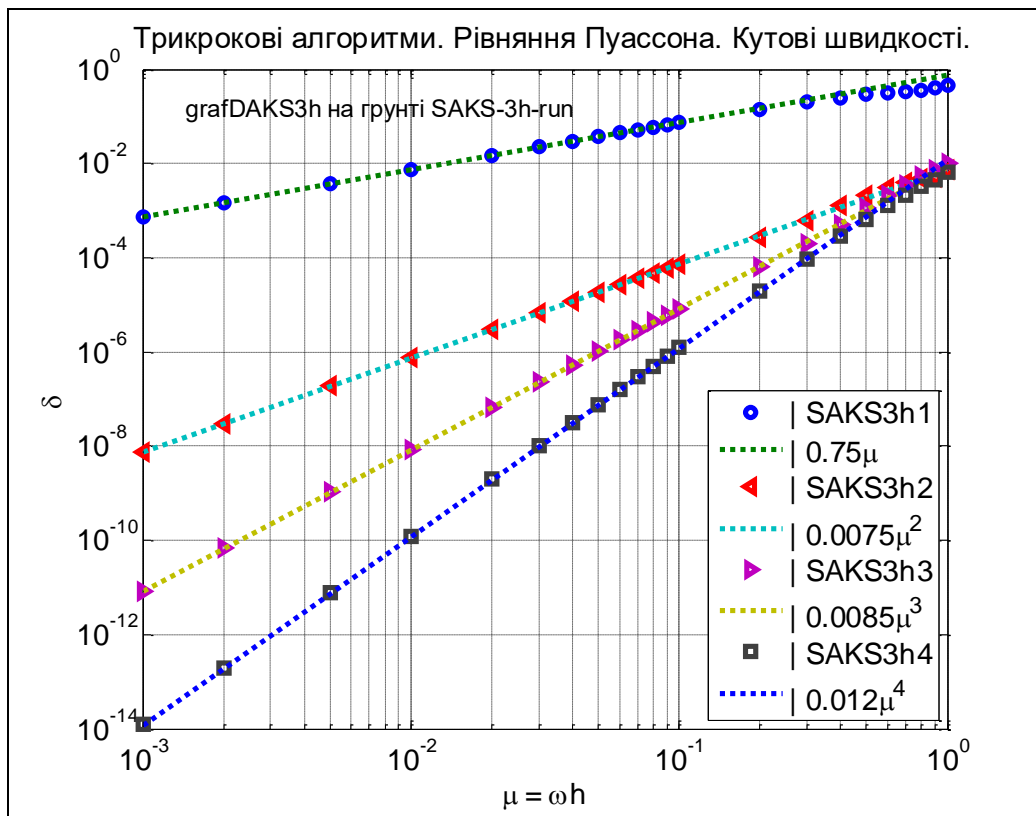


Рис. 5.5. Трикрокові алгоритми

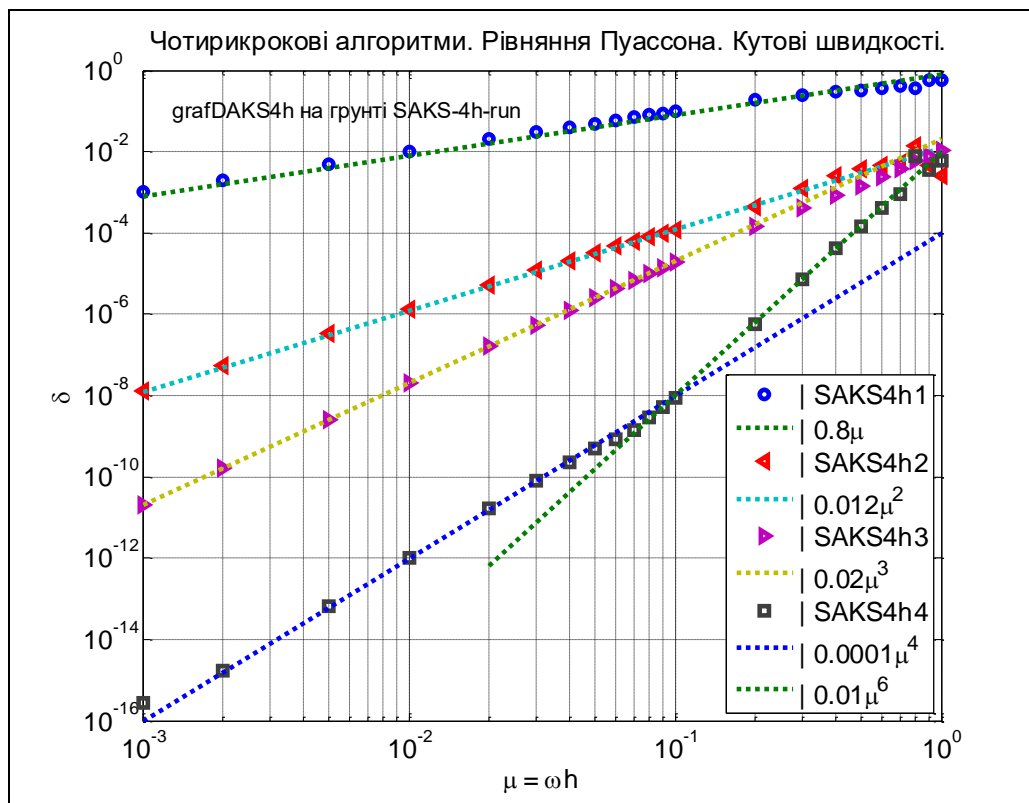


Рис. 5.6. Чотирикрокові алгоритми

На рисунку 5.3 зображені експериментальні залежності дрейфів однокрокових алгоритмів, визначених формулами (5.74)...(5.77). Рисунок 5.4 подає аналогічні залежності для двокрокових алгоритмів за формулами (5.81)...(5.84), рисунок 5.5 – трикрокових алгоритмів за формулами (5.89)...(5.92), а рисунок 5.6 – чотирікрокових алгоритмів (формули (5.97)...(5.100)).

На основі отриманих графіків можна знайти параметри точності досліджених алгоритмів. Вони наведені у таблицях 5.3, 5.4, 5.5, 5.6.

Таблиця 5.3. Параметри точності однокрокових алгоритмів при вимірюванні кутової швидкості

Алгоритм	Кроковість	Наближення	Порядок точності N	k
SAKS1h1	1	1	1	0,25
SAKS1h2	1	2	2	0,084
SAKS1h3	1	3	2	0,084
SAKS1h4	1	4	2	0,084

Таблиця 5.4. Параметри точності двокрокових алгоритмів при вимірюванні кутової швидкості

Алгоритм	Кроковість	Наближення	Порядок точності N	k
SAKS2h1	2	1	1	0,5
SAKS2h2	2	2	2	0,033
SAKS2h3	2	3	3	0,025
SAKS2h4	2	4	4	0,084

Таблиця 5.5. Параметри точності трикрокових алгоритмів при вимірюванні кутової швидкості

Алгоритм	Кроковість	Наближення	Порядок точності N	k
SAKS3h1	3	1	1	0,75
SAKS3h2	3	2	2	0,0075
SAKS3h3	3	3	3	0,0085
SAKS3h4	3	4	4	0,012

Таблиця 5.6. Параметри точності чотирикрокових алгоритмів при вимірюванні кутової швидкості

Алгоритм	Кроковість	Наближення	Порядок точності N	k
SAKS4h1	4	1	1	0,8
SAKS4h2	4	2	2	0,012
SAKS4h3	4	3	3	0,02
SAKS4h4	4	4	4	0,0001

Висновки

Отримані результати дозволяють зробити такі висновки:

- 1) точність однокрокових алгоритмів SAKS1h2 (Пуассон) і KMNU1h (Ейлер) практично однакова;
- 2) точність двокрокового алгоритму SAKS2h4 (Пуассон) перевищує точність двокрокового алгоритму KMNU2h (Ейлер) у 5 разів;
- 3) точності трикрокових алгоритмів SAKS3h4 (Пуассон) і KMNU3h (Ейлер) є близькими;
- 4) чотирикроковий алгоритм SAKS3h4 (Пуассон) є найкращим за точністю серед усіх розглянутих алгоритмів;
- 5) при цьому відповідні алгоритми на ґрунті рівняння Пуассона набагато складніше і потребують значно більшого комп'ютерного часу.

5.3.2. Вимірюються прирости квазікоординат

Однокрокові алгоритми

У випадку однокрокового алгоритму, оскільки відоме лише одно виміряне за цей єдиний крок опитування значення – приріст \mathbf{q}_1 квазікоординат, слід припустити, що вектор кутової швидкості основи є незмінним на цьому кроці і дорівнює:

$$\boldsymbol{\omega}(\tau) = \boldsymbol{\omega}_o = \text{const}.$$

Щоб відшукати невідомий векторний коефіцієнт ω_o , обчислимо приріст квазікоординат:

$$\mathbf{q}_1 = \int_0^h \omega(\zeta) d\zeta = \omega_o h,$$

з чого випливає:

$$\omega_o = \frac{1}{h} \mathbf{q}_1. \quad (5.101)$$

Тепер з співвідношення (5.68) виходить, що у першому наближенні матимемо:

$$\Delta P_{(1)}(\tau) = \int_0^\tau [\omega(\zeta) \times] d\zeta = \frac{1}{h} (\mathbf{q}_1 \times) \tau.$$

У кінці кроку інтегрування (при $\tau = h$) величина матриці набуде вигляду:

$$\boxed{P_{(1)}(h) = E + (\mathbf{q}_1 \times)}. \quad (5.102)$$

Для знаходження однокрокового алгоритму другого наближення, обчислимо додатковий доданок (5.69):

$$\Delta P_{(2)}(\tau) = \int_0^\tau \frac{1}{h} (\mathbf{q}_1 \times) \xi \cdot \left[\frac{1}{h} (\mathbf{q}_1 \times) \right] d\xi = \frac{1}{h^2} (\mathbf{q}_1 \times)^2 \int_0^\tau \xi \cdot d\xi = \frac{1}{2} (\mathbf{q}_1 \times)^2 \left(\frac{\tau}{h} \right)^2.$$

Тому формула алгоритму у другому наближенні виходить такою:

$$\boxed{P_{(2)}(h) = E + (\mathbf{q}_1 \times) + \frac{1}{2} (\mathbf{q}_1 \times)^2}. \quad (5.103)$$

У третьому наближенні отримаємо:

$$\Delta P_{(3)}(\tau) = \frac{1}{h^3} \int_0^\tau \frac{1}{2} (\mathbf{q}_1 \times)^2 \zeta^2 \cdot [(\mathbf{q}_1 \times)] d\zeta = \frac{1}{6} (\mathbf{q}_1 \times)^3 \left(\frac{\tau}{h} \right)^3,$$

а тому формула однокрокового алгоритму у третьому наближенні стає наступною:

$$\boxed{P_{(3)}(h) = E + (\mathbf{q}_1 \times) + \frac{1}{2} (\mathbf{q}_1 \times)^2 + \frac{1}{6} (\mathbf{q}_1 \times)^3}. \quad (5.104)$$

Аналогічно, у четвертому наближенні матимемо:

$$\Delta P_{(4)}(\tau) = \int_0^\tau \frac{1}{6} (\mathbf{q}_1 \times)^3 \zeta^3 \cdot [(\mathbf{q}_1 \times)] d\zeta = \frac{1}{6} (\mathbf{q}_1 \times)^4 \int_0^\tau \zeta^3 \cdot d\zeta = \frac{1}{24} (\mathbf{q}_1 \times)^4 \zeta^4. \quad (5.105)$$

Тут і далі використовується позначення відносного часу

$$\zeta = \frac{\tau}{h}. \quad (5.106)$$

Формула алгоритму у четвертому наближенні ($\zeta = 1$) набуває вигляду:

$$P_{(4)}(h) = E + (\mathbf{q}_1 \times) + \frac{1}{2}(\mathbf{q}_1 \times)^2 + \frac{1}{6}(\mathbf{q}_1 \times)^3 + \frac{1}{24}(\mathbf{q}_1 \times)^4. \quad (5.107)$$

Двокрокові алгоритми

Якщо шуканий алгоритм є двокроковим, шукатимемо кутову швидкість у вигляді лінійної функції часу:

$$\boldsymbol{\omega}(\tau) = \boldsymbol{\omega}_o + 2\boldsymbol{\varepsilon}\tau,$$

оскільки у цьому випадку відомі два значення приростів квазікоординат за два кроки опитування, що складають один крок інтегрування.

