

*Ю. Ф. Лазарєв
Я. Г. Бобровицька*

**РОЗРОБЛЕННЯ І МОДЕЛЮВАННЯ
АЛГОРИТМІВ
БЕЗПЛАТФОРМОВОЇ СИСТЕМИ ОРІЄН-
ТАЦІЇ**

Електронний навчальний посібник

Рекомендовано Методичною радою НТУУ "КПІ"

Київ – КПІ – 2011

УДК 681.3(0.75)
Л17

Лазарев Ю. Ф., Бобровицька Я. Г.

Л17 Розроблення і моделювання алгоритмів безплатформової системи орієнтації.
Електронний навчальний посібник. – Київ: НТУУ "КПІ", 2011. – 135 с.

Викладені основи теорії безплатформових інерціальних систем навігації (БІНС). Головна увага приділена питанням обрання алгоритмів частини БІНС – безплатформової системи орієнтації (БСО). Детально розглянуті різні види кінематичних рівнянь орієнтації і різні види алгоритмів чисельного інтегрування цих рівнянь з точки зору похибок визначення кутів орієнтації основи. Показані шляхи утворення програм у середовищі Matlab, які здійснюють моделювання поведінки БСО за цими алгоритмами. Проведено моделювання головних видів алгоритмів. Виявлені деякі неточності щодо оцінок похибок алгоритмів, а також і можливості і особливості застосування MATLAB для моделювання БСО.

Призначено для курсового і дипломного проектування студентів вищих навчальних закладів. Може бути корисним для інженерів і науковців.

Іл. 41, табл. 8

Зміст

ПЕРЕДМОВА	5
ВСТУП. РЕЗУЛЬТАТИ РОЗРОБОК ІНЕРЦІАЛЬНИХ НАВІГАЦІЙНИХ СИСТЕМ	7
1. ПРИНЦИПИ ПОБУДОВИ БІНС	13
1.1. ХАРАКТЕРИСТИКА ОСОБЛИВОСТЕЙ РЕАЛІЗАЦІЇ НАВІГАЦІЙНИХ ОБЧИСЛЕНЬ У БІНС.....	13
1.2. ПОЗНАЧЕННЯ.....	15
1.3. ПОСТАНОВКА ЗАДАЧІ	16
1.4. РОЗРОБКА МОДЕЛЮВАЛЬНОЇ СХЕМИ АЛГОРИТМУ	17
1.4.1. Системи координат. Форми основного рівняння інерціальної навігації.....	18
1.4.2. Моделювальна схема 1	22
1.4.3. Моделювальна схема 2	23
1.4.4. Порівняння моделювальних схем	24
1.5. АЛГОРИТМИ РОЗВ'ЯЗУВАННЯ СИСТЕМ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ	25
1.5.1. Розв'язування задачі орієнтації.....	25
1.5.2. Розв'язування задачі визначення відносної швидкості.	28
1.6. ВИСНОВКИ.....	29
2. ОЦІНЮВАННЯ ПОХИБОК ВИЗНАЧЕННЯ ОРІЄНТАЦІЇ	30
2.1. РІВНЯННЯ ПОХИБОК ОРІЄНТАЦІЇ, ОБУМОВЛЕНІ ПОХИБКАМИ ВИМІРЮВАННЯ КУТОВОЇ ШВИДКОСТІ	31
2.2. ДЖЕРЕЛА ПОХИБОК ОРІЄНТАЦІЇ І ПРИВЕДЕННЯ ЇХ ДО ЕКВІВАЛЕНТНИХ ПОХИБОК ЧЭ.	35
2.2.1. Похибки через відхилення осей чутливості від розрахункових положень.....	36
2.2.2. Похибки аналого-цифрового перетворювача	37
2.2.3. Похибки алгоритмів визначення орієнтації.....	37
3. КІНЕМАТИЧНІ РІВНЯННЯ ОРІЄНТАЦІЇ І ЗАГАЛЬНІ ШЛЯХИ ПОБУДОВИ АЛГОРИТМІВ ЇХ ЧИСЕЛЬНОГО ІНТЕГРУВАННЯ	39
3.1. КУТОВІ ПАРАМЕТРИ	40
3.2. МАТРИЧНІ ПАРАМЕТРИ	41
3.3. ВЕКТОРНІ ПАРАМЕТРИ.....	41
3.4. ПОПЕРЕДНІ ВИСНОВКИ.....	43
3.5. ОСОБЛИВОСТІ ЗДІЙСНУВАННЯ ЧИСЕЛЬНОГО ІНТЕГРУВАННЯ РІВНЯНЬ ОРІЄНТАЦІЇ	44
4. АЛГОРИТМИ МЕТОДУ ПРОГНОЗ-КОРЕКЦІЇ	48
4.1. РОЗРОБКА АЛГОРИТМІВ	48
4.2. МЕТА МОДЕЛЮВАННЯ І ЗАСОБИ ЇЇ ДОСЯГНЕННЯ	49
4.2.1. Процедури обчислення параметрів поточного руху основи.....	49
4.2.2. Процедури методів прогноз-корекції.....	51
4.2.3. Процедури правих частин рівнянь орієнтації.....	53
4.3. ОПИС ПРОГРАМНИХ ЗАСОБІВ	54
4.4. РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ.....	66
4.5. ВИСНОВКИ.....	75
5. АЛГОРИТМИ МЕТОДУ ПІКАРА ДЛЯ РІВНЯННЯ ПУАССОНА	76
5.1. РОЗРОБЛЕННЯ АЛГОРИТМІВ.....	76
5.1.1. Постановка задачі	76
5.1.2. Побудова наближених розв'язків допоміжного рівняння.....	77
5.1.3. Уточнення розв'язків для поліноміальної апроксимації кутової швидкості.....	78
5.1.4. Визначення коефіцієнтів інтерполяції для випадку вимірювань кутової швидкості.....	80
5.1.5. Визначення коефіцієнтів інтерполяції для випадку вимірювань вектору псевдокоординат.....	82
5.1.6. Побудова алгоритмів	84
5.2. ЗАСОБИ МОДЕЛЮВАННЯ АЛГОРИТМІВ	86
5.3. ОПИС ПРОГРАМНИХ ЗАСОБІВ	88
5.4. РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ	95
5.5. ВИСНОВКИ.....	101

6. АЛГОРИТМИ МЕТОДА ПІКАРА ДЛЯ РІВНЯННЯ У КВАТЕРНІОНАХ.....	103
6.1. ПОБУДОВА АЛГОРИТМІВ	103
6.1.1. Побудова наближених розв'язків допоміжного рівняння.....	104
6.1.2. Уточнення розв'язків для поліноміальної апроксимації кутової швидкості.....	105
6.1.3. Побудова алгоритмів.....	107
6.2. ЗАСОБИ МОДЕЛЮВАННЯ АЛГОРИТМІВ	108
6.3. ОПИС ПРОГРАМНИХ ЗАСОБІВ МОДЕЛЮВАННЯ АЛГОРИТМІВ.....	110
6.4. РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ.....	116
6.5. ПОРІВНЯННЯ АЛГОРИТМІВ ЧИСЕЛЬНОГО ІНТЕГРУВАННЯ КІНЕМАТИЧНИХ РІВНЯНЬ ОРІЄНТАЦІЇ У КВАТЕРНІОНАХ ЗА МЕТОДОМ ПІКАРА	122
6.6. ВИСНОВКИ.....	125
ЗАГАЛЬНІ ВИСНОВКИ	126
ЛІТЕРАТУРА	126
ДОДАТОК. БІБЛІОТЕКА ДОПОМІЖНИХ ПРОГРАМ.....	128
АБЕТКОВИЙ ПОКАЖЧИК.....	136

Передмова

Безплатформові інерціальні системи навігації (БІНС) у теперішній час є найбільш розповсюдженими і перспективними для застосування у цілях навігації транспортних засобів найрізноманітнішого призначення (літаки, ракети, керовані снаряди, підводні кораблі тощо). Одним із шляхів підвищення ефективності і точності виконання ними свого призначення є розробка ефективних алгоритмів безплатформової системи орієнтації (БСО), що входить до їх складу. Незважаючи на значний обсяг літератури з розробки алгоритмів БСО, на наш час немає навчальних посібників, які забезпечили б достатньо дохідливе ознайомлення початківців з тематикою, задачами і методами розв'язування цієї проблеми.

Цей навчальний посібник призначений подолати цей недолік. Його присвячено введенню студентів у проблематику розроблення ефективних алгоритмів БСО і розроблення засобів моделювання точності цих алгоритмів.

Посібник складається з наступних розділів.

У вступі наведені результати розробок БІНС за матеріалами закордонної літератури. Розкрита динаміка застосування різних навігаційних інерціальних систем. Показано, що загальна тенденція полягає у все більшому використанні на практиці безплатформових інерціальних навігаційних систем.

У першій главі “Принципи побудови БІНС” наводяться основи побудови БІНС, розкриваються її задачі, головні алгоритми їх розв'язування на борті рухомого об'єкту, засоби їх реалізації. Проводиться їх порівняльний аналіз і робляться перші висновки щодо їх застосування.

Наступна друга глава “Оцінювання похибок визначення орієнтації” знайомить із джерелами похибок визначення поточної орієнтації рухомого об'єкта і методами їх оцінювання.

У третій главі “Кінематичні рівняння орієнтації і загальні шляхи побудови алгоритмів їх чисельного інтегрування” наводяться відомі форми кінематичного рівняння орієнтації і визначаються особливості здійснювання чисельного інтегрування рівнянь орієнтації на борті рухомого об'єкту.

Решта глав (четверта, пята і шоста) містять матеріали, що стосуються безпосередньо розробки чисельних алгоритмів БСО на основі інформації, що надається бортовими вимірювачами у дискретні моменти часу на бортовий обчислювач.

У четвертій главі розглядається найбільш простий підхід до створення алгоритмів – розповсюдження відомих алгоритмів методу прогноз-корекції на випадок застосування бортових вимірювачів проекцій кутової швидкості об'єкта. Наводиться детальний опис комплексу програм у середовищі Matlab, який забезпечує моделювання похибок розроблених алгоритмів. Результати подані у графічній формі і показують достатню ефективність розроблених алгоритмів. Встановлені порядки розроблених алгоритмів.

П'ята глава присвячена розробці алгоритмів найбільш поширеного на наш час методу послідовних наближень Пікара у застосуванні його до матричного рівняння орієнтації Пуассона. Також приведений опис програмних засобів

моделювання похибок, здійснене моделювання, яке показало ефективність розроблених алгоритмів.

У шостій главі метод Пікара застосований до кватерніонного рівняння орієнтації. Розглянуті випадки двох типів вимірювачів:

- 1) проєкцій кутової швидкості;
- 2) приростів псевдокоординат кутового положення об'єкту.

Показано, що алгоритми четвертого порядку у відомій літературі є помилковими. Розроблені дієві методи четвертого порядку.

У додатку наведені тексти процедур. Що здійснюють необхідні для моделювання операції над кватерніонами, матрицями напрямних косинусів і кутами орієнтації.

Наведені відомості і тексти програм стануть у нагоді студентам у курсовому і дипломному проектуванні, а також є корисними для розробників алгоритмів БСО.

Вступ. Результати розробок інерціальних навігаційних систем

(за матеріалами закордонної літератури)

Практичне використання інерціальних навігаційних систем (ІНС) на літальних апаратах було розпочато в США на початку 1960-х років, коли фірма Літтон після майже десятилітніх досліджень і розробок опанувала серійний випуск ІНС типу LN-3, що в ті часи мала високі точнісні характеристики:

- похибка визначення місця розташування – 3,7 км за годину;
- похибка визначення швидкості – 0,6 м/с;
- похибка визначення курсу – 6 кутові мінути;
- похибка визначення крену і тангажу – 3 кутові мінути;
- маса – 36,6 кг;
- об'єм – 53 дм³.

Спочатку ІНС встановлювалися на бойових літаках, а наприкінці 1960-х років на транспортних і пасажирських літаках, причому ІНС стала основним датчиком навігаційної інформації: місця розташування, швидкості, положення в просторі і курсу.

У порівнянні з іншими системами ІНС має наступні переваги:

- безперервно видає поточне значення координат місця і швидкості;
- не випромінює енергії і є абсолютно завадостійкою;
- видає інформацію на всіх широтах, у будь-яку погоду, не потребуючи наземного обладнання;
- навігаційна інформація, що видається ІНС, істотно не залежить від маневрування об'єкту;
- забезпечує вихідну інформацію про місце, шляхову швидкість, напрямки та вертикаль.

До недоліків ІНС слід віднести:

- точність інформації про місце і швидкість падає зі збільшенням тривалості роботи системи незалежно від того, пересувається чи ні об'єкт, на якому встановлена ІНС;
- відносно складне і дороге технічне обслуговування і висока вартість;
- необхідне попереднє виставлення (юстировка) ІНС.

Інерціальні навігаційні системи, що застосовуються в системах керування балістичних ракет, літаків, морських кораблів і інших рухомих об'єктах, зазвичай мають у своєму складі гіростабілізовану платформу (ГСП). Функції, що виконуються ГСП, зводяться, по-перше, до забезпечення визначеної просторової орієнтації тригранника, утвореного вимірювальними осями акселерометрів, і, по-друге, наявність стабілізованої платформи дозволяє одержати інформацію про кутове положення об'єкта щодо платформи, тобто визначити параметри орієнтації. Причому у якості параметрів, що характеризують орієнтацію об'єкта, використовується певна система кутів, вимірювання яких здійснюється датчиками, встановленими на осях підвісу платформи.

Гіростабілізована платформа ІНС є високопрезизійним електромеханічним пристроєм, до складу якого крім точних гіроскопів входять ще ряд елементів (моментні датчики, двигуни розвантаження, датчики кутів тощо), характеристики яких мають задовольняти досить жорстким вимогам.

При конструюванні платформи вирішується ряд складних суперечливих задач, таких, наприклад, як забезпечення потрібної жорсткості підвісу, підведення живлення і зняття електричних сигналів, забезпечення теплового режиму гіроскопів тощо. При цьому звичайно необхідно задовольнити деяким габаритним і масовим обмеженням. У силу складності і великої кількості електромеханічних елементів ГСП зазвичай є найбільш уразливим елементом ІНС із погляду надійності. Застосування ГСП у складі ІНС призводить до збільшення часу підготовки всієї системи до роботи, оскільки необхідно здійснювати так назване початкове виставлення системи, у ході якого ГСП надається необхідна орієнтація в просторі.

Платформи зазвичай мають обмежені кути «прокачування», що, у свою чергу, при використанні ІНС на високоманеврених об'єктах, призводить до необхідності встановлення додаткових рам підвісу і використання слідкуючих систем, що мають високу швидкодію.

Вищезазначене коло проблем, які виникають при проектуванні і побудові інерціальних навігаційних систем із ГСП, не є органічно властивим ІНС. Принципово система може бути побудована без включення в її склад стабілізованої платформи. Цей клас ІНС одержав назву безплатформових (безкарданових) інерціальних навігаційних систем (БІНС).

У системах інерціальної навігації без стабілізованої платформи чутливі елементи - акселерометри і вимірювачі параметрів орієнтації – монтуються на відповідній платі, що кріпиться на корпусі об'єкта. При цьому кількість вимірювачів має бути такою, щоб забезпечувалося одержання інформації про вектор удаваного прискорення точки об'єкту, на якому встановлені вимірювачі, і про векторну величину (наприклад, вектор абсолютної кутової швидкості), що характеризує обертання основи. Цієї інформації разом із апріорними даними про гравітаційне поле і початкові умови руху достатньо для наступного обчислення на борту рухомого об'єкту навігаційних параметрів і параметрів, що характеризують орієнтацію.

До числа потенційних переваг БІНС у порівнянні з платформовими ІНС можна віднести:

- менші масу, розміри й енергоємність;
- істотне спрощення механічної частини системи і її компоновки;
- відсутність обмежень на кути повороту;
- скорочення часу початкового виставлення;
- підвищення універсальності системи, оскільки перехід до визначення тих або інших параметрів навігації здійснюється алгоритмічно;
- спрощення розв'язку задачі резервування і контролю працездатності системи і її елементів.

Разом з тим при створенні БНС виникають принципові труднощі, головними з яких є:

- розробка чутливих елементів, що мають широкий діапазон вимірювання і прийнятну точність в умовах їхнього жорсткого кріплення на борту об'єкту;
- розробка бортових цифрових обчислювальних машин (БЦОМ), що мають високу швидкодію, і їхнього математичного забезпечення.

Існуючі вимоги по точності ІНС встановлюють максимальну похибку визначення місця розташування у 1,85 км за годину, швидкості – 0,75 м/с.

Підвищення точності ІНС може бути забезпечено, у першу чергу, підвищенням точності гіроскопів і акселерометрів.

Першими прецизійними гіроскопами для ІНС були поплавцеві гіроскопи, причому при їхньому створенні були досягнуті великі успіхи: випадковий дрейф був зменшений з 0,01 до 0,001 °/годину. Однак недоліки, властиві поплавцевим гіроскопам (висока вартість, малий термін служби, необхідність створення особливих умов для обробки, завеликий час готовності, вимоги жорсткого контролю робочої температури гіроскопа), стали причиною створення гіроскопів на нових фізичних принципах: динамічно настроєних, електростатичних, волоконно-оптичних, лазерних, акустичних і інших.

Динамічно настроєні гіроскопи (ДНГ) широко застосовуються у тих випадках, коли передбачаються високі кутові швидкості (декілька сот градусів за секунду), високий діапазон прискорень (кілька сотень), невеликі розміри і маса, висока стабільність (декілька сотих часток градуса за годину) і низька вартість. У порівнянні з іншими гіроскопами ДНГ мають істотні переваги:

- відносно стабільна повторюваність постійної складової дрейфу;
- висока надійність і тривалий термін служби, обмежений шарикопідшипниками, що застосовуються, або електронними елементами;
- низька вартість;
- швидкий перехід із режиму прогріву;
- низький рівень випадкового шуму;
- відсутність контакту рідин з чутливим елементом;
- відносна нечутливість до радіації.

До основних недоліків ДНГ відносяться:

- чутливість до механічних резонансів на високих частотах і до вібрацій з подвійною частотою власного обертання;
- наявність високих магнітних полів у швидкообертovому роторі;
- наявність шуму на окремих гармоніках частоти власного обертання.

Типовими робочими характеристиками ДНГ є:

- стабільність дрейфу від запуску до запуску – 0,01...1 °/годину ;
- динамічний діапазон - 10^8 ... 10^9 °/годину;
- діапазон перевантажень – 50...400 g ;
- діапазон вимірювання кутової швидкості – 100...1000 °/годину;
- чутливість до впливу температур – 0,0056°;
- споживана потужність – 0,1...1 Вт.

Фірма СФІМ (Франція) у 1980 р. почала розробку міні ДНГ типу GAM5. Гіроскоп має діаметр 1,3 см і довжину 1,2 см. У закінченому варіанті (з електронними схемами) об'єм гіроскопа не перевищував 4 см^3 , а маса – 10 г. Вимірювання сил, необхідних для осьового центрування маховика, забезпечує визначення кутового прискорення корпусу гіроскопа. Таким чином, гіроскоп GAM5 є одночасно й акселерометром. Конструкція гіроскопа має достатню жорсткість, витримуючи перевантаження до 10000-20000 уздовж осі обертання. Споживана потужність датчика моменту менша за 20 мВт.

Німецька філія американської фірми Літтон-Літеф випускає ДНГ К-273, які є чутливими елементами ІНС LTR-81, розробленої в США фірмою Літтон.

Середній час напрацювання на відмову системи LTR-81 дорівнює 8000 годин. Таку надійність мають лише лазерні інерціальні системи. Система LTR-81 у два рази дешевша й у два рази легша за лазерну інерціальну систему.

Понад 1000 гіроскопів К-273 встановлено на літаках А-300, MD-80, «Челенджер», F-50, F-100 і ін. Навігаційні задачі на цих літаках розв'язуються за допомогою системи LTR-81.

Головною причиною інтересу до лазерних гіроскопів є їхня висока надійність. Через відсутність у кільцевому лазерному гіроскопі (КЛГ) рухомих частин його наробіток на відмову складає до 30000 годин. Комплектує електронне устаткування всієї лазерної ІНС обчислюватиметься тисячами годин, а це набагато більше за звичайну ІНС.

За іншими критеріями перевага лазерних ІНС на сучасному етапі розвитку технології не є настільки явною:

- при рівних точнісних характеристиках вартість ІНС на механічних гіроскопах несуттєво відрізняється від вартості ІНС на лазерних гіроскопах;
- точність лазерних гіроскопів нижча, ніж у кращих механічних, і недостатньо точне визначення кутових швидкостей обмежує їхнє застосування на керованих снарядах.

У той же час вартість технічного обслуговування систем на лазерних гіроскопах у 4 рази нижча за звичайні ІНС.

Характеристики лазерного гіроскопа визначаються двома основними параметрами – систематичною похибкою і масштабним коефіцієнтом.

У некорегованих КЛГ вихідний імпульс кутової швидкості гіроскопа зникає при малих кутових швидкостях ($1 \text{ }^\circ/\text{с}$ - область захоплення), є нелінійним при середніх кутових швидкостях і тільки при більш високих кутових швидкостях ($100^\circ/\text{с}$) наближається до ідеального.

Основну масу лазерних гіроскопів передбачається використовувати в ІНС середньої точності з похибкою не більше за 1,48 км/годину.

Таблиця 0.1. Точнісні характеристики гіроскопів

	Тип гіроскопа	Швидкість кутового дрейфу ($\text{в}^0/\text{годину}$)
1.	Поплавцевий інтегруючий IG-8	2
2.	ДНГ G-7 фірми Літтон	0,7
3.	Лазерний газонаповнений	0,6

4.	Найкращий триступеневий	0,07
5.	Поплавцевий інтегруючий ITGG	0,06
6.	Волоконно-оптичний	2 (1988 р.) 0,01 (2005 р.)

У таблиці 0.1. наведені для порівняння точнісні характеристики електро-механічних і лазерних гіроскопів.

Гіроскопи, розраховані на використання в безкарданових інерціальних платформах, задовольняють вимогам до військової апаратури по радіаційній, вібраційній і ударній міцності, а також спряженню з військовим стандартом MIL-SID-1750A.

Надійність ІНС із трьома КЛГ військового призначення має середній наробіток на відмову порядку 2000 годин, а витрати на придбання й експлуатацію її в шість разів менші за витрати на механічні інерціальні системи.

Висока надійність інерціальних систем на основі КЛГ, застосовуваних на цивільних літальних апаратах, переконала військових фахівців у їхній перспективності. Виставлення лазерних ІНС за курсом, дані про який зберігаються в пам'ятовуючому пристрої, відбувається з високою точністю і швидше ніж у звичайних ІНС.

Лазерна ІНС являє собою безплатформову систему, у якій для визначення просторового положення об'єкта замість електромеханічних гіроскопів використовуються КЛГ.