З цього маємо:

$$\mathbf{q}_1 = \int_0^h (\boldsymbol{\omega}_o + 2\boldsymbol{\varepsilon}\xi) d\xi = \boldsymbol{\omega}_o h + \boldsymbol{\varepsilon} h^2; \quad \mathbf{q}_2 = \int_h^{2h} (\boldsymbol{\omega}_o + 2\boldsymbol{\varepsilon}\xi) d\xi = \boldsymbol{\omega}_o h + 3\boldsymbol{\varepsilon} h^2.$$

Розв'язуючи ці лінійні рівняння, отримаємо невідомі векторні коефіцієнти:

$$\boldsymbol{\omega}_o = \frac{1}{2h}(3\mathbf{q}_1 - \mathbf{q}_2); \quad \boldsymbol{\varepsilon} = \frac{1}{2h^2}(\mathbf{q}_2 - \mathbf{q}_1).$$

Введемо позначення:

$$\mathbf{A} = \frac{1}{2}(3\mathbf{q}_1 - \mathbf{q}_2); \quad \mathbf{B} = \frac{1}{2}(\mathbf{q}_2 - \mathbf{q}_1). \quad (5.108)$$

Тоді кутову швидкість можна подати в формі:

$$\boldsymbol{\omega}(\tau) = \frac{1}{h}(\mathbf{A} + 2\mathbf{B}\zeta). \quad (5.109)$$

Перше наближення

Вираз (5.68) з врахуванням (5.109) трансформується у такий:

$$\Delta P_{(1)}(\tau) = \int_0^\tau [\boldsymbol{\omega}(\xi) \times] d\xi = \int_0^\zeta [(\mathbf{A} \times) + 2(\mathbf{B} \times)\xi] d\xi = (\mathbf{A} \times)\zeta + (\mathbf{B} \times)\zeta^2.$$

По закінченні кроку інтегрування (при $\zeta = 2$) величина матриці набуде значення:

$$P_{(1)}(2h) = E + [(3\mathbf{q}_1 - \mathbf{q}_2) \times] + 2[(\mathbf{q}_2 - \mathbf{q}_1) \times] = E + [(\mathbf{q}_1 + \mathbf{q}_2) \times]. \quad (5.110)$$

Друге наближення

Вираз (5.69) перетворюється у такий:

$$\begin{aligned}\Delta P_{(2)}(\tau) &= \int_0^\tau \Delta P_{(1)}(\xi) \cdot [\omega(\xi) \times] d\xi = \int_0^\tau [(\mathbf{A} \times) \zeta + (\mathbf{B} \times) \zeta^2] \cdot [(\mathbf{A} \times) + 2(\mathbf{B} \times) \zeta] d\zeta = \\ &= \frac{1}{2} (\mathbf{A} \times)^2 \zeta^2 + \frac{1}{3} [2(\mathbf{A} \times)(\mathbf{B} \times) + (\mathbf{B} \times)(\mathbf{A} \times)] \zeta^3 + \frac{1}{2} (\mathbf{B} \times)^2 \zeta^4.\end{aligned}\quad (5.111)$$

Запишемо (5.111) у спрощеному вигляді:

$$\Delta P_{(2)}(\tau) = \mathbf{K}_{22} \zeta^2 + \mathbf{K}_{23} \zeta^3 + \mathbf{K}_{24} \zeta^4, \quad (5.112)$$

де позначено

$$\mathbf{K}_{22} = \frac{1}{2} (\mathbf{A} \times)^2; \quad \mathbf{K}_{23} = \frac{1}{3} [2(\mathbf{A} \times)(\mathbf{B} \times) + (\mathbf{B} \times)(\mathbf{A} \times)]; \quad \mathbf{K}_{24} = \frac{1}{2} (\mathbf{B} \times)^2. \quad (5.113)$$

У кінці кроку інтегрування ($\zeta = 2$) матимемо:

$$\Delta P_{(2)}(2h) = 4\mathbf{K}_{22} + 8\mathbf{K}_{23} + 16\mathbf{K}_{24}.$$

Тому формулу двокрокового алгоритму в другому наближенні можна записати так:

$$\boxed{P_{(2)}(2h) = E + [(\mathbf{q}_1 + \mathbf{q}_2) \times] + 4\mathbf{K}_{22} + 8\mathbf{K}_{23} + 16\mathbf{K}_{24}}. \quad (5.144)$$

Третє наближення

З виразу (5.70) випливає:

$$\begin{aligned}\Delta P_{(3)}(\tau) &= \int_0^\tau \Delta P_{(2)}(\xi) \cdot [\omega(\xi) \times] d\xi = \int_0^\tau \{ \mathbf{K}_{22} \zeta^2 + \mathbf{K}_{23} \zeta^3 + \mathbf{K}_{24} \zeta^4 \} \times \\ &\times [(\mathbf{A} \times) + 2(\mathbf{B} \times) \zeta] d\zeta = \frac{1}{3} \mathbf{K}_{22} (\mathbf{A} \times) \zeta^3 + \frac{1}{4} [2\mathbf{K}_{22} (\mathbf{B} \times) + \mathbf{K}_{23} (\mathbf{A} \times)] \zeta^4 + \\ &+ \frac{1}{5} [2\mathbf{K}_{23} (\mathbf{B} \times) + \mathbf{K}_{24} (\mathbf{A} \times)] \zeta^5 + \frac{1}{6} \mathbf{K}_{24} (\mathbf{B} \times) \zeta^6 = \\ &= \mathbf{K}_{33} \zeta^3 + \mathbf{K}_{34} \zeta^4 + \mathbf{K}_{35} \zeta^5 + \mathbf{K}_{36} \zeta^6,\end{aligned}\quad (5.115)$$

де

$$\begin{aligned}\boxed{\mathbf{K}_{33} = \frac{1}{3} \mathbf{K}_{22} (\mathbf{A} \times)}; \quad & \boxed{\mathbf{K}_{34} = \frac{1}{4} [2\mathbf{K}_{22} (\mathbf{B} \times) + \mathbf{K}_{23} (\mathbf{A} \times)]}; \\ \boxed{\mathbf{K}_{35} = \frac{1}{5} [2\mathbf{K}_{23} (\mathbf{B} \times) + \mathbf{K}_{24} (\mathbf{A} \times)]}; \quad & \boxed{\mathbf{K}_{36} = \frac{1}{6} \mathbf{K}_{24} (\mathbf{B} \times)}.\end{aligned}$$

У кінці інтервалу інтегрування ($\zeta = 2$) значення матриці-приросту стане таким:

$$\boxed{\Delta P_{(3)}(2h) = 8\mathbf{K}_{33} + 16\mathbf{K}_{34} + 32\mathbf{K}_{35} + 64\mathbf{K}_{36}}. \quad (5.116)$$

Остаточно формулу двокрокового алгоритму у третьому наближенні отримаємо, враховуючи додатковий доданок (5.116) до формули (5.114) другого наближення.

Четверте наближення

З виразу (5.71) виходить:

$$\begin{aligned}
 \Delta P_{(4)}(\tau) &= \int_0^{\tau} \Delta P_{(3)}(\xi) \cdot [\omega(\xi) \times] d\xi = \\
 &= \int_0^{\xi} \{ \mathbf{K}_{33} \xi^3 + \mathbf{K}_{34} \xi^4 + \mathbf{K}_{35} \xi^5 + \mathbf{K}_{36} \xi^6 \} \times [(\mathbf{A} \times) + 2(\mathbf{B} \times) \xi] d\xi = \\
 &= \frac{1}{4} \mathbf{K}_{33} (\mathbf{A} \times) \xi^4 + \frac{1}{5} [2\mathbf{K}_{33} (\mathbf{B} \times) + \mathbf{K}_{34} (\mathbf{A} \times)] \xi^5 + \\
 &+ \frac{1}{6} [2\mathbf{K}_{34} (\mathbf{B} \times) + \mathbf{K}_{35} (\mathbf{A} \times)] \xi^6 + \frac{1}{7} [2\mathbf{K}_{35} (\mathbf{B} \times) + \mathbf{K}_{36} (\mathbf{A} \times)] \xi^7 + \frac{1}{4} \mathbf{K}_{36} (\mathbf{B} \times) \xi^8 = \\
 &= \mathbf{K}_{44} \xi^4 + \mathbf{K}_{45} \xi^5 + \mathbf{K}_{46} \xi^6 + \mathbf{K}_{47} \xi^7 + \mathbf{K}_{48} \xi^8.
 \end{aligned} \tag{5.117}$$

де

$$\begin{aligned}
 \mathbf{K}_{44} &= \frac{1}{4} \mathbf{K}_{33} (\mathbf{A} \times); & \mathbf{K}_{45} &= \frac{1}{5} [2\mathbf{K}_{33} (\mathbf{B} \times) + \mathbf{K}_{34} (\mathbf{A} \times)]; \\
 \mathbf{K}_{46} &= \frac{1}{6} [2\mathbf{K}_{34} (\mathbf{B} \times) + \mathbf{K}_{35} (\mathbf{A} \times)]; & \mathbf{K}_{47} &= \frac{1}{7} [2\mathbf{K}_{35} (\mathbf{B} \times) + \mathbf{K}_{36} (\mathbf{A} \times)]; \\
 & & \mathbf{K}_{48} &= \frac{1}{4} \mathbf{K}_{36} (\mathbf{B} \times).
 \end{aligned}$$

На кінці кроку інтегрування ($\xi = 2$) значення матриці (5.117) стане таким:

$$\Delta P_{(4)}(2h) = 16\mathbf{K}_{44} + 32\mathbf{K}_{45} + 64\mathbf{K}_{46} + 128\mathbf{K}_{47} + 256\mathbf{K}_{48}. \tag{5.118}$$

Формула двокрокового алгоритму у четвертому наближенні виходить доповненням додатком (5.118) формули (5.116) третього наближення.

Трикрокові алгоритми

При трикроковому алгоритмі на кожному кроці інтегрування є відомими три значення вектора приросту квазікоординат, і тому кутову швидкість слід шукати у вигляді квадратного поліному:

$$\omega(\tau) = \omega_o + 2\varepsilon\tau + 3a\tau^2.$$

Обчисливши прирости квазікоординат на кожному з трьох кроків опитування, дійдемо до системи трьох рівнянь:

$$\begin{aligned} \mathbf{q}_1 &= \int_0^h (\omega_o + 2\varepsilon\xi + 3a\xi^2) d\xi = \omega_o h + \varepsilon h^2 + ah^3; \\ \mathbf{q}_2 &= \int_h^{2h} (\omega_o + 2\varepsilon\xi + 3a\xi^2) d\xi = 2\omega_o h + 4\varepsilon h^2 + 8ah^3 - (\omega_o h + \varepsilon h^2 + ah^3) = \\ &= \omega_o h + 3\varepsilon h^2 + 7ah^3 \\ \mathbf{q}_3 &= \int_{2h}^{3h} (\omega_o + 2\varepsilon\xi + 3a\xi^2) d\xi = 3\omega_o h + 9\varepsilon h^2 + 27ah^3 - (2\omega_o h + 4\varepsilon h^2 + 8ah^3) = \\ &= \omega_o h + 5\varepsilon h^2 + 19ah^3 \end{aligned}$$

Розв'язок цієї системи виглядає у такий спосіб:

$$\omega_o h = \frac{1}{6}(1\mathbf{q}_1 - 7\mathbf{q}_2 + 2\mathbf{q}_3); \quad \varepsilon h^2 = \frac{1}{2}(-2\mathbf{q}_1 + 3\mathbf{q}_2 - \mathbf{q}_3); \quad ah^3 = \frac{1}{6}(\mathbf{q}_1 - 2\mathbf{q}_2 + \mathbf{q}_3). \quad (5.119)$$

Введемо позначення:

$$\begin{aligned} \mathbf{A} &= \frac{1}{6}(1\mathbf{q}_1 - 7\mathbf{q}_2 + 2\mathbf{q}_3); & \mathbf{B} &= \frac{1}{2}(-2\mathbf{q}_1 + 3\mathbf{q}_2 - \mathbf{q}_3); \\ \mathbf{C} &= \frac{1}{6}(\mathbf{q}_1 - 2\mathbf{q}_2 + \mathbf{q}_3). \end{aligned} \quad (5.120)$$

Тоді вирази (5.59) можна записати у вигляді:

$$\omega_o = \frac{1}{h} \mathbf{A}; \quad \varepsilon = \frac{1}{h^2} \mathbf{B}; \quad a = \frac{1}{h^3} \mathbf{C},$$

а тому функція кутової швидкості набуде вигляду:

$$\omega(\tau) = \frac{1}{h} (\mathbf{A} + 2\mathbf{B}\zeta + 3\mathbf{C}\zeta^2). \quad (5.121)$$

Перше наближення

Вираз (5.68) з врахуванням (5.121) перетвориться на такий:

$$\Delta P_{(1)}(\tau) = \int_0^\tau [\omega(\xi) \times] d\xi = \int_0^\zeta [(\mathbf{A} + 2\mathbf{B}\zeta + 3\mathbf{C}\zeta^2) \times] d\zeta = (\mathbf{A} \times) \zeta + (\mathbf{B} \times) \zeta^2 + (\mathbf{C} \times) \zeta^3.$$

При $\zeta = 3$ величина матриці-приросту набуде значення:

$$\Delta P_{(1)}(3h) = 3(\mathbf{A} \times) + 9(\mathbf{B} \times) + 27(\mathbf{C} \times) = (\mathbf{q}_1 \times) + (\mathbf{q}_2 \times) + (\mathbf{q}_3 \times). \quad (5.122)$$

Друге наближення

З виразу (5.69) випливає:

$$\begin{aligned}
\Delta P_{(2)}(\tau) &= \int_0^\tau \Delta P_{(1)}(\xi) \cdot [\omega(\xi) \times] d\xi = \\
&= \int_0^\xi [(\mathbf{A} \times) \zeta + (\mathbf{B} \times) \zeta^2 + (\mathbf{C} \times) \zeta^3] \cdot [(\mathbf{A} \times) + 2(\mathbf{B} \times) \zeta + 3(\mathbf{C} \times) \zeta^2] d\zeta = \\
&= \zeta^2 \left\{ \frac{1}{2} (\mathbf{A} \times)^2 + \frac{1}{3} [2(\mathbf{A} \times)(\mathbf{B} \times) + (\mathbf{B} \times)(\mathbf{A} \times)] \zeta + \right. \\
&\quad + \frac{1}{4} [3(\mathbf{A} \times)(\mathbf{C} \times) + 2(\mathbf{B} \times)^2 + (\mathbf{C} \times)(\mathbf{A} \times)] \zeta^2 + \\
&\quad \left. + \frac{1}{5} [3(\mathbf{B} \times)(\mathbf{C} \times) + 2(\mathbf{C} \times)(\mathbf{B} \times)] \zeta^3 + \frac{1}{2} (\mathbf{C} \times)^2 \zeta^4 \right\} = \\
&= \mathbf{K}_{22} \zeta^2 + \mathbf{K}_{23} \zeta^3 + \mathbf{K}_{24} \zeta^4 + \mathbf{K}_{25} \zeta^5 + \mathbf{K}_{26} \zeta^6.
\end{aligned} \tag{5.123}$$

де

$$\begin{aligned}
\boxed{\mathbf{K}_{22} = \frac{1}{2} (\mathbf{A} \times)^2}; \quad & \boxed{\mathbf{K}_{23} = \frac{1}{3} [2(\mathbf{A} \times)(\mathbf{B} \times) + (\mathbf{B} \times)(\mathbf{A} \times)]}; \\
\boxed{\mathbf{K}_{24} = \frac{1}{4} [3(\mathbf{A} \times)(\mathbf{C} \times) + 2(\mathbf{B} \times)^2 + (\mathbf{C} \times)(\mathbf{A} \times)]}; \\
\boxed{\mathbf{K}_{25} = \frac{1}{5} [3(\mathbf{B} \times)(\mathbf{C} \times) + 2(\mathbf{C} \times)(\mathbf{B} \times)]}; \quad & \boxed{\mathbf{K}_{26} = \frac{1}{2} (\mathbf{C} \times)^2}.
\end{aligned} \tag{5.124}$$

Наприкінці кроку інтегрування ($\zeta = 3$) матриця (5.123) стане такою:

$$\Delta P_{(2)}(3h) = 9\mathbf{K}_{22} + 27\mathbf{K}_{23} + 81\mathbf{K}_{24} + 234\mathbf{K}_{25} + 729\mathbf{K}_{26}.$$