Волоконно-оптичні гіроскопи (ВОГ) – відносно новий тип оптичних датчиків. Принцип його дії, так само як і КЛГ, заснований на «вихровому» ефекті Сан'яка.

До основних характеристик ВОГ можна віднести наступні:

- потенційно високу (вище 10^{-3} °/годину) чутливість приладу, що вже зараз складає 0,1...0,01°/годину;
- малі габарити і масу, особливо в зв'язку з можливістю створення ВОГ цілком на оптичних інтегральних схемах (ОІС);
- невисоку вартість (\$ 1000...1500) і відносну простоту конструкції;
- надзвичайно мале енергоспоживання, що особливо важливо при використанні ВОГ у бортовій апаратурі;
- великий динамічний діапазон вимірюваних кутових швидкостей;
- відсутність обертових механічних елементів, що підвищує надійність і робить дешевшим виробництво;
- миттєва готовність до роботи;
- практично необмежений термін зберігання;
- нечутливість до великих лінійних прискорень;
- висока завадостійкість і нечутливість до потужних зовнішніх електромагнітних впливів;
- мала вразливість до дії гамма-нейтронної радіації.

До недоліків ВОГ слід віднести його чутливість до динамічних температурних градієнтів уздовж волокна, акустичним шумам і вібраціям, оптичним нелінійним ефектам, флуктуаціям інтенсивності і поляризації випромінювання,

дробовим шумам фотодетектора тощо, що призводить до паразитних дрейфів нуля і погіршує точність приладу.

У США проведені дослідження з вивчення перспектив постачань ІНС для авіаційно-космічного застосування за участю основних американських фірм – постачальників бортового електронного устаткування. Результати досліджень відобразили тенденцію до явного переходу від систем із ГСП до БІНС, що показано в таблиці 0.2.

Таблиця 0.2. Співвідношення випуску платформових і безплатформових ІНС авіаційно-космічного застосування

Об'єкт установки	1984 р.		1989 р.	
	Із ГСП	БІНС	Із ГСП	БІНС
Воєнні літаки	До 100%	-----	50%	50%
Пасажирські літаки	30%	70%	10%	90%
Тактичні крилаті ракети	83%	17%	34%	66%
Стратегічні ракети	84%	16%	56%	44%

У дослідженні також відзначаються потенційні переваги ВОГ – відсутність частин, що рухаються, можливість одержання різних рівнів точності, низька собівартість.

1. Принципи побудови БІНС

1.1. Характеристика особливостей реалізації навігаційних обчислень у БІНС

Датчики первинної інформації БІНС – гіротахметри й акселерометри – встановлюються жорстко на літальних апаратах. Важкі умови роботи датчиків інформації призводять до великих похибок, а це означає, що необхідно здійснити аналітичну компенсацію похибок вимірювачів перш ніж їх сигнали будуть використані для розрахунку параметрів орієнтації і для визначення складових удаваного прискорення уздовж навігаційних осей.



Рис. 1.1. Схема обробки навігаційної інформації

Для корекції показань гіротахметрів необхідна математична модель вимірювача, у якій зазвичай враховують:

- нелінійність;
- неспіввісність осей гіротахметрів;
- дрейф;
- спотворення масштабного коефіцієнта.

Математична модель акселерометра зазвичай враховує:

- спотворення масштабного коефіцієнта;
- нелінійність;
- неспіввісність осей акселерометрів.

Сигнали з виходу аналітичного компенсатора похибок використовуються для визначення параметрів орієнтації. Потім, знаючи ці параметри, визначають матрицю напрямних косинусів C . Така послідовність визначення параметрів орієнтації пояснюється зручністю компенсації похибок параметрів орієнтації, що виникають у процесі їхнього розрахунку. Перетворення складових удаваного прискорення $a_{x,y,z}$ від осей об'єкта до осей навігаційної системи координат здійснюється за допомогою матриці напрямних косинусів C .

Для БНС, що працюють з використанням гіротахometrів специфічними задачами є:

- аналітична компенсація похибок гіротахometrів і акселерометрів перед використанням їхніх сигналів для побудови параметрів орієнтації й одержання проєкцій удаваного прискорення на осі навігаційної системи координат;
- визначення параметрів орієнтації і розрахунок напрямних косинусів;
- перетворення сигналів акселерометрів, скоректованих у компенсаторі похибок, від осей об'єкта до осей навігаційної системи координат.

Лише розв'язок задачі навігаційних обчислень у БНС є аналогічним за такий у платформовій ІНС. Три перераховані вище задачі властиві тільки БНС і їх розв'язок передуює навігаційним обчисленням.

У процесі розв'язування задач розрахунку параметрів орієнтації і перетворення сигналів акселерометрів виникають специфічні похибки через обчислення, які необхідно зменшувати правильним вибором робочих алгоритмів, а якщо можливо, то і компенсувати їх у самих чутливих елементах.

Обсяг обчислень БНС є значним. Це пояснюється в основному тим фактом, що БЦОМ повинна мати справу з динамікою поступального руху літального апарату (ЛА). Поступальні швидкості ЛА відносно малі. Наприклад, швидкість при польоті в напрямку на північ 1100 км/годину відповідає швидкості змінювання широти усього 10°/годину.

Інтегрування для одержання швидкості і місця розташування може здійснюватися досить точно з використанням простих методів чисельного інтегрування при низькій частоті повторення (у типовому випадку 10...20 Гц).

Кутові швидкості ракети в типовому випадку по величині на кілька порядків більші за ті, що викликані поступальними швидкостями. У результаті цього інтегрування рівнянь кутового положення в БНС пов'язане з жорсткими вимогами до БЦОМ. Інтегрування для одержання кутового положення являє собою операцію на яку витрачається велика частина часу. Для алгоритму Рунге-Кутта четвертого порядку при частоті відновлення 50 Гц у БЦОМ для визначення кутового положення використовується приблизно 70% доступного часу роботи цього пристрою.

Для основних функцій розкладу прискорень і визначення навігаційних даних потрібно близько 30% доступного часу роботи навігаційного обчислювача.

Компенсація похибок вимірювачів вимагає витрати близько 40% доступного часу роботи пристрою керування «Введення-виведення».

У таблиці нижче показані основні етапи роботи БНС з розв'язування навігаційних задач.

Таблиця 1.1. Етапи роботи БНС і їх особливості

Функції	Вхід	Операція	Частота вих., Гц	Вихід
Обробка сигналів гіроскопів	Приріст кутів	Компенсація зміщення Виставка осей Масштабування Виділення швидкості	100	Кутова швидкість в осях об'єкта
Орієнтація БНС	Кутові швидкості об'єкту та Землі	Орієнтація шляхом інтегрування кватерніону Ортонормалізація кватерніону Переведення кватерніону в напрямні косинуси	100	Напрямні косинуси для переходу до координат із місцевою вертикаллю
Обробка сигналів акселерометрів	Приріст швидкості	Компенсація і виставлення акселерометрів Компенсація прискорення	100 20	Приріст швидкості в координатах із місцевою вертикаллю
Обчислення швидкості	Приріст швидкості в СК із місцевою вертикаллю	Інтегрування Диференціювання висоти для отримання вертикальної швидкості	20	Швидкість в земній системі координат із місцевою вертикаллю
Навігація	Висота; Швидкість в земній СК	Обчислення широті і довготи, курсу, складових швидкості обертання Землі	20	Геодезична широта і довгота Курс Висота

1.2. Позначення

Уведемо наступні позначення:

O - система координат x, y, z , пов'язана з об'єктом, початок якої розміщений в точці O центра мас об'єкта; надалі будемо називати її ОСВ – «об'єктова система відліку»;

I - інерціальна система координат (ІСВ – інерціальна система відліку), одна з осей якої спрямована по лінії географічних полюсів, а початок міститься у центрі Землі;

E - система координат, жорстко пов'язана з Землею (земна система відліку – ЗСВ); початок її розміщується у центрі Землі, одна вісь спрямована по лінії географічних полюсів, інша вісь лежить у площині Гринвічського меридіана;

N - навігаційна система координат (НСВ); початок у центрі Землі, одна з осей спрямована по вертикалі місця, тобто по лінії, що з'єднує центр Землі і центр мас об'єкта.

ω^e – абсолютна кутова швидкість Землі (ЗСВ);

ω^o – абсолютна кутова швидкість об'єкта (ОСВ);

ω^{ne} – кутова швидкість повороту НСВ щодо Землі;

λ, φ - географічні довгота і широта місця об'єкта;

v^o – вектор абсолютної швидкості центра мас об'єкта;

v^{oe} – вектор швидкості об'єкта щодо Землі;

r – радіус-вектор центра мас об'єкта щодо центра Землі;

C^{SP} – матриця напрямних косинусів (МНК) переходу від системи відліку S до системи відліку P :

C^{SP}	1_P	2_P	3_P
1_S	c_{11}^{sp}	c_{11}^{sp}	c_{11}^{sp}
2_S	c_{11}^{sp}	c_{11}^{sp}	c_{11}^{sp}
3_S	c_{11}^{sp}	c_{11}^{sp}	c_{11}^{sp}

d_S – матриця - стовпець із проєкцій вектора \mathbf{d} на осі системи відліку S ;

$(d_S \times)$ – кососиметрична матриця з проєкцій вектора \mathbf{d} на осі системи відліку s вигляду:

$$(d_S \times) = \begin{bmatrix} 0 & -d_{3S} & d_{2S} \\ d_{3S} & 0 & -d_{1S} \\ -d_{2S} & d_{1S} & 0 \end{bmatrix}. \quad (1.1)$$

1.3. Постановка задачі

Потрібно розробити алгоритм перетворення інформації блоку вимірювальних перетворювачів БНС із метою одержання поточних значень наступних параметрів руху об'єкта:

- географічної широти φ , довготи λ , висоти над рівнем океану H (барометричної висоти);
- проєкцій вектора швидкості руху центра мас об'єкта щодо Землі, на осі географічного тригранника v_n^{oe} ;

- кутів, що характеризують поточну орієнтацію будівельних осей об'єкта щодо осей географічного супровідного тригранника: ψ - істинний курс, θ – тангаж, γ - крен;

- відстань S від точки старту до точки перебування об'єкта.

Передбачається, що інформація про кутовий рух об'єкта вимірюється гіротахометрами і надходить на вхід алгоритму у вигляді приростів кутів повороту об'єкта навколо осей чутливості гіроскопів у дискретні моменти часу $t_i - ih$; ($i = 1, 2, \dots$), де h – період опитування датчиків:

$$\Delta\theta_{j,i} = \int_{t_{i-1}}^{t_i} \omega_{0j}^o(\tau) d\tau, \quad (j = 1, 2, 3); \quad (1.2)$$

ω_{0j}^o проекції вектора абсолютної кутової швидкості обертання об'єкта на осі чутливості гіроскопів.

Інформація з акселерометрів надходить у вигляді приростів удаваної швидкості в моменти $t_i - ih$; ($i = 1, 2, \dots$):

$$\Delta V_{j,i} = \int_{t_{i-1}}^{t_i} a_{0j}(\tau) d\tau, \quad (j = 1, 2, 3); \quad (1.3)$$

де a_{0j} – проекції удаваного прискорення на осі чутливості акселерометрів.

Вхідні параметри БНС мають визначатися в дискретні моменти часу з частотою $f_{вх}$. Припускається, що гіроскопи й акселерометри жорстко закріплені на деякій основі (площадці), осі якої виставлені з достатньою точністю паралельно будівельним осям об'єкта. Осі чутливості гіроскопів і акселерометрів приблизно співвісні осям об'єкту. Шляхом вимірювань з необхідною точністю отримують матриці напрямних косинусів C_r і C_a (неортогональні), що описують реальне розташування осей чутливості гіроскопів і акселерометрів щодо осей об'єкта.