Тому формула для трикрокового алгоритму у другому наближенні буде мати вигляд:

$$\boxed{P_{(2)}(3h) = E + [(\mathbf{q}_1 + \mathbf{q}_2) \times] + 9\mathbf{K}_{22} + 27\mathbf{K}_{23} + 81\mathbf{K}_{24} + 243\mathbf{K}_{25} + 729\mathbf{K}_{26}}. \tag{5.125}$$

Третє наближення

З співвідношення (5.70) отримаємо добавку третього наближення:

$$\begin{aligned}
\Delta P_{(3)}(\tau) &= \int_0^\tau \Delta P_{(2)}(\xi) \cdot [\omega(\xi) \times] d\xi = \\
&= \int_0^\xi [\mathbf{K}_{22} \zeta^2 + \mathbf{K}_{23} \zeta^3 + \mathbf{K}_{24} \zeta^4 + \mathbf{K}_{25} \zeta^5 + \mathbf{K}_{26} \zeta^6] \cdot [(\mathbf{A} \times) + 2(\mathbf{B} \times) \zeta + 3(\mathbf{C} \times) \zeta^2] d\zeta.
\end{aligned} \tag{5.126}$$

Після здійснення операцій множення і інтегрування отримаємо вираз:

$$\Delta P_{(3)}(\tau) = \sum_{i=3}^9 \mathbf{K}_{3i} \zeta^i, \quad (5.127)$$

де значення векторних коефіцієнтів обчислюються за формулами

$$\begin{aligned} \mathbf{K}_{33} &= \frac{1}{3} \mathbf{K}_{22}(\mathbf{A} \times); & \mathbf{K}_{34} &= \frac{1}{4} [2\mathbf{K}_{22}(\mathbf{B} \times) + \mathbf{K}_{23}(\mathbf{A} \times)]; \\ \mathbf{K}_{35} &= \frac{1}{5} [3\mathbf{K}_{22}(\mathbf{C} \times) + 2\mathbf{K}_{23}(\mathbf{B} \times) + \mathbf{K}_{24}(\mathbf{A} \times)]; \\ \mathbf{K}_{36} &= \frac{1}{6} [3\mathbf{K}_{23}(\mathbf{C} \times) + 2\mathbf{K}_{24}(\mathbf{B} \times) + \mathbf{K}_{25}(\mathbf{A} \times)]; \\ \mathbf{K}_{37} &= \frac{1}{7} [3\mathbf{K}_{24}(\mathbf{C} \times) + 2\mathbf{K}_{25}(\mathbf{B} \times) + \mathbf{K}_{26}(\mathbf{A} \times)]; \\ \mathbf{K}_{38} &= \frac{1}{8} [3\mathbf{K}_{25}(\mathbf{C} \times) + 2\mathbf{K}_{26}(\mathbf{B} \times)]; & \mathbf{K}_{39} &= \frac{1}{3} \mathbf{K}_{26}(\mathbf{C} \times). \end{aligned} \quad (5.128)$$

При $\zeta = 3$ звідси виходить добавка, яку слід додати до виразу (5.125), щоб дійти до формули трикрокового алгоритму у третьому наближенні:

$$\Delta P_{(3)}(3h) = 27\mathbf{K}_{33} + 81\mathbf{K}_{34} + 243\mathbf{K}_{35} + 729\mathbf{K}_{36} + 2181\mathbf{K}_{37} + 6543\mathbf{K}_{38} + 19629\mathbf{K}_{39}. \quad (5.129)$$

Четверте наближення

Добавку четвертого наближення отримаємо зі співвідношення (5.71):

$$\begin{aligned} \Delta P_{(4)}(\tau) &= \int_0^{\tau} \Delta P_{(3)}(\xi) \cdot [\boldsymbol{\omega}(\xi) \times] d\xi = \\ &= \int_0^{\zeta} \left[\sum_{i=3}^9 \mathbf{K}_{3i} \zeta^i \right] [(\mathbf{A} \times) + 2(\mathbf{B} \times)\zeta + 3(\mathbf{C} \times)\zeta^2] d\zeta = \sum_{i=4}^{12} \mathbf{K}_{4i} \zeta^i. \end{aligned} \quad (5.130)$$

Здійснивши множення та інтегрування, знайдемо вирази коефіцієнтів \mathbf{K}_{4i} :

$$\begin{aligned} \mathbf{K}_{44} &= \frac{1}{4} \mathbf{K}_{33}(\mathbf{A} \times); & \mathbf{K}_{45} &= \frac{1}{5} [2\mathbf{K}_{33}(\mathbf{B} \times) + \mathbf{K}_{34}(\mathbf{A} \times)]; \\ \mathbf{K}_{46} &= \frac{1}{6} [3\mathbf{K}_{33}(\mathbf{C} \times) + 2\mathbf{K}_{34}(\mathbf{B} \times) + \mathbf{K}_{35}(\mathbf{A} \times)]; \\ \mathbf{K}_{47} &= \frac{1}{7} [3\mathbf{K}_{34}(\mathbf{C} \times) + 2\mathbf{K}_{35}(\mathbf{B} \times) + \mathbf{K}_{36}(\mathbf{A} \times)]; \\ \mathbf{K}_{48} &= \frac{1}{8} [3\mathbf{K}_{35}(\mathbf{C} \times) + 2\mathbf{K}_{36}(\mathbf{B} \times) + \mathbf{K}_{37}(\mathbf{A} \times)]; \\ \mathbf{K}_{49} &= \frac{1}{9} [3\mathbf{K}_{36}(\mathbf{C} \times) + 2\mathbf{K}_{37}(\mathbf{B} \times) + \mathbf{K}_{38}(\mathbf{A} \times)]; \end{aligned} \quad (5.131)$$

$$\mathbf{K}_{410} = \frac{1}{10} [3\mathbf{K}_{37}(\mathbf{C} \times) + 2\mathbf{K}_{38}(\mathbf{B} \times) + \mathbf{K}_{39}(\mathbf{A} \times)];$$

$$\mathbf{K}_{411} = \frac{1}{11} [3\mathbf{K}_{38}(\mathbf{C} \times) + 2\mathbf{K}_{39}(\mathbf{B} \times)]; \quad \mathbf{K}_{412} = \frac{1}{4} \mathbf{K}_{39}(\mathbf{C} \times).$$

Формулу трикрокового алгоритму у четвертому наближенні отримаємо, обчисливши значення матриці (5.130) при $\zeta = 3$ і додавши результат до виразів (5.129) і (5.125).

Чотирикрокові алгоритми

У випадку чотирикрокових алгоритмів кутова швидкість може бути поданою кубічним поліномом:

$$\omega(\tau) = \omega_o + 2\varepsilon\tau + 3a\tau^2 + 4b\tau^3.$$

З цього знайдемо прирости квазікоординат :

$$\mathbf{q}_1 = \int_0^h (\omega_o + 2\varepsilon\xi + 3a\xi^2 + 4b\xi^3) d\xi = \omega_o h + \varepsilon h^2 + ah^3 + bh^4;$$

$$\mathbf{q}_2 = \int_h^{2h} (\omega_o + 2\varepsilon\xi + 3a\xi^2 + 4b\xi^3) d\xi = \omega_o h + 3\varepsilon h^2 + 7ah^3 + 15bh^4;$$

$$\mathbf{q}_3 = \int_{2h}^{3h} (\omega_o + 2\varepsilon\xi + 3a\xi^2 + 4b\xi^3) d\xi = \omega_o h + 5\varepsilon h^2 + 19ah^3 + 65bh^4;$$

$$\mathbf{q}_4 = \int_{3h}^{4h} (\omega_o + 2\varepsilon\xi + 3a\xi^2 + 4b\xi^3) d\xi = \omega_o h + 7\varepsilon h^2 + 37ah^3 + 175bh^4.$$

Розв'язавши цю систему рівнянь відносно невідомих векторних коефіцієнтів кутову швидкість можна подати у вигляді

$$\omega(\tau) = \frac{1}{h} (\mathbf{A} + 2\mathbf{B}\zeta + 3\mathbf{C}\zeta^2 + 4\mathbf{D}\zeta^3), \quad (5.132)$$

де

$$\begin{aligned} \mathbf{A} &= \frac{1}{12} (25\mathbf{q}_1 - 23\mathbf{q}_2 + 13\mathbf{q}_3 - 3\mathbf{q}_4); & \mathbf{B} &= \frac{1}{24} (-35\mathbf{q}_1 + 69\mathbf{q}_2 - 45\mathbf{q}_3 + 11\mathbf{q}_4); \\ \mathbf{C} &= \frac{1}{12} (5\mathbf{q}_1 - 13\mathbf{q}_2 + 11\mathbf{q}_3 - 3\mathbf{q}_4); & \mathbf{D} &= \frac{1}{24} (-\mathbf{q}_1 + 3\mathbf{q}_2 - 3\mathbf{q}_3 + \mathbf{q}_4), \end{aligned} \quad (5.133)$$

Перше наближення

У відповідності до виразу (5.68) маємо:

$$\begin{aligned}\Delta P_{(1)}(\tau) &= \int_0^{\tau} [\boldsymbol{\omega}(\xi) \times] d\xi = \int_0^{\tau} [(\mathbf{A} \times) + 2(\mathbf{B} \times)\xi + 3(\mathbf{C} \times)\xi^2 + 4(\mathbf{D} \times)\xi^3] d\xi = \\ &= [(\mathbf{A} \times)\xi + (\mathbf{B} \times)\xi^2 + (\mathbf{C} \times)\xi^3 + (\mathbf{D} \times)\xi^4]\end{aligned}\quad (5.134)$$

При $\zeta = 4$ величина матриці P прийме значення:

$$\begin{aligned}P_{(1)}(4h) &= E + 4[(\mathbf{A} \times) + 4(\mathbf{B} \times) + 16(\mathbf{C} \times) + 64(\mathbf{D} \times)] = \\ &= E + 4[(\mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3 + \mathbf{q}_4) \times]\end{aligned}\quad (5.135)$$

Друге наближення

З виразу (5.69) виходить:

$$\begin{aligned}\Delta P_{(2)}(\tau) &= \int_0^{\tau} \Delta P_{(1)}(\xi) \cdot [\boldsymbol{\omega}(\xi) \times] d\xi = \\ &= \int_0^{\zeta} [(\mathbf{A} \times)\xi + (\mathbf{B} \times)\xi^2 + (\mathbf{C} \times)\xi^3 + (\mathbf{D} \times)\xi^4] [(\mathbf{A} \times) + 2(\mathbf{B} \times)\xi + 3(\mathbf{C} \times)\xi^2 + 4(\mathbf{D} \times)\xi^3] d\xi = \\ &= \mathbf{K}_{22}\zeta^2 + \mathbf{K}_{23}\zeta^3 + \mathbf{K}_{24}\zeta^4 + \mathbf{K}_{25}\zeta^5 + \mathbf{K}_{26}\zeta^6 + \mathbf{K}_{27}\zeta^7 + \mathbf{K}_{28}\zeta^8 = \sum_{k=2}^8 \mathbf{K}_{2k}\zeta^k,\end{aligned}\quad (5.136)$$

де

$$\begin{aligned}\mathbf{K}_{22} &= \frac{1}{2}(\mathbf{A} \times)^2; & \mathbf{K}_{23} &= \frac{1}{3}[2(\mathbf{A} \times)(\mathbf{B} \times) + (\mathbf{B} \times)(\mathbf{A} \times)]; \\ \mathbf{K}_{24} &= \frac{1}{4}[3(\mathbf{A} \times)(\mathbf{C} \times) + 2(\mathbf{B} \times)^2 + (\mathbf{C} \times)(\mathbf{A} \times)]; \\ \mathbf{K}_{25} &= \frac{1}{5}[4(\mathbf{A} \times)(\mathbf{D} \times) + 3(\mathbf{B} \times)(\mathbf{C} \times) + 2(\mathbf{C} \times)(\mathbf{B} \times) + (\mathbf{D} \times)(\mathbf{A} \times)]; \\ \mathbf{K}_{26} &= \frac{1}{6}[4(\mathbf{B} \times)(\mathbf{D} \times) + 3(\mathbf{C} \times)^2 + 2(\mathbf{D} \times)(\mathbf{B} \times)]; \\ \mathbf{K}_{27} &= \frac{1}{7}[4(\mathbf{C} \times)(\mathbf{D} \times) + 3(\mathbf{D} \times)(\mathbf{C} \times)]; & \mathbf{K}_{28} &= \frac{1}{2}(\mathbf{D} \times)^2.\end{aligned}$$

Якщо покласти $\zeta = 4$, будемо мати:

$$\Delta P_{(2)}(4h) = \sum_{k=2}^8 4^k \mathbf{K}_{2k}.\quad (5.137)$$

Формула для чотирикрокового алгоритму у другому наближенні має вигляд:

$$P_{(2)}(4h) = E + [(\mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3 + \mathbf{q}_4) \times] + \sum_{k=2}^8 4^k \mathbf{K}_{2k}. \quad (5.138)$$

Третє наближення

З співвідношення (5.70) виходить добавка у третьому наближенні:

$$\begin{aligned} \Delta P_{(3)}(\tau) &= \int_0^\tau \Delta P_{(2)}(\xi) \cdot [\boldsymbol{\omega}(\xi) \times] d\xi = \int_0^\xi [\mathbf{K}_{22}\zeta^2 + \mathbf{K}_{23}\zeta^3 + \mathbf{K}_{24}\zeta^4 + \mathbf{K}_{25}\zeta^5 + \mathbf{K}_{26}\zeta^6] \times \\ &\quad \times [(\mathbf{A} \times) + 2(\mathbf{B} \times)\zeta + 3(\mathbf{C} \times)\zeta^2 + 4(\mathbf{D} \times)\zeta^3] d\zeta = \\ &= \mathbf{K}_{33}\zeta^3 + \mathbf{K}_{34}\zeta^4 + \mathbf{K}_{35}\zeta^5 + \mathbf{K}_{36}\zeta^6 + \mathbf{K}_{37}\zeta^7 + \mathbf{K}_{38}\zeta^8 + \mathbf{K}_{39}\zeta^9 + \\ &\quad + \mathbf{K}_{310}\zeta^{10} + \mathbf{K}_{311}\zeta^{11} + \mathbf{K}_{312}\zeta^{12} = \sum_{k=3}^{12} \mathbf{K}_{3k} \zeta^k, \end{aligned} \quad (5.139)$$

де

$$\begin{aligned} \mathbf{K}_{33} &= \frac{1}{3} \mathbf{K}_{22}(\mathbf{A} \times); & \mathbf{K}_{34} &= \frac{1}{4} [2\mathbf{K}_{22}(\mathbf{B} \times) + \mathbf{K}_{23}(\mathbf{A} \times)]; \\ \mathbf{K}_{35} &= \frac{1}{5} [3\mathbf{K}_{22}(\mathbf{C} \times) + 2\mathbf{K}_{23}(\mathbf{B} \times) + \mathbf{K}_{24}(\mathbf{A} \times)]; \\ \mathbf{K}_{36} &= \frac{1}{6} [4\mathbf{K}_{22}(\mathbf{D} \times) + 3\mathbf{K}_{23}(\mathbf{C} \times) + 2\mathbf{K}_{24}(\mathbf{B} \times) + \mathbf{K}_{25}(\mathbf{A} \times)]; \\ \mathbf{K}_{37} &= \frac{1}{7} [4\mathbf{K}_{23}(\mathbf{D} \times) + 3\mathbf{K}_{24}(\mathbf{C} \times) + 2\mathbf{K}_{25}(\mathbf{B} \times) + \mathbf{K}_{26}(\mathbf{A} \times)]; \\ \mathbf{K}_{38} &= \frac{1}{8} [4\mathbf{K}_{24}(\mathbf{D} \times) + 3\mathbf{K}_{25}(\mathbf{C} \times) + 2\mathbf{K}_{26}(\mathbf{B} \times) + \mathbf{K}_{27}(\mathbf{A} \times)]; \\ \mathbf{K}_{39} &= \frac{1}{9} [4\mathbf{K}_{25}(\mathbf{D} \times) + 3\mathbf{K}_{26}(\mathbf{C} \times) + 2\mathbf{K}_{27}(\mathbf{B} \times) + \mathbf{K}_{28}(\mathbf{A} \times)]; \\ \mathbf{K}_{310} &= \frac{1}{10} [4\mathbf{K}_{26}(\mathbf{D} \times) + 3\mathbf{K}_{27}(\mathbf{C} \times) + 2\mathbf{K}_{28}(\mathbf{B} \times)]; \\ \mathbf{K}_{311} &= \frac{1}{11} [4\mathbf{K}_{27}(\mathbf{D} \times) + 3\mathbf{K}_{28}(\mathbf{C} \times)]; & \mathbf{K}_{312} &= \frac{1}{3} 4\mathbf{K}_{28}(\mathbf{D} \times). \end{aligned} \quad (5.140)$$

У кінці кроку інтегрування ($\zeta = 4$) отримаємо добавку:

$$\Delta P_{(3)}(4h) = \left[\sum_{k=3}^{12} 4^k \mathbf{K}_{3k} \right]. \quad (5.141)$$

Формула для чотирьокрокового алгоритму у третьому наближенні має вигляд:

$$P_{(3)}(4h) = E + [(\mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3 + \mathbf{q}_4) \times] + \sum_{i=2}^8 4^i \mathbf{K}_{2i} + \sum_{k=3}^{12} 4^k \mathbf{K}_{3k}. \quad (5.142)$$

Четверте наближення

Додавку четвертого наближення отримаємо з співвідношення (5.71):

$$\begin{aligned}\Delta P_{(4)}(\tau) &= \int_0^{\tau} \Delta P_{(3)}(\xi) \cdot [\omega(\xi) \times] d\xi = \\ &= \int_0^{\xi} \left[\sum_{k=3}^{12} \mathbf{K}_{3k} \zeta^k \right] [(\mathbf{A} \times) + 2(\mathbf{B} \times)\zeta + 3(\mathbf{C} \times)\zeta^2 + 4(\mathbf{D} \times)\zeta^3] d\zeta.\end{aligned}\quad (5.143)$$

Здійснюючи операції множення і інтегрування, дійдемо виразів матричних коефіцієнтів \mathbf{K}_{3k} при кожному ζ^k :

$$\begin{aligned}\mathbf{K}_{44} &= \frac{1}{4} \mathbf{K}_{33}(\mathbf{A} \times); & \mathbf{K}_{45} &= \frac{1}{5} [2\mathbf{K}_{33}(\mathbf{B} \times) + \mathbf{K}_{34}(\mathbf{A} \times)]; \\ \mathbf{K}_{46} &= \frac{1}{6} [3\mathbf{K}_{33}(\mathbf{C} \times) + 2\mathbf{K}_{34}(\mathbf{B} \times) + \mathbf{K}_{35}(\mathbf{A} \times)]; \\ \mathbf{K}_{47} &= \frac{1}{7} [4\mathbf{K}_{33}(\mathbf{D} \times) + 3\mathbf{K}_{34}(\mathbf{C} \times) + 2\mathbf{K}_{35}(\mathbf{B} \times) + \mathbf{K}_{36}(\mathbf{A} \times)]; \\ \mathbf{K}_{48} &= \frac{1}{8} [4\mathbf{K}_{34}(\mathbf{D} \times) + 3\mathbf{K}_{35}(\mathbf{C} \times) + 2\mathbf{K}_{36}(\mathbf{B} \times) + \mathbf{K}_{37}(\mathbf{A} \times)]; \\ \mathbf{K}_{49} &= \frac{1}{9} [4\mathbf{K}_{35}(\mathbf{D} \times) + 3\mathbf{K}_{36}(\mathbf{C} \times) + 2\mathbf{K}_{37}(\mathbf{B} \times) + \mathbf{K}_{38}(\mathbf{A} \times)]; \\ \mathbf{K}_{410} &= \frac{1}{10} [4\mathbf{K}_{36}(\mathbf{D} \times) + 3\mathbf{K}_{37}(\mathbf{C} \times) + 2\mathbf{K}_{38}(\mathbf{B} \times) + \mathbf{K}_{39}(\mathbf{A} \times)]; \\ \mathbf{K}_{411} &= \frac{1}{11} [4\mathbf{K}_{37}(\mathbf{D} \times) + 3\mathbf{K}_{38}(\mathbf{C} \times) + 2\mathbf{K}_{39}(\mathbf{B} \times) + \mathbf{K}_{310}(\mathbf{A} \times)]; \\ \mathbf{K}_{412} &= \frac{1}{12} [4\mathbf{K}_{38}(\mathbf{D} \times) + 3\mathbf{K}_{39}(\mathbf{C} \times) + 2\mathbf{K}_{310}(\mathbf{B} \times) + \mathbf{K}_{311}(\mathbf{A} \times)]; \\ \mathbf{K}_{413} &= \frac{1}{13} [4\mathbf{K}_{39}(\mathbf{D} \times) + 3\mathbf{K}_{310}(\mathbf{C} \times) + 2\mathbf{K}_{311}(\mathbf{B} \times) + \mathbf{K}_{312}(\mathbf{A} \times)]; \\ \mathbf{K}_{414} &= \frac{1}{14} [4\mathbf{K}_{310}(\mathbf{D} \times) + 3\mathbf{K}_{311}(\mathbf{C} \times) + 2\mathbf{K}_{312}(\mathbf{B} \times)]; \\ \mathbf{K}_{415} &= \frac{1}{15} [4\mathbf{K}_{311}(\mathbf{D} \times) + 3\mathbf{K}_{312}(\mathbf{C} \times)]; & \mathbf{K}_{416} &= \frac{1}{4} \mathbf{K}_{312}(\mathbf{D} \times).\end{aligned}\quad (5.144)$$

Після чого замість (5.83) отримаємо:

$$\Delta P_{(4)}(\tau) = \sum_{k=4}^{16} \mathbf{K}_{4k} \zeta^k. \quad (5.145)$$

Додавши цю складову при $\zeta = 4$ до правої частини формули (5.42), дійдемо формули чотирикрокового алгоритму у четвертому наближенні:

$$P_{(4)}(4h) = E + [(q_1 + q_2 + q_3 + q_4) \times] + \sum_{i=2}^8 4^i K_{2i} + \sum_{j=3}^{12} 4^j K_{3j} + \sum_{k=4}^{16} 4^k K_{4k}. \quad (5.146)$$

Дослідження похибок алгоритмів

Результати модельного дослідження похибок алгоритмів, що спираються на вимірювання кутової швидкості подані у графічній формі на рисунках 5.7, 5.8, 5.9, та 5.10.