Розробка алгоритму передбачає виконання наступних робіт:

- вибір і обґрунтування моделювальної схеми алгоритму;
- вибір чисельних методів розв'язування систем диференціальних рівнянь, що входять в алгоритм;
- розробку і налагодження програм, що реалізують обраний алгоритм;
- моделювання алгоритму з метою визначення його точнісних характеристик, оптимізації параметрів алгоритму.

1.4. Розробка моделювальної схеми алгоритму

При проектуванні БНС перш за все потрібно вибрати й обґрунтувати структуру алгоритму і вибрати орієнтовно чисельні методи розв'язання систем диференціальних рівнянь, що входять в алгоритм.

1.4.1. Системи координат. Форми основного рівняння інерціальної навігації

Алгоритм інерціальної навігаційної системи являє собою замкнену систему аналітичних залежностей, що дозволяє по вхідних даних (початкових значеннях географічних координат і кутів повороту об'єкта в географічній системі, показаннях гіроскопів і акселерометрів) визначати необхідні вихідні величини. Схему, що зображує функціональні зв'язки між окремими аналітичними залежностями, називають моделювальною схемою алгоритму.

Опишемо системи координат, зв'язки між ними, а також моделі фігури Землі і гравітаційного прискорення.

Як модель фігури Землі будемо використовувати сфероїд з параметрами Ф.Н. Красовського:

- велика піввісь $a = 6378245 \text{ м}$;
- мала піввісь $b = 6356856 \text{ м}$;
- стиснення $\alpha = (a - b) / a = 0,003352$;
- квадрат першого ексцентриситету $\varepsilon^2 = (a^2 - b^2) / b^2 = 0,006692$;
- квадрат другого ексцентриситету $\varepsilon'^2 = (a^2 - b^2) / a^2 = 0,006738$.

При розробці алгоритму будемо використовувати наступні праві декартові системи координат:

1) інерціальна геоцентрична екваторіальна система відліку (ІСВ) $\mathbf{I}(0,1_i,2_i,3_i)$; початок системи координат (рис. 1.2) розміщений у центрі Землі, орт 1_i спрямований уздовж осі обертання Землі, орти 2_i і 3_i лежать у площині екватора, причому в початковий момент часу t_0 орт 2_i лежить у площині Гринвічського меридіана; система координат \mathbf{I} зберігає незмінне кутове положення відносно напрямку на віддалені зірки;

2) земна геоцентрична екваторіальна система відліку (ЗСВ) $\mathbf{E}(0,1_e,2_e,3_e)$; початок системи координат (рис. 1.3) міститься у центрі Землі, орт 1_e спрямований уздовж осі обертання Землі, орти $2_e, 3_e$ лежать у площині екватора, причому орт 2_e у будь-який момент часу знаходиться в площині Гринвічського меридіана; орти $2_e, 3_e$ обертаються разом із Землею із кутовою швидкістю $\omega^e = 7,29 \cdot 10^{-5} \text{ 1/с}$ навколо орта 1_e ; у початковий момент часу t_0 орти \mathbf{E} збігаються з однойменними ортами системи \mathbf{I} ; орієнтація системи координат \mathbf{E} відносно \mathbf{I} в довільний момент часу t описується матрицею напрямних косинусів C^{ie} :

$$C^{ie} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \lambda_u & -\sin \lambda_u \\ 0 & \sin \lambda_u & \cos \lambda_u \end{bmatrix}, \quad (1.4)$$

$$\text{де} \quad \lambda_u = \omega^e t. \quad (1.5)$$

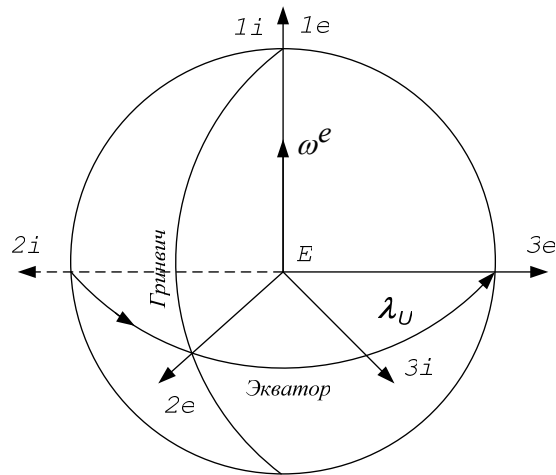


Рис. 1.2. Інерціальна геоцентрична система координат

3) навігаційна система відліку (НСВ) $\mathbf{N}(0,1_n,2_n,3_n)$; початок її розташований у центрі Землі, орт 2_n спрямований по географічній вертикалі місця, орти $1_n, 3_n$ лежать у площині місцевого горизонту, причому орт 1_n лежить також у площині меридіана і спрямований на Північ, а орт 3_n спрямований на Схід; кутове положення системи координат \mathbf{N} щодо системи координат \mathbf{E} визначається кутами довготи λ і географічної широти φ точки місця розташування об'єкта; орієнтація описується матрицею C^{en} :

$$C^{en} = \begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\cos \lambda \sin \varphi & \cos \lambda \cos \varphi & -\sin \lambda \\ -\sin \varphi \sin \lambda & \sin \lambda \cos \varphi & \cos \lambda \end{bmatrix}; \quad (1.6)$$

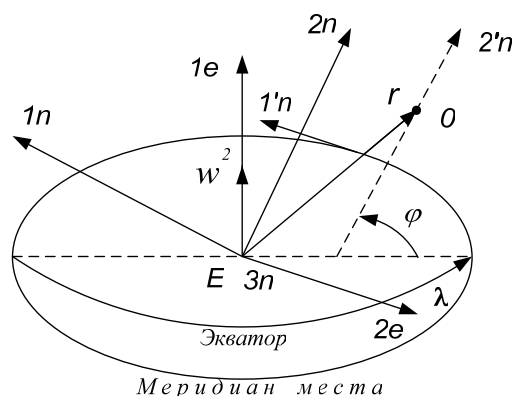


Рис. 1.3. Навігаційна система координат

4) система координат, пов'язана з об'єктом (ОСВ) $\mathbf{O}(0,1_o,2_o,3_o)$; початок системи координат (рис. 1.4) міститься в центрі

мас об'єкта; орти $1_o, 2_o, 3_o$ спрямовані паралельно будівельним осям об'єкта відповідно повздовжній (x), нормальній (y) і поперечній (z) осям; кутове положення системи ОСВ щодо навігаційної системи координат описується трьома кутами: курсу ψ , тангажу θ і крену γ ; орієнтація ОСВ відносно НСВ описується матрицею напрямних косинусів C^{no} :

$$C^{no} = \begin{bmatrix} \cos\theta\cos\psi & -\sin\gamma\sin\psi - \cos\gamma\cos\psi\sin\theta & -\cos\gamma\sin\psi + \sin\gamma\cos\psi\sin\theta \\ \sin\theta & \cos\theta\cos\gamma & -\sin\gamma\cos\theta \\ \cos\theta\sin\psi & \sin\gamma\cos\psi - \cos\gamma\sin\psi\sin\theta & \cos\gamma\cos\psi + \sin\gamma\sin\psi\sin\theta \end{bmatrix}$$

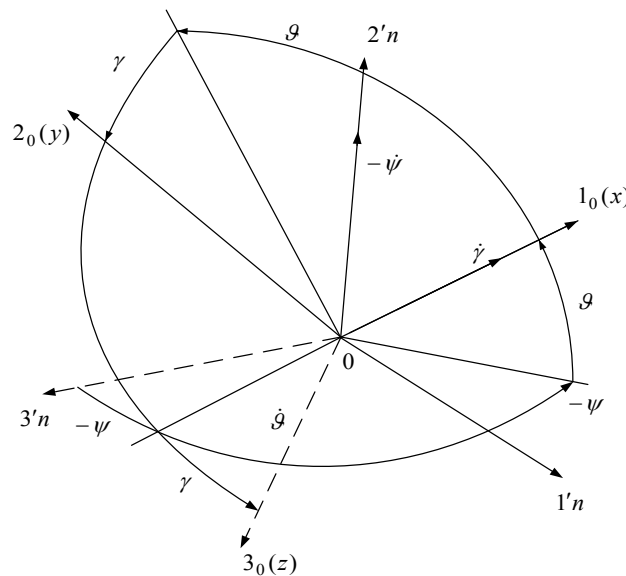


Рис. 1.4. Об'єктова система координат

Модель прискорення в інерціальній системі координат **I**:

$$g_{3i} = -\frac{g'r_{3i}}{r}(1 - 3\mu\frac{r_{1i}^2}{r^2}); \quad g_{1i} = -\frac{g'r_{1i}}{r}\left(1 + 3\mu\frac{(r_{2i}^2 + r_{3i}^2)}{r^2}\right); \quad (1.7)$$

$$g_{2i} = -\frac{g'r_{2i}}{r}(1 - 3\mu\frac{r_{1i}^2}{r^2}); \quad g' = \frac{g_{\text{э.о.}}a^2}{r^2}(1 - \beta\frac{r_{1i}^2}{r^2}).$$

$$\beta = 1,437 \cdot 10^{-3}, \quad 3\mu = 3,27 \cdot 10^{-3}, \quad g_{\text{э.о.}} = 9,81438 \text{ м/с}^2, \quad r^2 = r_{1i}^2 + r_{2i}^2 + r_{3i}^2.$$

Модель прискорення сили тяжіння в НСВ має вигляд:

$$g_{1n}^* = 0; \quad g_{2n}^* = g(1 + 5,2884 \cdot 10^{-3} \sin^2 \varphi)[1 - \frac{2H}{a}(1 - \varepsilon^2 \sin^2 \varphi)]; \quad g_{3n}^* = 0, \quad (1.8)$$

де H – барометрична висота об'єкта.

Позначимо: v^o – вектор абсолютної (відносно ІСВ) швидкості об'єкта; v^{oe} – вектор відносної (щодо Землі) швидкості об'єкта; a – вектор удаваного прискорення; ω^o – абсолютна кутова швидкість обертання об'єкта.

Основне рівняння інерціальної навігації може бути подане в різних формах, у залежності від того, яка зі швидкостей v^o або v^{oe} використовується, і від того, в проекціях на яку систему відліку записуються рівняння. Із цих багатьох форм найбільший інтерес становлять дві. Будемо називати їх «схемою 1» і «схемою 2».