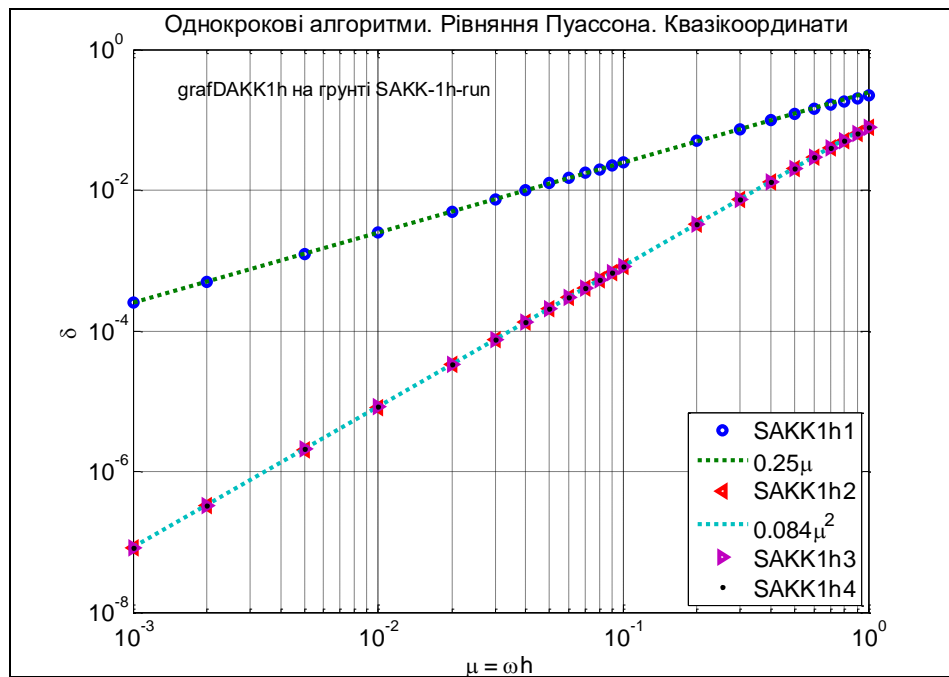


Рис. 5.7. Однокрокові алгоритми

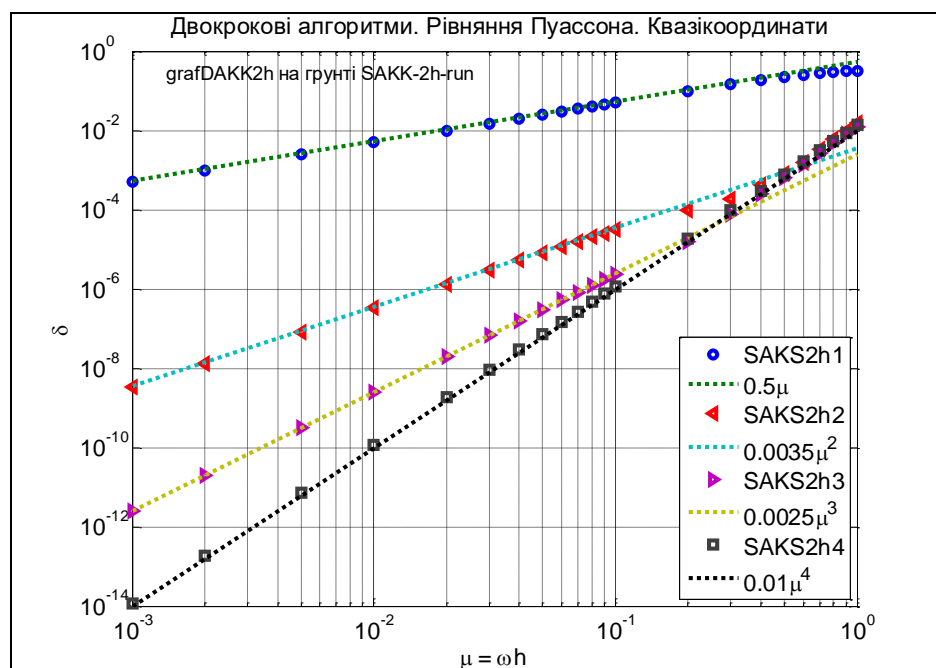


Рис. 5.8. Двокрокові алгоритми

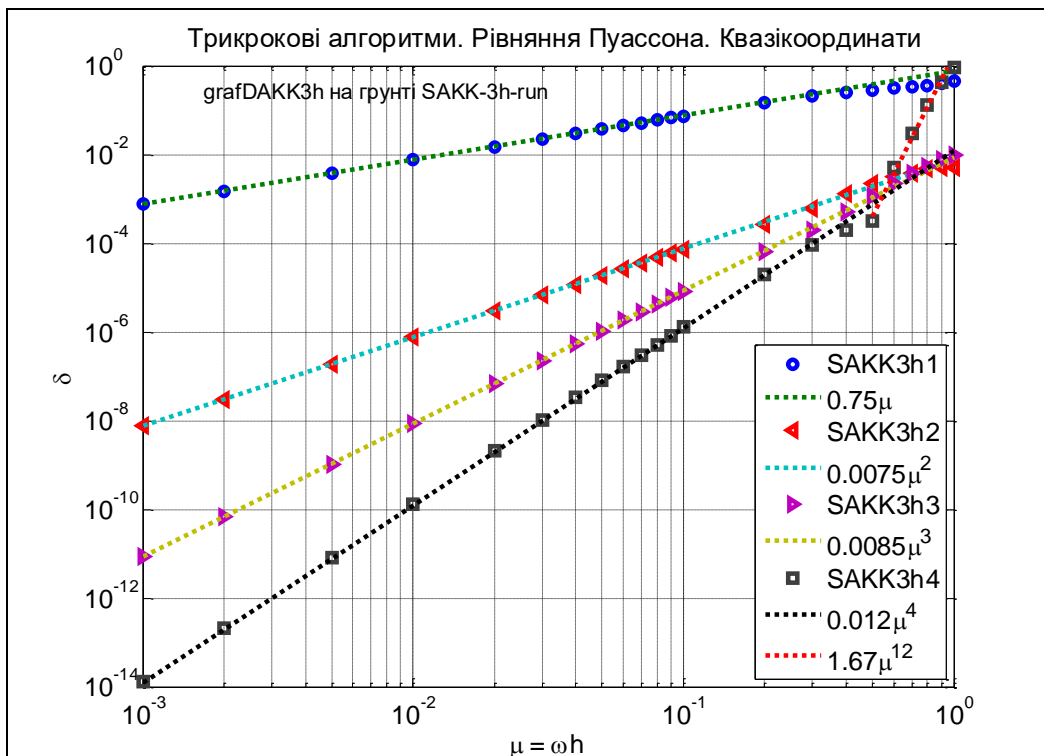


Рис. 5.9. Трикрокові алгоритми

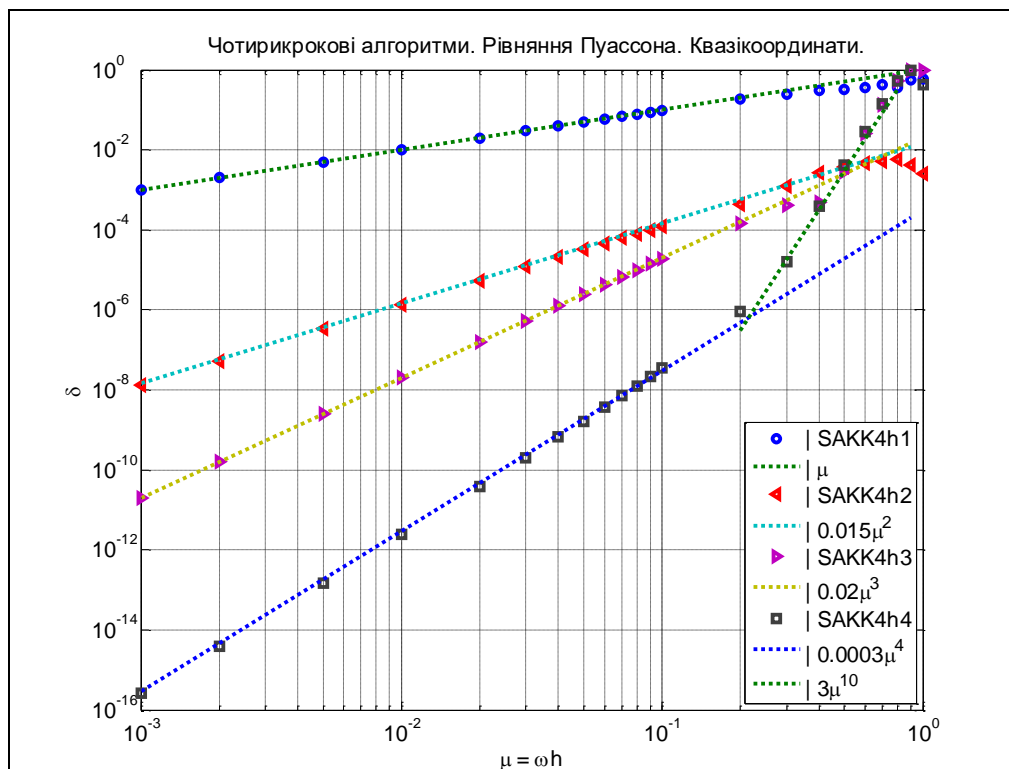


Рис. 5.10. Чотирикрокові алгоритми

Параметри точності досліджених алгоритмів подані у таблицях 5.7, 5.8, 5.9, 5.10.

Таблиця 5.7. Параметри точності однокрокових алгоритмів при вимірюванні квазікоординат

Алгоритм	Кроковість	Наближення	Порядок точності N	k
SAKK1h1	1	1	1	0,25
SAKK1h2	1	2	2	0,084
SAKK1h3	1	3	2	0,084
SAKK1h4	1	4	2	0,084

Таблиця 5.8. Параметри точності двокрокових алгоритмів при вимірюванні квазікоординат

Алгоритм	Кроковість	Наближення	Порядок точності N	k
SAKK2h1	2	1	1	0,5
SAKK2h2	2	2	2	0,033
SAKK2h3	2	3	3	0,025
SAKK2h4	2	4	4	0,011

Таблиця 5.9. Параметри точності трикрокових алгоритмів при вимірюванні квазікоординат

Алгоритм	Кроковість	Наближення	Порядок точності N	k
SAKK3h1	3	1	1	0,75
SAKK3h2	3	2	2	0,0075
SAKK3h3	3	3	3	0,0085
SAKK3h4	3	4	4	0,012

Таблиця 5.10. Параметри точності чотирьокрокових алгоритмів при вимірюванні квазікоординат

Алгоритм	Кроковість	Наближення	Порядок точності N	k
SAKK4h1	4	1	1	1
SAKK4h2	4	2	2	0,015
SAKK4h3	4	3	3	0,02
SAKK4h4	4	4	4	0,0003

Висновки

Порівнюючи ці результати з отриманими раніше у випадку застосування векторного рівняння Ейлера, можна дійти таких висновків:

6) практично однаковою виявляється точність алгоритмів

однокрокових $O1h$ (Ейлер) і $SAKK1h2$ (Пуассон);

двокрокових $O2h$ (Ейлер) і $SAKK2h4$ (Пуассон);

трикрокових $O3h$ (Ейлер) і $SAKK3h4$ (Пуассон);

чотирикрокових $O4h$ (Ейлер) і $SAKK4h4$ (Пуассон);

7) при цьому відповідні алгоритми на ґрунті рівняння Пуассона набагато складніше і потребують значно більшого комп'ютерного часу.

6. Синтез розгінних алгоритмів

Розглянемо застосування методу Пікара до *кватерніонного рівняння орієнтації*. Причому, на відміну від попереднього, займемося синтезом *розгінних* алгоритмів.

6.1. Побудова алгоритмів для кватерніонного рівняння орієнтації

6.1.1. Постановка задачі

За методом Пікара чисельний розв'язок диференціального кінематичного рівняння орієнтації у кватерніонах

$$\frac{d\Lambda(t)}{dt} = \frac{1}{2} \Lambda(t) \circ \omega(t)$$

зазвичай відшукується у вигляді [10]

$$\Lambda_{n+1} = \Lambda_n \circ \mathbf{N}_{n+1},$$

де $\Lambda = \lambda_0 + \mathbf{i}\lambda_x + \mathbf{j}\lambda_y + \mathbf{k}\lambda_z$ – кватерніон повороту об'єкта, $\omega(t) = 0 + \mathbf{i}\omega_x(t) + \mathbf{j}\omega_y(t) + \mathbf{k}\omega_z(t)$ – вектор-кватерніон з проекцій кутової швидкості об'єкта на осі, пов'язані з об'єктом, \circ – позначка кватерніонного множення; Λ_n – значення кватерніона повороту на n -му кроці інтегрування; Λ_{n+1} – шукане значення цього кватерніона на наступному $(n+1)$ -му кроці інтегрування, причому \mathbf{N}_{n+1} є розв'язком рівняння

$$\mathbf{N}(t) = 1 + \frac{1}{2} \int_{t_n}^t \mathbf{N}(\xi) \circ \omega(\xi) d\xi \quad (6.1)$$

у момент часу $t = t_{n+1} = t_n + h$.

Проміжне рівняння (6.1), у відповідності з методом Пікара, розв'язується послідовними наближеннями.

6.1.2. Побудова наближених розв'язків допоміжного рівняння

У нульовому наближенні приймемо $\mathbf{N}_{(0)}(t) = 1$. Підставивши це у праву частину (6.1), одержимо розв'язок рівняння у першому наближенні:

$$\mathbf{N}_{(1)}(t) = 1 + \frac{1}{2} \int_{t_n}^t \mathbf{N}_{(0)}(\xi) \circ \boldsymbol{\omega}(\xi) d\xi = 1 + \Delta \mathbf{N}_{(1)}(t).$$

$$\Delta \mathbf{N}_{(1)}(t) = \frac{1}{2} \int_{t_n}^t \boldsymbol{\omega}(\xi) d\xi$$

(6.2)

Друге наближення отримаємо шляхом підставлення результатів першого наближення у вираз (6.1):

$$\begin{aligned} \mathbf{N}_{(2)}(t) &= 1 + \frac{1}{2} \int_{t_n}^t \mathbf{N}_{(1)}(\xi) \circ \boldsymbol{\omega}(\xi) d\xi = 1 + \frac{1}{2} \int_{t_n}^t [1 + \Delta \mathbf{N}_{(1)}(\xi)] \circ \boldsymbol{\omega}(\xi) d\xi = \\ &= \mathbf{N}_{(1)}(t) + \Delta \mathbf{N}_{(2)}(t). \end{aligned}$$

$$\Delta \mathbf{N}_{(2)}(t) = \frac{1}{4} \int_{t_n}^t \left[\int_{t_n}^{\xi} \boldsymbol{\omega}(\eta) d\eta \right] \circ \boldsymbol{\omega}(\xi) d\xi$$

(6.3)

Використовуючи знайдене значення кватерніону $\mathbf{N}_{(2)}(t)$ у другому наближенні, відшукуємо третє наближення $\mathbf{N}_{(3)}(t)$

$$\begin{aligned} \mathbf{N}_{(3)}(t) &= 1 + \frac{1}{2} \int_{t_n}^t \mathbf{N}_{(2)}(\xi) \circ \boldsymbol{\omega}(\xi) d\xi = \\ &= 1 + \frac{1}{2} \int_{t_n}^t \boldsymbol{\omega}(\xi) d\xi + \frac{1}{4} \int_{t_n}^t \left(\int_{t_n}^{\xi} \boldsymbol{\omega}(\eta) d\eta \right) \circ \boldsymbol{\omega}(\xi) d\xi + \frac{1}{8} \int_{t_n}^t \left[\int_{t_n}^{\xi} \left(\int_{t_n}^{\eta} \boldsymbol{\omega}(\zeta) d\zeta \right) \circ \boldsymbol{\omega}(\eta) d\eta \right] \circ \boldsymbol{\omega}(\xi) d\xi = \\ &= \mathbf{N}_{(2)}(t) + \Delta \mathbf{N}_{(3)}(t). \end{aligned}$$

$$\Delta \mathbf{N}_{(3)}(t) = \frac{1}{8} \int_{t_n}^t \left[\int_{t_n}^{\xi} \left(\int_{t_n}^{\eta} \boldsymbol{\omega}(\zeta) d\zeta \right) \circ \boldsymbol{\omega}(\eta) d\eta \right] \circ \boldsymbol{\omega}(\xi) d\xi. \quad (6.4)$$

Аналогічно на основі третього відшукується четверте наближення

$$\mathbf{N}_{(4)}(t) = 1 + \frac{1}{2} \int_{t_n}^t \mathbf{N}_{(3)}(\xi) \circ \boldsymbol{\omega}(\xi) d\xi = \mathbf{N}_{(3)}(\xi) + \Delta \mathbf{N}_{(4)}(\tau),$$

$$\Delta \mathbf{N}_{(4)}(\tau) = \frac{1}{16} \int_{t_n}^t \left\{ \int_{t_n}^{\xi} \left[\int_{t_n}^{\eta} \left(\int_{t_n}^{\varsigma} \boldsymbol{\omega}(\nu) d\nu \right) \circ \boldsymbol{\omega}(\zeta) d\zeta \right] \circ \boldsymbol{\omega}(\eta) d\eta \right\} \circ \boldsymbol{\omega}(\xi) d\xi. \quad (6.5)$$

6.1.3. Побудова алгоритмів

Обмежуючись у подальшому відшукуванням алгоритмів чисельного інтегрування з точністю не більше за четвертий порядок мализни відносно малої величини кроку інтегрування, припустимо, що в діапазоні від t_n до t_{n+1} (від $\tau = 0$ до $\tau = h$) вектор $\boldsymbol{\omega} = \mathbf{i}\omega_x + \mathbf{j}\omega_y + \mathbf{k}\omega_z$ залежності вимірюваних проєкцій кутової швидкості об'єкту від часу можна подати у вигляді поліному третього степеня:

$$\boldsymbol{\omega} = \boldsymbol{\omega}_o + 2\boldsymbol{\varepsilon} \cdot \tau + 3\mathbf{a} \cdot \tau^2 + 4\mathbf{b} \cdot \tau^3, \quad (6.6)$$

де $\boldsymbol{\omega}_o$, $\boldsymbol{\varepsilon}$, \mathbf{a} і \mathbf{b} – постійні вектори, значення яких на кожному кроці інтегрування визначаються шляхом поліноміального інтерполювання по вимірних значеннях кутової швидкості (чи інтеграла від неї) на кількох попередніх кроках. З врахуванням цього відшукаємо значення інтегралів, що входять у (6.2), (6.3), (6.4) і (6.5), залишаючи члени не вище за четвертий порядок мализни відносно малого кроку інтегрування.

$$\int_{t_n}^t \boldsymbol{\omega}(\xi) d\xi = \int_0^{\tau} (\boldsymbol{\omega}_o + 2\boldsymbol{\varepsilon}\xi + 3\mathbf{a}\xi^2 + 4\mathbf{b}\xi^3) d\xi = (\boldsymbol{\omega}_o\tau + \boldsymbol{\varepsilon}\tau^2 + \mathbf{a}\tau^3 + \mathbf{b}\tau^4). \quad (6.7)$$

Далі, використовуючи знайдений вираз (6.7) знайдемо наступні інтеграли, нехтуючи членами більш високого порядку аніж четвертий відносно малої величини. Подвійний інтеграл набуде виду:

$$\int_0^\tau \left[\int_0^\xi \boldsymbol{\omega}(\eta) d\eta \right] \circ \boldsymbol{\omega}(\xi) d\xi \approx$$

$$\approx -\frac{1}{2} |\boldsymbol{\omega}_o|^2 \tau^2 + \frac{1}{3} (-3\boldsymbol{\omega}_o \cdot \boldsymbol{\varepsilon} + \boldsymbol{\omega}_o \times \boldsymbol{\varepsilon}) \tau^3 + \frac{1}{2} (-2\boldsymbol{\omega}_o \cdot \mathbf{a} + \boldsymbol{\omega}_o \times \mathbf{a} - |\boldsymbol{\varepsilon}|^2) \tau^4.$$

Наступний (потрійний) інтеграл матиме вигляд:

$$\int_0^\tau \left[\int_0^\xi \left(\int_0^\eta \boldsymbol{\omega}(\zeta) d\zeta \right) \circ \boldsymbol{\omega}(\eta) d\eta \right] \circ \boldsymbol{\omega}(\xi) d\xi \approx -\frac{1}{6} |\boldsymbol{\omega}_o|^2 \boldsymbol{\omega}_o \tau^3 - \frac{1}{6} [|\boldsymbol{\omega}_o|^2 \boldsymbol{\varepsilon} + 2(\boldsymbol{\omega}_o \cdot \boldsymbol{\varepsilon}) \boldsymbol{\omega}_o] \tau^4.$$

Нарешті четвертий інтеграл матиме наступний вигляд:

$$\int_0^\tau \left\{ \int_0^\xi \left[\int_0^\eta \left(\int_0^\varsigma \boldsymbol{\omega}(\nu) d\nu \right) \circ \boldsymbol{\omega}(\varsigma) d\varsigma \right] \circ \boldsymbol{\omega}(\eta) d\eta \right\} \circ \boldsymbol{\omega}(\xi) d\xi \approx \frac{1}{24} |\boldsymbol{\omega}_o|^4 \tau^4.$$

У підсумку, з точністю до малих четвертого порядку величина \mathbf{N}_{n+1} при $\tau = h$ визначається формулою

$$\begin{aligned} \mathbf{N}_{n+1} = & 1 + \frac{1}{2} (\boldsymbol{\omega}_o h + \boldsymbol{\varepsilon} h^2 + \mathbf{a} h^3 + \mathbf{b} h^4) + \\ & + \frac{1}{4} \left[-\frac{1}{2} |\boldsymbol{\omega}_o|^2 h^2 + \frac{1}{3} (-3\boldsymbol{\omega}_o \boldsymbol{\varepsilon} + \boldsymbol{\omega}_o \times \boldsymbol{\varepsilon}) h^3 + \frac{1}{2} (-2\boldsymbol{\omega}_o \mathbf{a} + \boldsymbol{\omega}_o \times \mathbf{a} - |\boldsymbol{\varepsilon}|^2) h^4 \right] + \\ & + \frac{1}{8} \left\{ -\frac{1}{6} |\boldsymbol{\omega}_o|^2 \boldsymbol{\omega}_o h^3 - \frac{1}{6} [|\boldsymbol{\omega}_o|^2 \boldsymbol{\varepsilon} + 2(\boldsymbol{\omega}_o \cdot \boldsymbol{\varepsilon}) \boldsymbol{\omega}_o] h^4 \right\} + \frac{1}{16} \left(\frac{1}{24} |\boldsymbol{\omega}_o|^4 h^4 \right). \end{aligned} \quad (6.8)$$

Якщо обмежитися утримуванням членів у правій частині виразу (6.8) не вище за перший порядок мализни відносно малої величини кроку опитування h , отримаємо формулу методу Пікара *першого наближення*:

$$\mathbf{N}_{(1)}(h) = 1 + \frac{1}{2} \boldsymbol{\omega}_o h \quad (6.9)$$

Залишивши члени до другого порядку мализни включно, дійдемо формули для методу Пікара *другого наближення*:

$$\mathbf{N}_{(2)}(h) = 1 + \frac{1}{2} \boldsymbol{\omega}_o h + \frac{1}{2} (\boldsymbol{\varepsilon} - \frac{1}{4} |\boldsymbol{\omega}_o|^2) h^2 \quad (6.10)$$

Аналогічно одержимо формулу для методу Пікара *третього наближення*:

$$\mathbf{N}_{(3)}(h) = 1 + \frac{1}{2} \boldsymbol{\omega}_o h + \frac{1}{2} (\boldsymbol{\varepsilon} - \frac{1}{4} |\boldsymbol{\omega}_o|^2) h^2 + \frac{1}{2} [\mathbf{a} + \frac{1}{6} (-3\boldsymbol{\omega}_o \cdot \boldsymbol{\varepsilon} + \boldsymbol{\omega}_o \times \boldsymbol{\varepsilon} - \frac{1}{4} |\boldsymbol{\omega}_o|^2 \boldsymbol{\omega}_o)] h^3. \quad (6.11)$$

Метод Пікара *четвертого наближення* визначиться формулою (6.8)

$$\begin{aligned} \mathbf{N}_{(4)}(h) = & 1 + \frac{1}{2} \boldsymbol{\omega}_o h + \frac{1}{2} (\boldsymbol{\varepsilon} - \frac{1}{4} |\boldsymbol{\omega}_o|^2) h^2 + \frac{1}{2} [\mathbf{a} + \frac{1}{6} (-3\boldsymbol{\omega}_o \cdot \boldsymbol{\varepsilon} + \boldsymbol{\omega}_o \times \boldsymbol{\varepsilon} - \frac{1}{4} |\boldsymbol{\omega}_o|^2 \boldsymbol{\omega}_o)] h^3 + \\ & + \frac{1}{2} [b + \frac{1}{4} (-2\boldsymbol{\omega}_o \cdot \mathbf{a} + \boldsymbol{\omega}_o \times \mathbf{a} - \boldsymbol{\varepsilon}^2) - \frac{1}{24} (|\boldsymbol{\omega}_o|^2 \boldsymbol{\varepsilon} + 2(\boldsymbol{\omega}_o \cdot \boldsymbol{\varepsilon}) \boldsymbol{\omega}_o) + \frac{1}{192} |\boldsymbol{\omega}_o|^4] h^4. \end{aligned} \quad (6.12)$$

6.1.4. Конкретизація значень коефіцієнтів

Нехай вимірюються величини псевдокоординат, тобто інтегралів від проекцій кутової швидкості об'єкта на координатні осі. Безпосередньо на кожному кроці опитування вимірювачів вимірюється вектор приростів квазікоординат

$$\mathbf{q}_n = \int_{t_n-h}^{t_n} \boldsymbol{\omega}(\xi) d\xi. \quad (6.13)$$

Розглядатимемо лише розгінні алгоритми.

Введемо час τ , початок якого збігається з моментом t_N початку поточного кроку опитування h . Тоді моментам часу $t = t_{N-3}; t_{N-2}; t_{N-1}; t_N; t_{N+1}$ відповідатимуть значення $\tau = -3h; -2h; -h; 0; h$. Користуючись попередньою залежністю (6.7) вектора кутової швидкості від часу, отримаємо:

$$\mathbf{q}_n = \int_{\tau_n-h}^{\tau_n} \boldsymbol{\omega}(\xi) d\xi = h [\boldsymbol{\omega}_o + \boldsymbol{\varepsilon}(2\tau_n - h) + \mathbf{a}(3\tau_n^2 - 3\tau_n h + h^2) + \mathbf{b}(4\tau_n^3 - 6\tau_n^2 h + 4\tau_n h^2 - h^3)].$$

Розпишемо значення цієї величини у моменти часу $\tau = h, 0, -h, -2h$

$$\begin{cases} \mathbf{q}_1 = h(\boldsymbol{\omega}_o + h\boldsymbol{\varepsilon} + h^2\mathbf{a} + h^3\mathbf{b}) \\ \mathbf{q}_0 = h(\boldsymbol{\omega}_o - h\boldsymbol{\varepsilon} + h^2\mathbf{a} - h^3\mathbf{b}) \\ \mathbf{q}_{-1} = h(\boldsymbol{\omega}_o - 3h\boldsymbol{\varepsilon} + 7h^2\mathbf{a} - 15h^3\mathbf{b}) \\ \mathbf{q}_{-2} = h(\boldsymbol{\omega}_o - 5h\boldsymbol{\varepsilon} + 19h^2\mathbf{a} - 65h^3\mathbf{b}) \end{cases} \quad (6.14)$$

Ці рівняння і є основою для обчислення невідомих коефіцієнтів $\boldsymbol{\omega}_o$, $\boldsymbol{\varepsilon}$, \mathbf{a} і \mathbf{b} .

Задля зручності використаємо позначення

$$\nabla = \nabla \mathbf{q}_1 = \mathbf{q}_1 - \mathbf{q}_0;$$

$$\nabla^2 = \nabla^2 \mathbf{q}_1 = \nabla \mathbf{q}_1 - \nabla \mathbf{q}_0 = \mathbf{q}_1 - 2\mathbf{q}_0 + \mathbf{q}_{-1};$$

$$\nabla^3 = \nabla^3 \mathbf{q}_1 = \nabla^2 \mathbf{q}_1 - \nabla^2 \mathbf{q}_0 = \mathbf{q}_1 - 3\mathbf{q}_0 + 3\mathbf{q}_{-1} - \mathbf{q}_{-2}.$$

Якщо обмежуватися лише *однокроковими алгоритмами*, в яких використовується вимірний вектор лише на поточному кроці інтегрування, то замість чотирьох рівнянь (6.14) матимемо одне, що дозволяє визначити лише один векторний коефіцієнт, а решту потрібно вважати рівними нулю:

$$\omega_o h = \mathbf{q}_1; \quad \varepsilon = \mathbf{a} = \mathbf{b} = 0. \quad (6.15)$$

Для побудови *двокрокових алгоритмів* достатньо обмежитися першими двома рівняннями (6.14) і припустити $\mathbf{a} = 0$ і $\mathbf{b} = 0$. В результаті отримаємо наступні значення векторних коефіцієнтів ω_o і ε :

$$\omega_o h = \mathbf{q}_1 - \frac{1}{2} \nabla; \quad \varepsilon h^2 = \frac{1}{2} \nabla; \quad \mathbf{a} = \mathbf{b} = 0, \quad (6.16)$$

інакше

$$\omega_o h = \frac{1}{2}(\mathbf{q}_1 + \mathbf{q}_2); \quad \varepsilon h^2 = \frac{1}{2}(\mathbf{q}_1 - \mathbf{q}_2).$$

Для *трикрокових алгоритмів* в (6.14) достатньо припустити $\mathbf{b} = 0$. Розв'язуючи отриману систему рівнянь, дійдемо до таких значень коефіцієнтів:

$$\omega_o h = \mathbf{q}_1 - \frac{1}{2} \nabla - \frac{1}{6} \nabla^2; \quad \varepsilon h^2 = \frac{1}{2} \nabla; \quad \mathbf{a} h^3 = \frac{1}{6} \nabla^2; \quad \mathbf{b} = 0. \quad (6.17)$$

Наостанок, після розв'язання рівнянь (6.14), які характерні *чотирикроковим алгоритмам*, векторні коефіцієнти набувають таких значень:

$$\begin{aligned} \omega_o h &= \mathbf{q}_1 - \frac{1}{2} \nabla - \frac{1}{6} \nabla^2 - \frac{1}{12} \nabla^3; & \varepsilon h^2 &= \frac{1}{2} \left(\nabla - \frac{1}{12} \nabla^3 \right); \\ \mathbf{a} h^3 &= \frac{1}{6} \left(\nabla^2 + \frac{1}{2} \nabla^3 \right); & \mathbf{b} h^4 &= \frac{1}{24} \nabla^3. \end{aligned} \quad (6.18)$$

6.1.5. Побудова алгоритмів

Поєднаємо отримані результати у єдиний алгоритм окремо для другого, третього та четвертого порядків. Для цього підставимо значення коефіцієнтів у вирази (6.6), (6.7) та (6.8) відповідно до порядків алгоритмів і залишимо в одержаних виразах члени порядку мализни, не вище за порядок методу. При цьому зважимо, що скінченна різниця наступного порядку є малою величиною більш високого порядку мализни, аніж різниця попереднього порядку. Можна вважати, що порядок мализни скінченної різниці є на одиницю вищою за порядок самої різниці. З врахуванням цього з попереднього випливають наступні алгоритми:

– однокроковий:
$$\mathbf{N}_{(1)}(h) = 1 + \frac{1}{2} \mathbf{q}_1; \quad (6.19)$$

– двокроковий:
$$\mathbf{N}_{(2)}(h) = 1 + \frac{1}{2} \mathbf{q}_1 - \frac{1}{32} |(2\mathbf{q}_1 - \nabla)|^2; \quad (6.20)$$

– трикроковий
$$\mathbf{N}_{n+1} = 1 + \frac{1}{2} \nabla - \frac{1}{8} |\nabla|^2 - \frac{1}{48} |\nabla|^2 \cdot \nabla + \frac{1}{24} \nabla \times \nabla^2; \quad (6.21)$$

– чотирікроковий

$$\mathbf{N}_{n+1} = 1 + \frac{1}{2} \nabla - \frac{1}{8} |\nabla|^2 - \frac{1}{48} |\nabla|^2 \cdot \nabla + \frac{1}{24} \nabla \times \nabla^2 + \frac{1}{48} \nabla \times \nabla^3 + \frac{1}{384} |\nabla|^4. \quad (6.22)$$

6.2. Створення комп'ютерної моделі

Задля дослідження похибок алгоритмів утворимо комп'ютерну модель у середовищі Matlab.

Перш за все потрібно створити процедури, які б на кожному кроці інтегрування забезпечували розрахунки кватерніону повороту згідно знайдених алгоритмів.