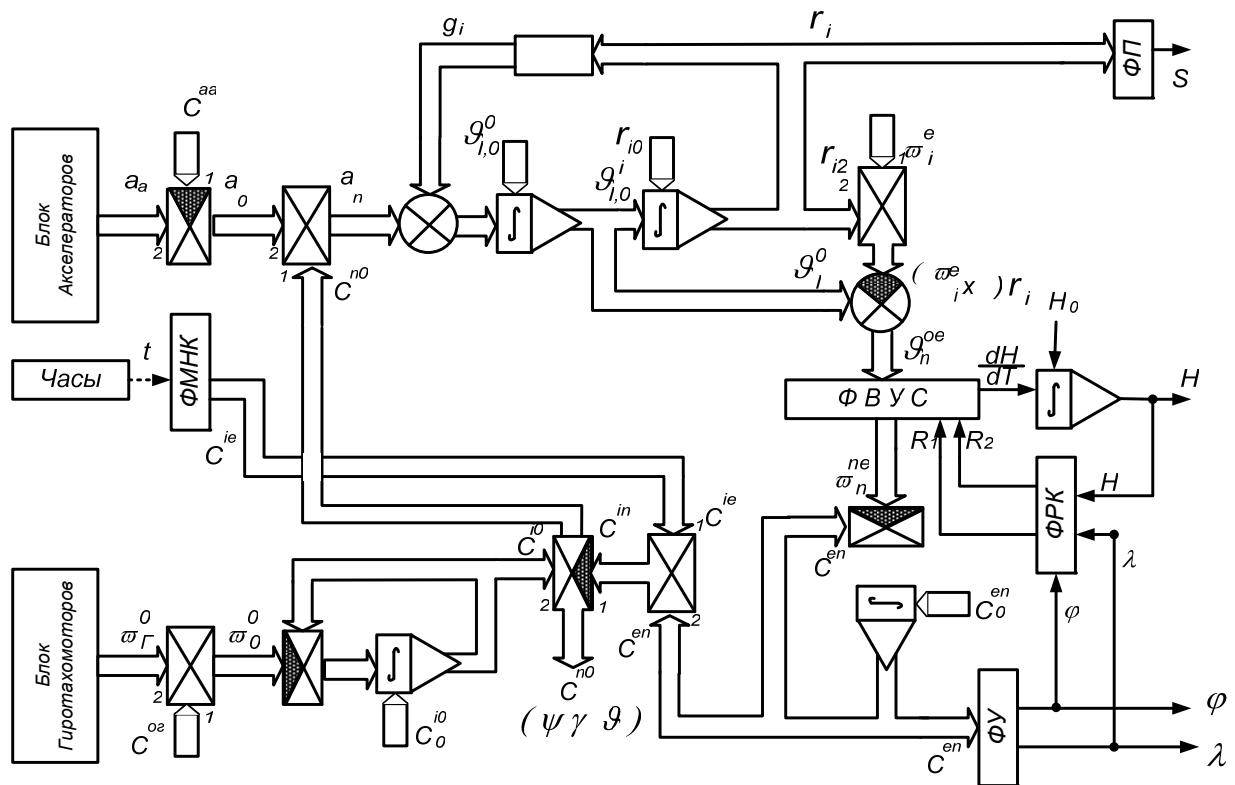


Рис. 1.5. Схема моделювального алгоритму 1

У відповідності зі схемою 1 (рис. 1.5), основне рівняння інерціальної навігації записується для вектора абсолютної швидкості, вираженого в проекціях на осі ІСВ:

$$\frac{dv_i^o}{dt} = C^{io} a_o + g_i, \quad v_i^o(t_o) = v_i^o, 0; \quad \frac{dr_i}{dt} = v_i^o, \quad r_i(t_o) = r_i, 0;$$

де C^{io} - матриця напрямних косинусів ОСВ відносно ІСВ; r_i - матриця-стовпець проєкцій радіуса-вектора центра мас об'єкта на осі ІСВ; v_i^o - матриця-стовпець проєкцій вектора v^o на осі ІСВ; a_o - матриця-стовпець проєкцій вектора удаваного прискорення на осі ОСВ.

У відповідності зі схемою 2 (рис. 1.6), основне рівняння інерціальної навігації записується для вектора відносної швидкості, вираженого в проекціях на осі НСВ:

$$\frac{dv_n^{oe}}{dt} = C^{no} a_o + g_n^* - [(\omega_n^n \times) + (\omega_n^e \times)] v_n^{oe}, \quad v_n^{oe}(t_o) = v_n^{oe}, 0$$

$$\frac{dr_n}{dt} = v_n^{oe} - (\omega_n^{ne} \times) r_n, \quad r_n(t_0) = r_n, 0$$

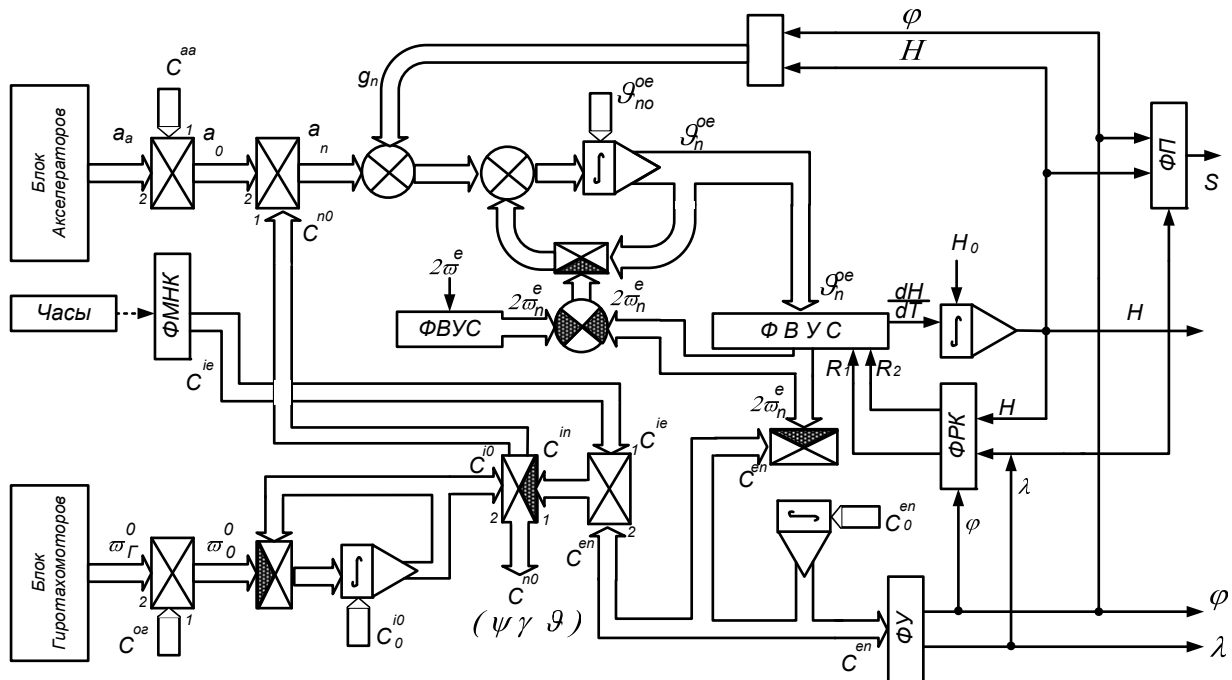


Рис.1.6. Схема моделювального алгоритму 2

Тут використані позначення: C^{no} – матриця напрямних косинусів ОСВ відносно НСВ; r_n – матриця-стовпець проєкцій радіуса-вектора центра мас об'єкта на осі НСВ; v_n^{oe} – матриця-стовпець проєкцій вектора v^{oe} на осі НСВ; g_n^* – матриця-стовпець проєкцій вектора сили тяжіння в точці перебування об'єкта; ω_n^n – матриця-стовпець проєкцій вектора абсолютної кутової швидкості НСВ на осі НСВ; ω_n^e – матриця-стовпець проєкцій вектора кутової швидкості обертання Землі на осі НСО; ω_n^{ne} – матриця-стовпець проєкцій вектора відносної кутової швидкості НСВ (щодо Землі) на осі НСВ.

Величини проєкцій кутових швидкостей визначаються (у випадку, якщо як НСВ обрати географічну систему відліку) співвідношеннями:

$$\omega_n^{ne} = \begin{bmatrix} \frac{v_{3n}^{oe}}{R_1}; & \frac{v_{3n}^{oe}}{R_1} \operatorname{tg} \varphi; & -\frac{v_{1n}^{oe}}{R_2} \end{bmatrix}^t; \quad (1.9)$$

$$\omega_n^e = \begin{bmatrix} \omega^e \cos \varphi; & \omega^e \sin \varphi; & 0 \end{bmatrix}^t; \quad (1.10)$$

$$\omega_n^n = \omega_n^{ne} + \omega_n^e.$$

1.4.2. Моделювальна схема 1

Замкнена система рівнянь, що дозволяє визначити задану сукупність вихідних змінних, для цієї схеми має вигляд:

$$\begin{aligned} \frac{dv_i^o}{dt} &= C^{io} a_o + g_i, & v_i^o, 0; \\ \frac{dr_i}{dt} &= v_i^o, & r_i, 0; \\ \frac{dC^{io}}{dt} &= C^{io} (\omega_o^o \times), & C^{io}, 0; \\ v_i^{oe} &= v_i^o - (\omega_i^e \times) r_i \\ \frac{dC^{en}}{dt} &= C^{en} (\omega_n^{ne} \times), & C^{en}, 0; & \longrightarrow \boxed{\varphi; \lambda} \\ C^{in} &= C^{ie} C^{en}, \\ C^{ni} &= (C^{in})^t, \\ v_n^{oe} &= C^{ni} v_i^{oe}, & \longrightarrow \boxed{v_{1n}^{oe}; v_{2n}^{oe}; v_{3n}^{oe}} \end{aligned}$$

де C^{ie} , ω_n^{ne} визначається із (1.4), (1.5), (1.9).

$$\begin{aligned} C^{no} &= C^{ni} C^{io}, & \longrightarrow \boxed{\psi, \theta, \gamma} \\ \frac{dH}{dt} &= \frac{v_{3n}^{oe}}{\kappa}, \quad H_0 & \longrightarrow \boxed{H} \end{aligned}$$

$$\kappa = \sqrt{1 - \varepsilon^2 \sin^2 \varphi};$$

$$R_1 = \frac{a + H}{\kappa};$$

$$R_2 = \frac{(a + H)(1 - \varepsilon^2)}{\kappa^3};$$

$$S = \sqrt{(r_{1i} - r_{1i}, 0)^2 + (r_{2i} - r_{2i}, 0)^2 + (r_{3i} - r_{3i}, 0)^2}; \quad \longrightarrow \boxed{S}$$

S – відстань від точки старту до точки поточного місця розташування об'єкта, а проєкції вектора g визначаються співвідношеннями (1.7).

1.4.3. Моделювальна схема 2

Замкнена система рівнянь у цьому випадку має вигляд:

$$\frac{dv_n^{oe}}{dt} = C^{no} a_o + g_n^* - [(\omega_n^n \times) + (\omega_n^e \times)] v_n^{oe}, \quad v_i^o, 0; \quad \longrightarrow \boxed{v_n^{oe}}$$

$$\frac{dC^{io}}{dt} = C^{io} (\omega_n^n \times), \quad C^{io}, 0;$$

$$\frac{dC^{en}}{dt} = C^{en} (\omega_n^{ne} \times), \quad C^{en}, 0; \quad \longrightarrow \boxed{\varphi; \lambda}$$

$$(\omega_n^n \times) + (\omega_n^e \times) - (\omega_n^{ne} \times) + 2(\omega_n^e \times),$$

$$C^{no} = C^{ne} C^{ei} C^{io}, \quad \longrightarrow \boxed{\psi, \theta, \gamma}$$

$$\boxed{H}$$

$$\frac{dH}{dt} = \frac{v_{3n}^{oe}}{\kappa}, \quad H_0 \quad \longrightarrow$$

$$\kappa = \sqrt{1 - \varepsilon^2 \sin^2 \varphi};$$

$$R_1 = \frac{a + H}{\kappa};$$

$$R_2 = \frac{(a + H)(1 - \varepsilon^2)}{\kappa^3};$$

$$(r_{1e}, r_{2e}, r_{3e}) = f(\varphi, \lambda, H);$$

$$S = \sqrt{(r_{1i} - r_{1e}, 0)^2 + (r_{2i} - r_{2e}, 0)^2 + (r_{3i} - r_{3e}, 0)^2}; \quad \longrightarrow \boxed{S}$$

S – відстань від точки старту до точки поточного місця розташування об'єкта, а проєкції вектора g^* визначаються співвідношеннями (1.8).