Текст цих процедур наведений нижче:

– процедура **Kv2m** двокрокового розгінного алгоритму

function Nout= Kv2m(h,t,q0,q1)

```

% Обчислення кватерніону повороту через два кроки опитування
% для випадку вимірювань приростів квазікоординат
% Лазарєв Ю. Ф. 02-04-2018
n1=q1/2; % Векторна частина кватерніону
n2=q1+q0; n3=n2*n2'/32;
n0=1-n3; % Скалярна частина кватерніону
Nout=[n0,n1]; % Кватерніон

```

– процедура **Kv3** трикрокового розгінного алгоритму

```

function Nout= Kv3(h,t,q0,q1)
% Обчислення кватерніону повороту через три кроки опитування
% для випадку вимірювання приростів квазікоординат
% Ю. Ф. Лазарєв 02-04-2018
n1=q1/2; kvdq1=(q1*q1');
n2=kvdq1/24; n3=kvdq1/8;
dQ1=q1-q0; n4=cross(q1,dQ1)/24;
n0=1-n3; % Скалярна частина кватерніону
n=n1*(1-n2)+n4; % Векторна частина кватерніону
Nout=[n0,n]; % Кватерніон

```

– процедура **Kv4m** чотирьохкрокового розгінного алгоритму

```

function Nout= Kv4m(h,t,q_2,q_1,q0,q1)
% Обчислення кватерніону повороту через чотири кроки опитування
% для випадку вимірювання приростів квазікоординат
% Ю. Ф. Лазарєв 02-04-2018
n1=q1/2; kvdq1=(q1*q1');
n2=kvdq1/24; n3=kvdq1/8;
dQ1=q1-q0; n4=cross(q1,dQ1)/24;
n0=1-n3; n=n1*(1-n2)+n4;
dQ0=q0-q_1; dQ21=dQ1-dQ0;
n5=cross(q1,dQ21)/48;
n6=kvdq1^2;
n0=n0+n6/384; % Скалярна частина кватерніону
n=n+n5; % Векторна частина кватерніону
Nout=[n0,n]; % Кватерніон

```

Далі подано текст керувальної програми **Kvasi_uprm**:

```
% Kvasi_uprm
% Дослідження похибок розгінних алгоритмів чисельного інтегрування
% 1) кватерніонного рівняння орієнтації
% 2) багатокроковими алгоритмами другого, третього і четвертого порядку
% 3) шляхом використання вимірювальної інформації про прирости
%   квазікоординат (інтегралів від проекцій кутової швидкості основи)
% Використовуються наступні процедури й функції:
%   idbso2Pseudo.m - процедура змінювання значень вхідних даних
%   UgDvObPseudo.m - функція обчислення параметрів поточного кутового руху
%                   основи (кутів, кутових швидкостей, квазікоординат )
%   Kv2m   - процедура двокрокового алгоритму
%   Kv3    - процедура трикрокового алгоритму
%   Kv4m   - процедура чотирокрокового алгоритму
%   grafik91.m - процедура побудови графіків 9 величин від однієї величини
%               на трьох підграфіках один під одним
%   dngrbsoPseudo.m - процедура виведення на поле графіків розрахункових
%                   даних у текстовій формі
% Лазарев Ю. Ф.           07-04-2018
echo off, clc
clear all
global sprogram sname
global D2
global D5
global Dr2 Dr3 Dr4
sprogram='Kvasi-uprm.m';
clc
disp(' Ви увійшли в програму Kvasi_uprm,')
disp('яка інтегрує кінематичні рівняння у кватерніонах')
disp('по виміряних квазікоординатах і обчислює похибки цього чисельного інтегрування'), disp(' ')
k0=menu('Що робити?','Продовжити роботу','Закінчити роботу');
if k0==1           % Вхід у програму
    sname='Лазарев Ю. Ф.';
    % Введення "вшитих" значень первісних параметрів руху основи
    ps0=0;   te0=0;   fi0=0;
    psm=0;   tem=0.1; fim=0.1;
    omps=0;  omte=1;  omfi=1;
    eps=0;   ete=90*pi/180;  efi=0;
    % параметрів інтегрування;
```

```

h=0.01;  Nt0=0;  Ndtf=5;
% Заповнення глобального вектора D;
D2=[ps0, te0, fi0, psm, tem, fim, omps, omte, omfi, eps, ete, efi];
D4=[h, Nt0, Ndtf];
      % ОРГАНІЗАЦІЯ ЗОВНІШНЬОГО ЦИКЛУ
while k0==1
    % Визначення величини кроку інтегрування h
    h = D4(1);      Tn0 = 1;
    % Визначення кінцевого моменту часу tfinal інтегрування
    tfinal= 100;      D5=[h Tn0 tfinal];
    % БЛОК ІНТЕГРУВАННЯ
    t=0;      % Завдання початкового моменту часу
    % Завдання вектора початкових умов у попередніх точках
    z=UGDVOBPpseudo(t-2*h,h); % Параметри руху ОСНОВИ у момент часу на два кроки назад
    pk2=[z(4) z(5) z(6)]; % Прирости квазікоординат у момент часу на два кроки назад
    z=UGDVOBPpseudo(t-h,h); % Параметри руху ОСНОВИ у попередній точці
    pk1=[z(4) z(5) z(6)]; % Прирости квазікоординат у попередній момент часу
    z=UGDVOBPpseudo(t,h); % Параметри руху ОСНОВИ у поточній точці
    pk=[z(4) z(5) z(6)]; % Прирости квазікоординат у поточний момент часу
    kv=[2 3 1]; Ug=[-z(1),z(2),z(3)];
    Ls=Ug2quat(Ug,kv); % Кути у поточний момент часу
    z=UGDVOBPpseudo(t+h,h); % Параметри руху ОСНОВИ у наступній точці
    pk_1=[z(4) z(5) z(6)]; % Прирости квазікоординат у наступний момент часу
    Ls2=Ls; Ls3=Ls; Ls4=Ls;
        imas=1; % Лічильник елементів вихідних масивів
        % ЦИКЛ ІНТЕГРУВАННЯ
    disp('Почекайте (йде обчислення)')
    while (imas-1)*h<tfinal
        t=(imas-1)*h;
        %%%%%%%%% Двокроковий алгоритм
        N2= Kv2m(h,t,pk,pk_1); L2 = Mult_Kwat(Ls2,N2); Ls2 = L2;
        %%%%%%%%% Трикроковий алгоритм
        N3= Kv3(h,t,pk,pk_1); L3 =Mult_Kwat(Ls3,N3); Ls3 = L3;
        %%%%%%%%% Чотирикроковий алгоритм
        N4= Kv4m(h,t,pk2,pk1,pk,pk_1); L4 =Mult_Kwat(Ls4,N4);
        Ls4 = L4;      pk2=pk1; pk1=pk; pk=pk_1;
        z=UGDVOBPpseudo(t+2*h,h); % Поточне положення основи
        pk_1=[z(4) z(5) z(6)];
        z=UGDVOBPpseudo(t+h,h);
    % Формування масивів похибок алгоритмів

```

```

        % для двокрокового алгоритму
Yg2=quat2ug(L2,kv);
dpsi2(imas)=Yg2(1)+z(1); % Масиви похибок
dteta2(imas)=Yg2(2)-z(2); % оцінювання
dfi2(imas)=Yg2(3)-z(3); % кутів повороту основи
        % для трикрокового алгоритму
Yg3=quat2ug(L3,kv);
dpsi3(imas)=Yg3(1)+z(1); % Масиви похибок
dteta3(imas)=Yg3(2)-z(2); % оцінювання
dfi3(imas)=Yg3(3)-z(3); % кутів повороту основи
        % для чотирікрокового алгоритму
Yg4=quat2ug(L4,kv);
dpsi4(imas)=Yg4(1)+z(1); % Масиви похибок
dteta4(imas)=Yg4(2)-z(2); % оцінювання
dfi4(imas)=Yg4(3)-z(3); % кутів повороту основи
tt(imas)=t; % Формування масиву моментів часу
imas=imas+1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
stit1='Погрешности алгоритмов (радианы)';
syab1='Kv2m - двухшаговый'; sxlal1=' '; stit2=' ';
syab2='Kv3 - трехшаговый'; sxlal2=' '; stit3=' ';
syab3='Kv4m - четырехшаговый'; sxlal3='Время, с';
ГРАФИК91(tt,dpsi2,dteta2,dfi2,stit1,syab1,sxlal1,...
    dpsi3,dteta3,dfi3,stit2,syab2,sxlal2,...
    dpsi4,dteta4,dfi4,stit3,syab3,sxlal3);
[Const,Dr2]=drift_L(tt,dpsi2); [Const,Dr3]=drift_L(tt,dpsi3);
[Const,Dr4]=drift_L(tt,dpsi4);
DnGrBSOPseudo; % Виведення на поле графіків розрахункових даних
% Кінець процедури виведення графіків
k0=menu('Що робити?','Продовжити роботу','Закінчити роботу');
end
end % Вихід з циклу програми
disp('Кінець програми')

```

6.3. Моделювання роботи БІСО

Для виявлення залежності похибок алгоритмів від зсуву фаз між коливаннями основи моделювання проведено при двох значеннях зсуву фаз $\varepsilon = 0^\circ$ та $\varepsilon = 90^\circ$ (рис. 6.1 і 6.2).

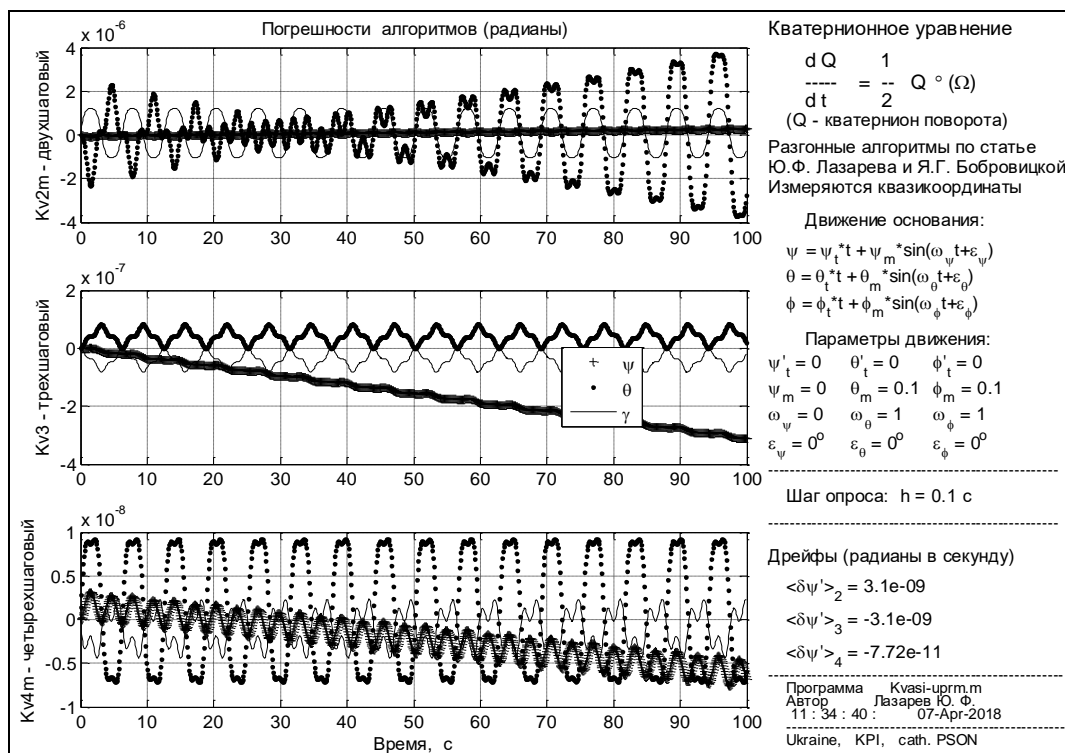


Рис. 6.1. Похибки алгоритмів при значенні кроку $h=0,1$ с і за зсуву фаз 0°

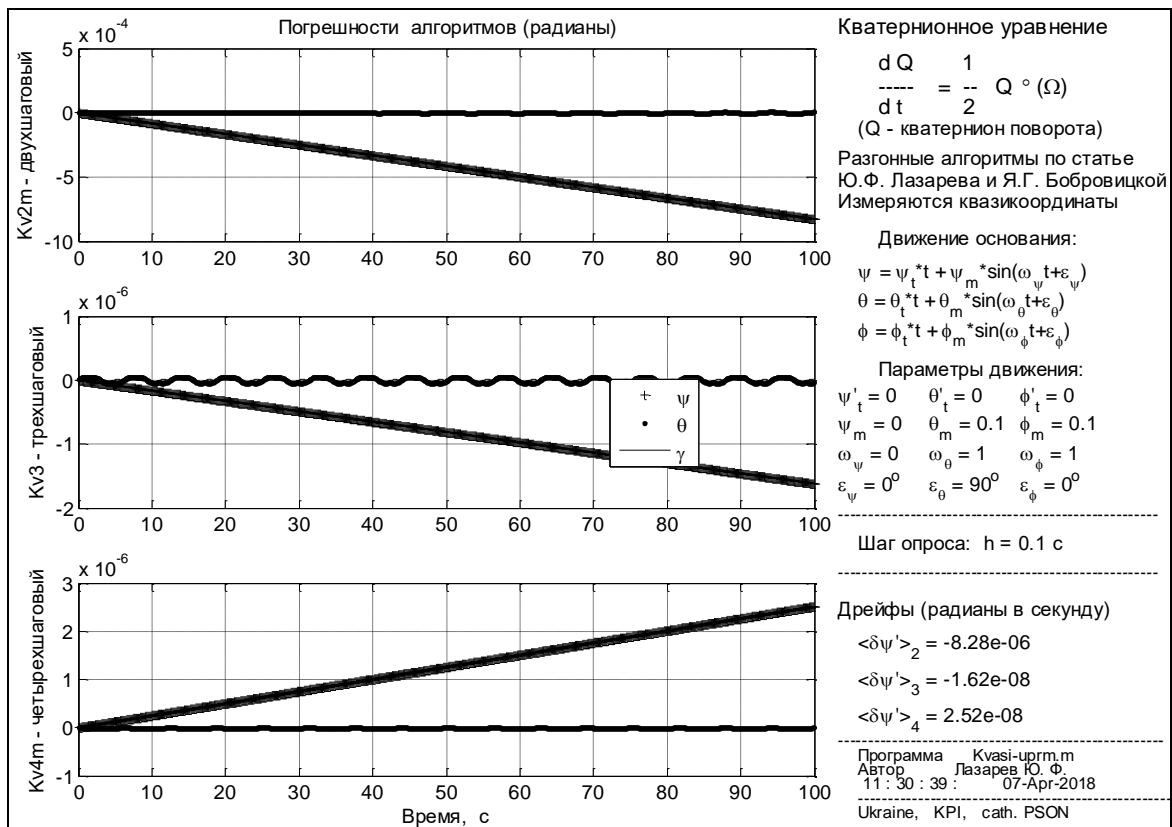


Рис. 6.2. Похибки алгоритмів при значенні кроку $h=0,1$ с і за зсуву фаз 90°

З них впливає, що найбільший за величиною дрейф спостерігається за зсуву фаз 90° .

Щоб встановити залежність максимальних дрейфів похибок від кроку інтегрування, моделювання проведено при двох значеннях кроку $h = 0,1$ с та $h = 0,01$ с. Результати моделювання наведені на рис. 6.3 та 6.4.