1.4.4. Порівняння моделювальних схем

Порівняння моделювальних схем показує, що друга схема має значні переваги за багатьма параметрами.

По-перше, вона потребує проведення значно меншої кількості обчислень. Через те, що використовується вектор сили тяжіння, немає необхідності в обчисленні r_n . Крім того, вектор g_n^* обчислюється за більш простими (ніж вектор g_i) залежностями. Необхідний в алгоритмі вектор v_n^{oe} у другій схемі визначається безпосередньо із розв'язку матричного диференціального рівняння.

По-друге, у другій схемі більш просте обчислення початкових умов, необхідних для розв'язку систем диференціальних рівнянь. Зазвичай початкове виставлення системи припускає одержання наступних параметрів:

$$(\psi_0, \theta_0, \gamma_0), \quad (\varphi_0, \lambda_0, H_0), \quad (v_{1n,0}^{oe}, v_{2n,0}^{oe}, v_{3n,0}^{oe})$$

Одержання початкових умов у двох варіантах алгоритмів вимагає наступних обчислень:

Схема 1	Схема 2
$(\varphi_0, \theta_0, \gamma_0) \rightarrow C_0^{no}$	$(\varphi_0, \theta_0, \gamma_0) \rightarrow C_0^{no}$
$(\varphi_0, \gamma_0) \rightarrow C_0^{en}$	$(\varphi_0, \gamma_0) \rightarrow C_0^{en}$
$C_0^{io} = C_0^{eo} = C_0^{en} C_0^{no}$	$C_0^{io} = C_0^{eo} = C_0^{en} C_0^{no}$
$v_{i,0}^{eo} = C_0^{en} v_{n,0}^{oe}$	$v_{n,0}^{oe}$ - задано
$v_{i,0}^o = v_{i,0}^{oe} + (\omega_i^e \times) r_{i,0}$	
$r_{i,0} = F(\varphi_0, \lambda_0, H_0)$	H_0 - задано

Удавана перевага першого варіанта полягає в тому, що в ньому можна виділити блок швидкого циклу:

$$\frac{dC^{io}}{dt} = C^{io}(\omega_o^o \times); \quad a_i = C^{io} a_o,$$

$$\frac{dv_i^o}{dt} = C^{io} a_o + g_i, \quad \frac{dr_i}{dt} = v_i^o,$$

і розв'язувати їх з високою частотою, а решту – частотою на порядок меншою.

В другій схемі, з урахуванням того, що перетворення $C^{no} a_o$ має бути у швидкому циклі, а, отже, C^{no} і весь алгоритм має працювати у швидкому циклі, здійснюється значно більший обсяг обчислень.

Однак цей недолік другої схеми легко усувається.

Дійсно, перетворення $C^{no} a_o$ можна записати у вигляді $C^{ni} C^{io} a_o$. Тоді $C^{io} a_o$ можна залишити у швидкому циклі, а $C^{ni} a_i$ перенести в повільний. Тоді друга схема фактично буде мати той же швидкий цикл, що і перша.

З огляду на це, візьмемо за моделювальну схему алгоритму другу схему з видозміненим блоком $C^{no} a_o$.

Для реалізації цієї моделювальної схеми необхідно вибрати чисельні методи й алгоритми розв'язку диференціальних рівнянь:

$$\frac{dC^{io}}{dt} = C^{io}(\omega_o^o \times), \quad \leftarrow C_0^{io}$$

$$\frac{dC^{en}}{dt} = C^{en}(\omega_n^{ne} \times), \quad \leftarrow C_0^{en}$$

$$\frac{dv_n^{oe}}{dt} = C^{no} a_o + g_n^* - [(\omega_n^n \times) + (\omega_n^e \times)] v_n^{oe}, \quad \leftarrow v_{n,0}^{oe}$$

$$\frac{dH}{dt} = \frac{v_{3n}^{oe}}{\kappa}, \quad \leftarrow H_0$$

1.5. Алгоритми розв'язування систем диференціальних рівнянь

1.5.1. Розв'язування задачі орієнтації

Алгоритм включає два блоки, що визначають значення матриць напрямних косинусів $C^{io}(t)$ і $C^{en}(t)$. Матриці можуть бути отримані безпосереднім розв'язуванням кінематичного рівняння Пуассона:

$$\frac{dC}{dt} = C(\omega \times), \quad \leftarrow C(t_0) = C_0$$

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

Однак цей шлях призводить до досить великих затрат машинного часу і не є оптимальним. Певний вигравш можна одержати, якщо як кінематичні параметри використовувати параметри Родріга-Гамільтона (кватерніони). У цьому випадку зменшується кількість диференціальних рівнянь. Однак ефект від цього зменшується через необхідність переходу від кватерніонів до матриці напрямних косинусів після розв'язку кінематичного рівняння. Найбільше поширення наразі набули чисельні методи розв'язування задачі орієнтації, які ґрунтуються на використанні координат вектора орієнтації. Суть методу полягає в наступному.

Розглянемо кутове положення приладового тригранника в два моменти часу t_{i-1} і $t_i = t_{i-1} + h$. Взаємне кутове положення тригранників \mathbf{E} і \mathbf{E}' можна описати за допомогою вектора орієнтації Φ . Напрямок цього вектора дає напрямок осі, повертаючи навколо якої базис \mathbf{E} можна одним плоским поворотом сумістити його з базисом \mathbf{E}' . Змінювання вектора орієнтації описується нелінійним диференціальним рівнянням:

$$\frac{d\Phi}{dt} = \omega - \frac{1}{2}(\omega \times)\Phi + \frac{1}{\Phi^2}\left(1 - \frac{\Phi}{2} \operatorname{tg} \frac{\Phi}{2}\right)(\Phi \times)\omega,$$

$$\Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{bmatrix}, \quad \omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix},$$

(Φ_1, Φ_2, Φ_3) – проєкції вектора Φ на базиси \mathbf{E} і \mathbf{E}' ; $(\omega_1, \omega_2, \omega_3)$ – проєкції вектора абсолютної кутової швидкості на базис \mathbf{E} .

На малих інтервалах часу h виконується умова $|\varphi| \ll 1$. Тоді нелінійний член рівняння, що має порядок $(\omega h)^3$, можна не враховувати і розглядати рівняння:

$$\frac{d\Phi}{dt} = \omega - \frac{1}{2}(\omega \times)\Phi.$$

З урахуванням інтегрального характеру первинної інформації, чисельні методи розв'язування цього рівняння доцільно будувати на основі методу послідовних наближень:

– нульове наближення: $\Phi^{(0)} = 0.$

– перше наближення: $\Phi^{(1)} = \int_{t_{i-1}}^{t_i} \omega(\tau) d\tau.$

– друге наближення:

$$\Phi^{(2)} = \int_{t_{i-1}}^{t_i} \omega(\tau) d\tau - \frac{1}{2} \int_{t_{i-1}}^{t_i} \left[(\omega(\tau') \times) \int_{t_{i-1}}^{\tau'} \omega(\tau) d\tau \right] d\tau'.$$

– третє наближення:

$$\Phi^{(3)} = \int_{t_{i-1}}^{t_i} \omega(\tau) d\tau - \frac{1}{2} \int_{t_{i-1}}^{t_i} \left[(\omega(\tau') \times) \int_{t_{i-1}}^{\tau'} \omega(\tau) d\tau \right] d\tau' + \\ + \frac{1}{4} \int_{t_{i-1}}^{t_i} \left\{ (\omega(\tau'') \times) \int_{t_{i-1}}^{\tau''} \left[(\omega(\tau') \times) \int_{t_{i-1}}^{\tau'} \omega(\tau) d\tau \right] d\tau' \right\} d\tau''$$

На практиці зазвичай використовують друге наближення. Апроксимуючи функцію $\omega(\tau)$ на інтервалі інтегрування кривими різного порядку, коефіцієнти яких визначаються за показаннями гіроскопів, одержують родину чисельних методів різної точності і різної кроковості. Наведемо алгоритми, що найбільш часто використовуються на практиці.

1. Однокрокові алгоритми (інтервал інтегрування дорівнює крокові h одержання інформації):

1) третього порядку точності:

$$\Phi_i = \Delta\theta_i + \frac{1}{12} (\Delta\theta_{i-1} \times) \Delta\theta_i, \quad \text{де } \Delta\theta_i = \begin{bmatrix} \Delta\theta_{1,i} \\ \Delta\theta_{2,i} \\ \Delta\theta_{3,i} \end{bmatrix};$$

$(\Delta\theta_1, \Delta\theta_2, \Delta\theta_3)$ – показання гіроскопів (див. (1.2)) у приростах кутів повороту об'єкта навколо осей чутливості гіротахometrів, якщо гіроскопи мають ортогональні осі чутливості, або результати приведення показань до ортогональних осей, якщо осі чутливості датчиків неортогональні; локальна похибка чисельного методу пропорційна величині $(\omega h)^4$;

2) четвертого порядку точності:

$$\Phi_i = \Delta\theta_i + \frac{1}{6} (\Delta\theta_{i-1} \times) \Delta\theta_i - \frac{1}{24} (\Delta\theta_{i-2} \times) \Delta\theta_i,$$

із похибкою, що пропорційна $(\omega h)^5$.

2. Двокроковий алгоритм (інтервал інтегрування дорівнює $2h$):

$$\Phi_i = \Delta\theta_{i-1} + \Delta\theta_i + \frac{2}{3} (\Delta\theta_{i-1} \times) \Delta\theta_i,$$

із похибкою, що пропорційна $(\omega h)^5$.

3. Чотирикроковий алгоритм (інтервал інтегрування дорівнює $4h$):

$$\Phi_i = \Delta\theta_{i-3} + \Delta\theta_{i-2} + \Delta\theta_{i-1} + \Delta\theta_i + \frac{4}{3} [(\Delta\theta_{i-3} \times) \Delta\theta_{i-1} + (\Delta\theta_{i-2} \times) \Delta\theta_i],$$

із похибкою, що пропорційна $(\omega h)^5$.

Вибір конкретного алгоритму вимагає додаткового дослідження і ретельного моделювання.

Після визначення Φ_i на інтервалі інтегрування обчислюється матриця напрямних косинусів ΔC_i , що описує перетворення $\mathbf{E} \rightarrow \mathbf{E}'$. Для цього використовується співвідношення:

$$\Delta C_i = \exp(\Phi_i) - I + (\Phi_i \times) + \frac{1}{2}(\Phi_i \times)^2 + \frac{1}{6}(\Phi_i \times)^3 + \dots$$

де I – одинична матриця.

Обмежуючись першими трьома членами ряду, одержуємо:

$$\Delta C_i = \begin{bmatrix} 1 - k_2(\Phi_{iy}^2 + \Phi_{iz}^2) & -k_1\Phi_{iz} + k_2\Phi_{ix}\Phi_{iy} & k_1\Phi_{iy} + k_2\Phi_{ix}\Phi_{iz} \\ k_1\Phi_{iz} + k_2\Phi_{ix}\Phi_{iy} & 1 - k_2(\Phi_{ix}^2 + \Phi_{iz}^2) & -k_1\Phi_{ix} + k_2\Phi_{iy}\Phi_{iz} \\ -k_1\Phi_{iy} + k_2\Phi_{ix}\Phi_{iz} & k_1\Phi_{ix} + k_2\Phi_{iy}\Phi_{iz} & 1 - k_2(\Phi_{ix}^2 + \Phi_{iy}^2) \end{bmatrix},$$

де:

$$k_1 = 1 - \frac{1}{6}|\Phi_i|^2; \quad k_2 = \frac{1}{2}\left[1 - \frac{1}{12}|\Phi_i|^2\right]; \quad |\Phi_i|^2 = \Phi_{ix}^2 + \Phi_{iy}^2 + \Phi_{iz}^2.$$

Матриця C_i визначається шляхом перемножування:

$$C_i = C_{i-1}\Delta C_i.$$

Отже, схема обчислень може бути представлена у вигляді:

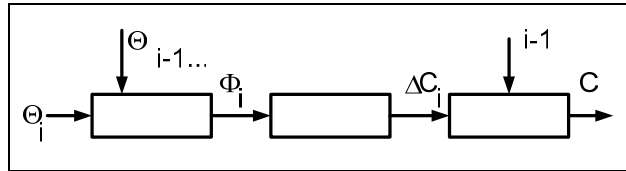


Рис. 1.7. Схема обчислень

1.5.2. Розв'язування задачі визначення відносної швидкості.

Відносна швидкість визначається шляхом розв'язування диференціального рівняння:

$$\frac{dv_n^{oe}}{dt} = C^{no} a_o + g_n^* - [(\omega_n^n \times) + (\omega_n^e \times)] v_n^{oe}, \quad \leftarrow v_{n,0}^{oe}$$

Або у вигляді системи рівнянь:

$$\frac{dv_{1n}^{oe}}{dt} = -(2\omega^e \sin \varphi + \omega_{2n}^{en}) v_{3n}^{oe} + \omega_{3n}^{en} v_{2n}^{oe} + a_{1n},$$

$$\frac{dv_{2n}^{oe}}{dt} = (2\omega^e \cos \varphi + \omega_{1n}^{en}) v_{3n}^{oe} + \omega_{3n}^{en} v_{1n}^{oe} + a_{2n} - g,$$

$$\frac{dv_{3n}^{oe}}{dt} = (2\omega^e \sin \varphi + \omega_{2n}^{en}) v_{1n}^{oe} - (2\omega^e \cos \varphi + \omega_{1n}^{en}) v_{2n}^{oe} + a_{3n},$$

де ω_n^{ne} визначаються по співвідношеннях (1.9).

Зважаючи на те, що $\omega_{1n}^{ne}, \omega_{2n}^{ne}, \omega_{3n}^{ne}$ – функції, що повільно змінюються в часі, обчислення $v_{n,i}^{oe}$ можна здійснювати, використовуючи $v_{n,i-1}^{oe}, \omega_{n,i-1}^{ne}$. Тоді:

$$v_{1n,i}^{oe} = v_{1n,i-1}^{oe} + \Delta V_{1n,i} - \omega_{1n,i-1}^{ne} v_{3n,i-1}^{oe} h_1 + \omega_{3n,i-1}^{ne} (v_{2n,i-1}^{oe} h_1 - 0,5gh_1^2);$$

$$v_{2n,i}^{oe} = v_{2n,i-1}^{oe} + \Delta V_{2n,i} + \omega_{2n,i-1}^{ne} v_{3n,i-1}^{oe} h_1 - \omega_{3n,i-1}^{ne} v_{1n,i-1}^{oe} h_1 - gh_1^2;$$

$$v_{3n,i}^{oe} = v_{3n,i-1}^{oe} + \Delta V_{3n,i} + \omega_{2n,i-1}^{ne} v_{1n,i-1}^{oe} h_1 - \omega_{1n,i-1}^{ne} (v_{2n,i-1}^{oe} h_1 - 0,5gh_1^2);$$

де $\Delta V_{1n}, \Delta V_{2n}, \Delta V_{3n}$ – прирости удаваної швидкості в проєкціях на осі навігаційної системи координат. Ці величини отримані в такий спосіб:

$$\Delta V_i = \sum_{k=1}^K C^{io} \Delta V_o; \quad \Delta V_n = C^{ni} \Delta V_i;$$

де $\Delta V_o = [\Delta V_{1o}, \Delta V_{2o}, \Delta V_{3o}]^t$ – величини, що вимірюються інтегровальними акселерометрами; K – кількість інтервалів швидкого циклу h в повільному циклі h_1 .

1.6. Висновки

Вибір моделювальної схеми головним чином визначається сукупністю вихідних параметрів системи. З огляду на те, що основними вихідними параметрами є параметри, що характеризують положення (координати, кути) і швидкості руху об'єкта щодо Землі, найкращою з погляду обсягу обчислення є моделювальна схема, заснована на обчисленні відносної швидкості в проєкціях на осі навігаційної системи координат. У цьому випадку найбільш просто обчислюються також початкові умови, необхідні для розв'язку систем диференціальних рівнянь і найбільш просто обчислюється прискорення сили тяжіння. Дана схема є аналогом полуаналітичної ІНС із ГСП, що реалізує географічний тригранник.

З частинних алгоритмів, що входять у моделювальну схему, основним є алгоритм визначення матриці напрямних косинусів між ортами зв'язаної і інерціальної систем координат. З безлічі можливих варіантів розв'язку цієї задачі зазвичай обирається варіант, у якому як проміжні параметри на малих інтервалах часу використовуються координати вектора орієнтації. При цьому кількість розв'язуваних диференціальних рівнянь всього три. Саме тому обсяг обчислень при реалізації цього алгоритму істотно менший, ніж при безпосередньому визначенні параметрів Родріга-Гамільтона або елементів матриці напрямних косинусів. Існують різні алгоритми, які реалізують даний метод, які відрізняються один від одного точністю й обсягом обчислень. Остаточний вибір конкретного алгоритму може бути здійснений на другому етапі після ретельного моделювання алгоритму в умовах руху, близьких до реального.

2. Оцінювання похибок визначення орієнтації

Вплив різних збурювальних факторів, що діють у реальних навігаційних системах, призводять до того, що обчислюване кутове положення об'єкта (отже, й пов'язаного з об'єктом тригранника E) відрізняється від істинного положення. Різницю між істинним і обчислюваним кутовими положеннями можна розглядати як похибку визначення орієнтації об'єкта навігаційною системою. Хоча якісно поняття похибки визначення орієнтації очевидно, однак методика кількісного оцінювання цієї похибки не є однозначною і потребує попереднього аналізу різних варіантів. Для аналізу будемо вважати, що орієнтація визначається матрицею напрямних косинусів.

Найбільш наочним способом опису похибки орієнтації (за аналогією з описом похибки скалярних величин) є визначення різниць:

$$\Delta C(t) = C(t) - C^*(t)$$

де $C(t)$ – матриця, отримана в результаті обчислень; $C^*(t)$ – істинна матриця.

Матриця $\Delta C(t)$ являє собою сукупність дев'яти величин, кожна з яких дорівнює різниці між величинами проєкцій ортів e_j обчисленої СК E на осі нерухомої СК I і ортів e_j^* істинної СК E^* на осі СК I .

Наявність дев'яти параметрів створює незручність кількісної оцінки похибки. Крім того, матриця ΔC не має чітко вираженого фізичного змісту в силу нелінійності різниць косинусів кутів (немає лінійної залежності між кутами й елементами матриці). З метою усунення перерахованих незручностей роботи з елементами матриці ΔC в літературі пропонувалися різні функції від цих елементів:

– норма матриці ΔC :

$$\|\Delta C\| = \left(\sum_{j,k=1}^3 |C_{jk}|^2 \right)^{\frac{1}{2}} \quad \text{або} \quad \|\Delta C\| = \max_j \left(\sum_{k=1}^3 |\Delta C_{jk}| \right),$$

– максимальний елемент матриці:

$$\Delta C_{\max} = \max_{j,k} (\Delta C_{jk}).$$

Однак ці критерії також ускладнюють розуміння фізичних причин похибок і можливість їхнього аналітичного оцінювання.

Способом опису похибки визначення орієнтації, який має чіткий фізичний зміст, є наступний:

– похибка визначається матрицею напрямних косинусів δC , що описує взаємну орієнтацію між обчисленою СК E і істинною СК E^* , тобто $\delta C = C_E^{E^*}$.

Матриця δC – це матриця, що описує перетворення СК E^* у СК E і має усі властивості матриць напрямних косинусів. Зв'язок між матрицями $C(t)$,

$C^*(t)$ і $\delta C(t)$ легко знайти, якщо перетворення систем координат $\mathbf{E} \rightarrow \mathbf{I}$ записати у вигляді послідовності двох перетворень $\mathbf{E} \rightarrow \mathbf{E}^* \rightarrow \mathbf{I}$. Тоді маємо:

$$C_I^E = C_I^{E^*} C_{E^*}^E,$$

або інакше

$$C = C^* \delta C.$$

Звідси одержуємо:

$$\delta C = C^{*T} C.$$

Через те, що ортогональне перетворення систем координат можна здійснити відповідно до теореми Ейлера за допомогою плаского повороту, то кількісно непогодженість між СК \mathbf{E} і СК \mathbf{E}^* можна оцінювати за допомогою тільки одного числового параметра – величини кута повороту Ейлера $\delta\chi$.

Величина $\delta\chi$ виражається через елементи матриці δC .

Із точністю до величин другого порядку мализни щодо кутів непогодженості маємо:

$$|\delta\chi| = (\delta\chi^T \cdot \delta\chi)^{\frac{1}{2}} = \sqrt{\delta C_{12}^2 + \delta C_{13}^2 + \delta C_{23}^2}.$$

З урахуванням перерахованих вище переваг, надалі як кількісну міру похибок визначення орієнтації будемо використовувати матрицю δC і параметр $|\delta\chi|$, що дорівнює куту, на який потрібно повернути обчислений тригранник E для його сполучення з істинним тригранником E^k . При цьому будемо пам'ятати про наступне обмеження – обчислена СК \mathbf{E} при цьому передбачається ортогональною. Якщо ж при обчисленнях порушуються умови ортогональності (відбувається деформація тригранника), то критерії δC і $|\delta\chi|$ можуть виявитися неправильними. При відсутності деформацій тригранників легко встановити зв'язок між двома зазначеними видами похибок $\Delta C(t)$ і $\delta C(t)$.

$$\delta C(t) = C^{*T}(t)C(t), \quad \Delta C(t) = C(t) - C^*(t).$$

Тоді, записуючи другий вираз у вигляді:

$$C(t) = C^*(t) + \Delta C(t),$$

і підставляючи його в перший вираз, одержуємо:

$$\delta C(t) = C^{*T}(t)(C^*(t) + \Delta C(t)) = C^{*T}C^* + C^{*T}\Delta C = E + C^{*T}\Delta C,$$

$$\delta C(t) = E + C^{*T}\Delta C, \quad \Delta C(t) = C^*(t)(\delta C - E).$$

2.1. Рівняння похибок орієнтації, обумовлені похибками вимірювання кутової швидкості

Орієнтація СК \mathbf{E} визначається розв'язком кінематичного рівняння:

$$\frac{dC}{dt} = C(\Omega \times), \quad C(t_0) = C_0. \quad (2.1)$$

При наявності похибки у кутовій швидкості $\Delta\omega = \bar{\omega} - \bar{\omega}^*$, де $\bar{\omega}$ - отримана в результаті вимірювання, а $\bar{\omega}^*$ істинна кутова швидкість, розв'язок $C(t)$ також буде відрізнятися від істинного значення на величину $\Delta C = C - C^*$.