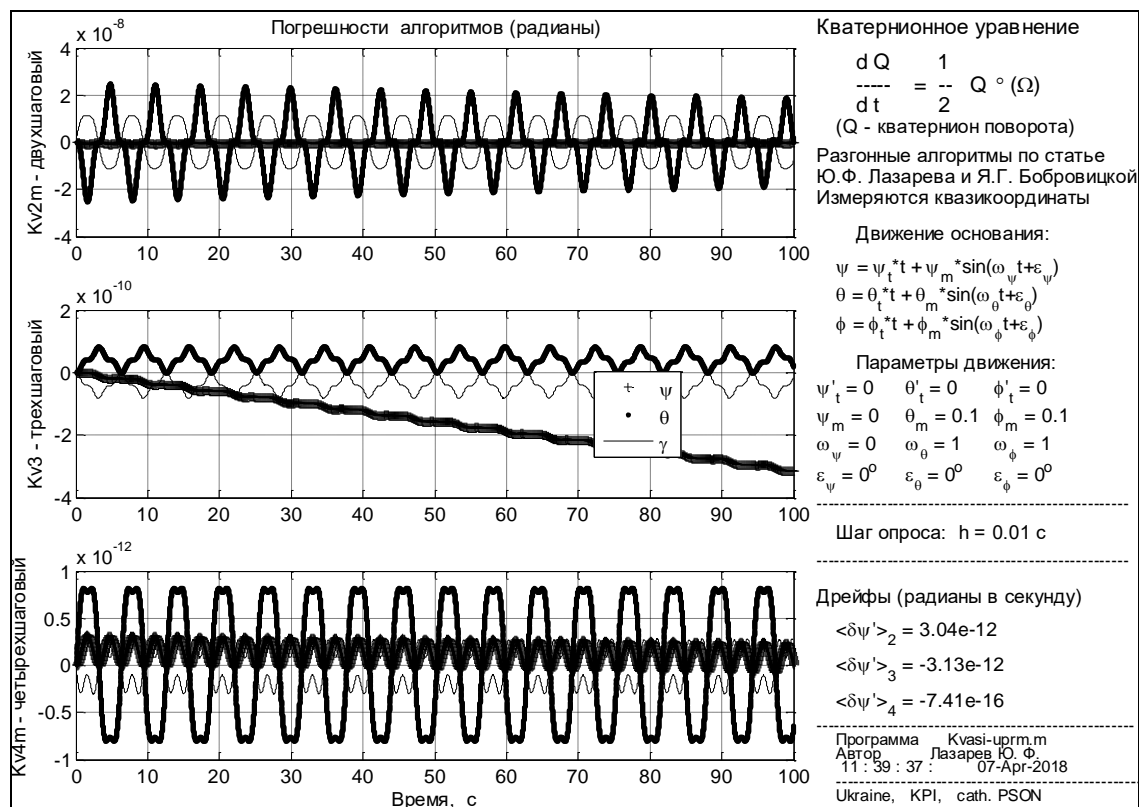


Рис. 6.3. Похибки алгоритмів при значенні кроку $h=0,01$ с і за зсуву фаз 0°

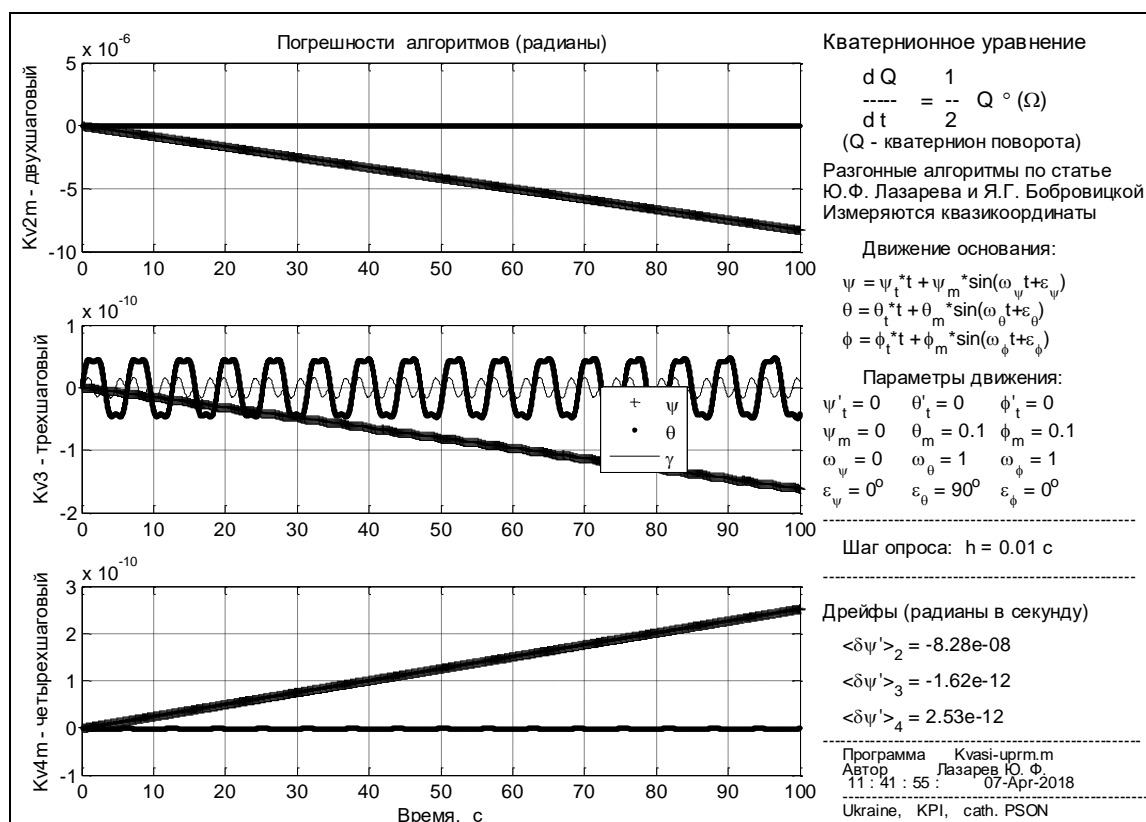


Рис. 6.4. Похибки алгоритмів при значенні кроку $h=0,01$ с і за зсуву фаз 90°

Ці експерименти дозволяють зробити висновок, що алгоритм $Kv4=2m$ має другий порядок точності, $Kv3$ і $Kv4m$ – четвертий порядок.

Більш детальні відомості, отримані моделюванням, містяться у тексті програми **Gr_RA**, поданому нижче.

```
% Gr_RA
% Графики залежостей безрозмірних максимальних дрейфів похибок
% розгінних алгоритмів
% Лазарев Ю.Ф. 07-04-2018
clear all, clc
am= 0.1; om=1; k=am^2*om;
h=[1 5e-1 2e-1 1e-1 5e-2 2e-2 1e-2 5e-3 2e-3 1e-3]; mu=om*h;
Kv2=[-7.87e-4 -2.04e-4 -3.31e-5 -8.28e-6 -2.07e-6 -3.31e-7 -8.28e-8 ...
-2.07e-8 -3.31e-9 -8.28e-10];
Kv3=[-1.44e-4 -9.83e-6 -2.58e-7 -1.62e-8 -1.01e-9 -2.59e-11 -3.13e-12 ...
-3.9e-13 -2.5e-14 -3.13e-15];
Kv4=[1.52e-4 1.4e-5 3.97e-7 2.52e-8 1.58e-9 4.05e-11 2.53e-12 ...
1.58e-13 4.05e-15 2.52e-16];
mu1=[1e-3 1]; kv2=8.3e-2*mu1.^2; kv4=2.5e-2*mu1.^4;
mu2=[1e-3 2e-2]; kv31=3.1e-4*mu2.^3;
mu3=[2e-2 1]; kv32=1.6e-2*mu3.^4;
loglog(mu,abs(Kv2)/k,'s',mu1,kv2,'.',mu,abs(Kv3)/k,'ko',...
mu2,kv31,'k--',mu3,kv32,'k:',...
mu,abs(Kv4)/k,'rp',mu1,kv4,'r:', 'LineWidth',3), grid
set(gca,'FontSize',14)
title('Розгінні алгоритми. Кватерніонне рівняння. Квazікоординати')
xlabel(' \mu = \omegah'), ylabel(' \delta ')
legend('Kv2m','0,083\mu^2','Kv3','3,1e-4\mu^3','0,016\mu^4',...
'Kv4m','0,025\mu^4',4)
```

Відповідні графіки наведено на рисунку 6.5.

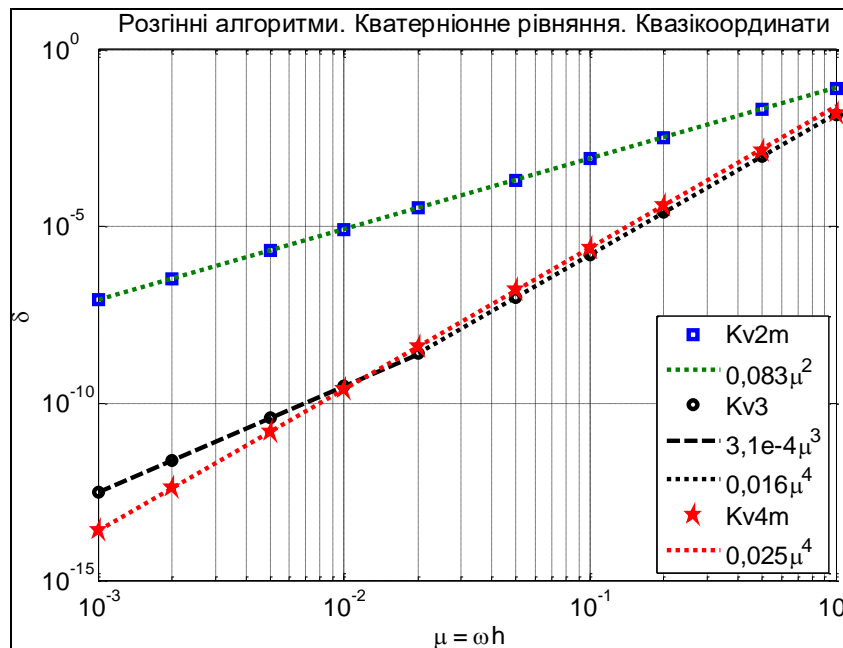


Рис. 6.5. Графічне подання залежностей безрозмірних дрейфів похибок від частотного параметра

На тому ж рисунку 6.5 представлені графіки відповідних апроксимувальних частотних степеневих функцій виду (3.4).

Параметри цих степеневих залежностей наведені у Таблиці 6.1.

Таблиця 6.1. Параметри точності розгінних алгоритмів

Алгоритм	Діапазон	k	N
Kv2m	$0,001 \leq \mu \leq 1$	0,083	2
Kv3	$0,001 \leq \mu \leq 0,02$	$3,1e-4$	3
Kv3	$0,02 \leq \mu \leq 1$	0,016	4
Kv4m	$0,001 \leq \mu \leq 1$	0,025	4

6.3. Висновки

Викладена методика побудови алгоритмів є надійною і дозволяє будувати алгоритми й більш високого порядку.

Щоб застосовувати знайдені апросимувальні степеневі залежності для знаходження реального можливого максимального дрейфу похибки досліджуваного алгоритму слід зробити наступне:

1) встановити дійсну частоту f (у Герцах) опитування вимірювачів досліджуваного БІСО, а отже, й пов'язану з неї величину кроку опитування $h = \frac{1}{f}$ (у секундах); наприклад, частоті опитування 100 Гц відповідає крок опитування 0,01 с:

2) ознайомитися зі спектральною характеристикою руху основи, на який встановлюється БІСО; виявити максимальну частоту ω_{\max} кутових коливань та відповідну максимально можливу величину амплітуди a_m цих коливань; припустимо, що $\omega_{\max} = 10$ радіан у секунду, а $a_m = 0,2$ радіана;

3) розрахувати частотний параметр $\mu = \omega_{\max} \cdot h$; для прийнятих значень $\mu = 0,1$;

4) розрахувати величину "опорного" дрейфу за формулою (див. (3.2))

$Dr_{op} = \omega_{\max} a_m^2$; у розглядуваному прикладі це буде $Dr_{op} = 0,4$ радіан в секунду;

5) за встановленими параметрами k і N апросимувальної степеневі залежності (3.4) обраного алгоритму БІСО визначити *безрозмірну* величину максимального дрейфу похибки; наприклад, для алгоритму Kv2m ($k = 0,083$; $N = 2$) і $\mu = 0,1$ він виявиться $\delta = 8,3 \cdot 10^{-4}$, а для алгоритму Kv4m ($k = 0,025$; $N = 4$) – $\delta = 2,5 \cdot 10^{-6}$;

6) тепер можна визначити величину максимального дрейфу обраного алгоритму за встановлених умов коливань основи за формулою

$$Dr_{\max} = \delta \cdot Dr_{op};$$

так, для алгоритму Kv2m $Dr_{\max} = 3,32 \cdot 10^{-4} \text{ рад/с} = 68,5$ градусів у годину,

а для алгоритму Kv4m – $Dr_{\max} = 1,0 \cdot 10^{-6} \text{ рад/с} = 0,21$ градусів у годину.

Література

1. Лазарев Ю. Ф. Кінематика твердого тіла: навчальний посібник. – К.: НТУУ "КПІ", 2014. – 61 с. [електронний ресурс: <http://kafpson.kpi.ua/arch.htm>]
2. Лазарев Ю. Ф. Моделювання на ЕОМ: Навчальний посібник. – К.: Корнійчук, 2007. – 290 с.
3. Бранец В. Н., Шмыглевский И. П. Введение в теорию бесплатформенных инерциальных навигационных систем. – М.: Наука, 1992. – 280 с
4. Лурье А. И. Аналитическая механика. – М.: ФизМатЛит, 1961. – 824 с.
5. Панов А. П. Математические основы теории инерциальной ориентации. – Киев: Наукова думка, 1995. – 280 с.
6. Марк Дж., Тазартес Д. Конические алгоритмы, учитывающие неидеальность частотной характеристики выходных сигналов гироскопов // Гироскопия и навигация. – 2000. – № 1(28). – С. 65-77
7. Буденный М. Ф. и др. Об автоматизированном проектировании системы программно-аппаратных средств на базе гиперкомплексных чисел для задач ориентации твердого тела. Часть 2.// Реєстрація, зберігання і обробка даних, 2002. – Т. 4. – № 4. – С. 77-84.
8. Лазарев Ю. Ф. и др. Результаты V-тестирования безразгонных алгоритмов бесплатформенных инерциальных систем ориентации // X Международная научно-техническая конференция "Гиротехнологии, навигация, управление движением и конструирование авиационно-космической техники". Сборник докладов. – К.: НТУУ "КПІ", 2015 – С. 475-482
9. Ільчук С. В. Алгоритми безплатформової інерціальної системи орієнтації на основі рівняння Пуассона \\\ Магістерська дисертація. – Київ, НТУУ "КПІ", 2017. – 158 с.

10. Лазарев Ю. Ф., Бобровицька Я. Г. Зіставлення алгоритмів чисельного інтегрування кінематичних рівнянь орієнтації твердого тіла у кватерніонах за методом Пікара // Наукові вісті НТУУ "КПІ". – 2006. № 6. – С. 113-118.