Підставляючи $C = C^* + \Delta C$ й $(\Omega \times) = (\Omega^* \times) + (\delta\Omega \times)$ у рівняння (2.1), одержуємо:

$$\begin{aligned} \frac{dC^*}{dt} + \frac{d\Delta C}{dt} &= (C^* + \Delta C)((\Omega^* \times) + (\delta\Omega \times)) = \\ &= C^*(\Omega^* \times) + C^*(\delta\Omega \times) + \Delta C(\Omega^* \times) + \Delta C(\delta\Omega \times), \\ C_0 &= C_0^* + \Delta C_0. \end{aligned}$$

З огляду на те, що

$$\frac{dC^*}{dt} = C^*(\Omega^* \times), \quad C^*(t_0) = C_0^*,$$

одержуємо:

$$\frac{d(\Delta C)}{dt} = \Delta C(\Omega^* \times) + C^*(\delta\Omega \times) + \Delta C(\delta\Omega \times).$$

Припускаючи відхилення ΔC , $(\delta\Omega \times)$ малими, членом $\Delta C(\delta\Omega \times)$ можна знехтувати.

Таким чином, похибка ΔC описується диференціальним рівнянням:

$$\frac{d(\Delta C)}{dt} = \Delta C(\Omega^* \times) + C^*(\delta\Omega \times); \quad \Delta C(t_0) = \Delta C_0. \quad (2.2)$$

Рівняння (2.2) неоднорідне лінійне, тому його загальний розв'язок можна представити у вигляді суми загального розв'язку однорідного рівняння:

$$\frac{d(\Delta C)}{dt} = \Delta C(\Omega^* \times), \quad \Delta C(t_0) = \Delta C_0, \quad (2.3)$$

і часткового розв'язку неоднорідного рівняння (2.2).

Однорідне рівняння (2.3) має такий самий вигляд як і рівняння (2.1), тобто ці рівняння подібні. Тому розв'язок рівняння (2.3) можна виразити через розв'язок рівняння (2.1):

$$\Delta C(t) = PC^*(t),$$

де P – деяка постійна матриця, що визначається із початкових умов:

$$\Delta C_0 = PC_0^*.$$

Таким чином, загальний розв'язок рівняння (2.2) має вигляд:

$$P = \Delta C_0 C_0^{*T}.$$

Частковий розв'язок неоднорідного рівняння можна знайти методом варіації констант:

$$\Delta C = PC^* + PC^* = PC^* + PC^*(\Omega^* \times) = PC^* + \Delta C(\Omega^* \times).$$

Через те, що $\Delta C - \Delta C(\Omega^* \times) = C^*(\delta\Omega \times)$, то $PC^* = C^*(\delta\Omega \times)$.

Тому:

$$P = C^* (\delta\Omega \times) C^{*T}, \quad P = \int_0^t (C^* (\delta\Omega \times) C^{*T}) d\tau.$$

Частковий розв'язок неоднорідного рівняння дорівнює:

$$\Delta C_{\text{неодн}}(t) = \left[\int_0^t (C^* (\delta\Omega \times) C^{*T}) d\tau \right] C^*(t).$$

І загальний розв'язок неоднорідного рівняння дорівнює:

$$\Delta C(t) = \Delta C_0 C_0^{*T} C^*(t) + \left[\int_0^t (C^* (\delta\Omega \times) C^{*T}) d\tau \right] C^*(t).$$

Використовуючи зв'язок між $\Delta C(t)$ і $\delta C(t)$:

$$\Delta C(t) = C^*(t) [\delta C(t) - E],$$

одержуємо:

$$\begin{aligned} \delta C(t) &= E + C^{*T}(t) [\Delta C_0 C_0^{*T}] C^*(t) + C^{*T}(t) \left[\int_0^t (C^*(\tau) (\delta\Omega \times) (\tau) C^{*T}(\tau)) d\tau \right] C^*(t) = \\ &= E + C^{*T}(t) [C_0^* (\delta C_0 - E) C_0^{*T}] C^*(t) + C^{*T}(t) \left[\int_0^t (C^*(\tau) (\delta\Omega \times) (\tau) C^{*T}(\tau)) d\tau \right] C^*(t); \end{aligned} \quad (2.4)$$

$$\delta C(t) = C^{*T}(t) C_0^* \delta C_0 C_0^{*T} C^*(t) + C^{*T}(t) \left[\int_0^t (C^*(\tau) (\delta\Omega \times) (\tau) C^{*T}(\tau)) d\tau \right] C^*(t).$$

Незважаючи на громіздкий вигляд, вираз має чіткий фізичний зміст. Щоб його виявити, перейдемо до векторного представлення. Можна показати, що матричному перетворенню:

$$R_I = C^* R_E C^{*T}$$

де

$$R_I = \begin{vmatrix} 0 & -r'_3 & r'_2 \\ r'_3 & 0 & -r'_1 \\ -r'_2 & r'_1 & 0 \end{vmatrix}; \quad R_E = \begin{vmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{vmatrix},$$

$C^* = C_I^{*E}$ – матриця перетворення $\mathbf{E} \rightarrow \mathbf{I}$, відповідає векторне перетворення:

$$\bar{r}_I = C^* \bar{r}_E,$$

де

$$\bar{r}_I = \begin{vmatrix} r'_1 \\ r'_2 \\ r'_3 \end{vmatrix}, \quad \bar{r}_E = \begin{vmatrix} r_1 \\ r_2 \\ r_3 \end{vmatrix}.$$

Тоді, вираховуючи з лівої і правої частин рівняння (2.4.) \mathbf{E} (щоб привести матриці δC і δC_0 до кососиметричного вигляду з нульовими елементами по діагоналі), одержуємо:

$$\begin{aligned} \delta C(t) - E &= C^{*T}(t)C_0^*(\delta C_0 - E)C_0^{*T}C^*(t) + \\ &+ C^{*T}(t) \left[\int_0^t (C^*(\tau)(\delta\Omega \times)(\tau)C^{*T}(\tau))d\tau \right] C^*(t), \end{aligned}$$

і відповідне даному рівнянню векторне рівняння:

$$\delta\bar{\chi}_{E^*}(t) = C^{*T}(t)C_0^*(\delta\bar{\chi}_0)_{E^*} + C^{*T} \int_0^t C^*(\tau)\delta\bar{\omega}_{E^*}(\tau)d\tau, \quad (2.5)$$

де $\delta\bar{\chi}_{E^*} = \begin{pmatrix} \delta r_1 \\ \delta r_2 \\ \delta r_3 \end{pmatrix}$ – вектор похибок орієнтації в проекціях на осі СК E^* ;

$\delta\bar{\omega}_{E^*} = \begin{pmatrix} \delta\omega_1 \\ \delta\omega_2 \\ \delta\omega_3 \end{pmatrix}$ – похибки вектора кутової швидкості.

Вираз (2.5) інтегрується в такий спосіб. Вектор орієнтації $\delta\bar{\chi}(t)$, визначений у проекціях на осі рухомої СК, що описує похибку визначення орієнтації, складається з двох складових. Перша складова визначається похибками в початковий момент. Вираз $C^{*T}(t)C_0^*(\delta\bar{\chi}_0)_{E^*}$ визначає закони зміни в часі проекцій вектора $(\delta\bar{\chi}_0)_{E^*}$ на осі рухомої СК.

Друга складова визначається похибками вимірювання кутової швидкості. За допомогою оператора $C^*(\tau)$ вектор $\delta\bar{\omega}_E$ перетвориться у вектор $\delta\bar{\omega}_I$ (у нерухому СК), де виконується його інтегрування (нагромадження). Далі оператор $C^{*T}(t)$ перетворить отриманий після інтегрування вектор до осей рухомого тригранника.

Якщо похибку початкового виставлення покласти рівною нулю $\delta\bar{\chi}_0 = 0$, то рівняння має вигляд:

$$\delta\bar{\chi}_E(t) = C^{*T}(t) \int_0^t C^*(\tau)\delta\bar{\omega}_E(\tau)d\tau.$$

Отримані вирази являють собою рівняння похибок визначення орієнтації в інтегральній формі. При заданих $C^*(t)$ і $\delta\bar{\omega}(t)$ ці рівняння можуть бути розв'язані (принаймні в квадратурах).

Ці вирази можуть бути покладені в основу аналітичних досліджень впливу різних джерел похибок на точність визначення орієнтації. Для зручності доцільно знайти вираз і для $|\delta\bar{\chi}(t)|$, що є кількісною мірою похибки визначення орієнтації.

Позначимо:

$$\delta\bar{\chi}_I(t) = \int_0^t C^*(\tau)\delta\bar{\omega}_E(\tau)d\tau,$$

тоді:

$$\begin{aligned}
\delta\bar{\chi}_E^T \cdot \delta\bar{\chi}_E &= \delta\bar{\chi}_I^T \cdot \delta\bar{\chi}_I = \left[\int_0^t C^*(\tau) \delta\bar{\omega}(\tau) d\tau \right]^T \left[\int_0^t C^*(\tau) \delta\bar{\omega}(\tau) d\tau \right] = \\
&= \sum_{j=1}^3 \left[\int_0^t \sum_{i=1}^3 C_{ji}^* \delta\omega_i d\tau \right]^2 = \sum_{j=1}^3 \int_0^t \int_0^t \left(\sum_{i=1}^3 C_{ji}(\tau_1) \delta\omega_i(\tau_1) \right) \left(\sum_{i=1}^3 C_{jk}(\tau_2) \delta\omega_k(\tau_2) \right) d\tau_1 d\tau_2 = \\
&= \sum_{j=1}^3 \int_0^t \int_0^t \sum_{i=1}^3 \sum_{k=1}^3 \left[C_{ji}(\tau_1) C_{jk}(\tau_2) \delta\omega_i(\tau_1) \delta\omega_k(\tau_2) \right] d\tau_2 d\tau_1
\end{aligned}$$

Отже виходить:

$$|\delta\chi(t)| = \left\{ \sum_{j=1}^3 \int_0^t \int_0^t \sum_{i=1}^3 \sum_{k=1}^3 \left[C_{ji}(\tau_1) C_{jk}(\tau_2) \delta\omega_i(\tau_1) \delta\omega_k(\tau_2) \right] d\tau_2 d\tau_1 \right\}^{\frac{1}{2}}.$$

2.2. Джерела похибок орієнтації і приведення їх до еквівалентних похибок ЧЭ.

Розглянемо два варіанти системи визначення орієнтації $C_{E^*}^I$. У першому використовуються датчики кутової швидкості й АЦП для зв'язку датчиків на ЕОМ, у другому інтегруючи датчики з дискретним виходом. Джерела похибок орієнтації легко бачити зі схеми (рис.2.1):

- похибки датчиків кутової швидкості;
- похибки АЦП;
- методичні похибки алгоритму обчислення ΔC_n ;
- похибки початкового виставлення;
- похибки через обмеженість розрядної сітки ЕОМ.

До похибок датчиків можна віднести і похибки через відхилення осей чутливості датчиків від свого розрахункового положення.

Усі перераховані похибки можна привести до трьох груп, кожна з яких має свої закони зміни в часі і свої способи оцінювання:

- похибки кутової швидкості;
- похибки початкового виставлення;
- похибки ЕОМ (інструментальні).

Найбільша частина джерел похибок відноситься до першої групи. Уведемо поняття еквівалентної похибки датчиків.

При ідеальній роботі системи на вхід алгоритму розв'язування рівняння орієнтації надходить істинне значення кутової швидкості $\omega^*(t)$ (або $\Delta\theta^*(t)$ для інтегруючих датчиків). Вплив збурювального фактора призводить до того, що реальний $\omega(t)$ сигнал на вході алгоритму відрізняється від $\omega^*(t)$. Відхилення $\delta\omega(t) = \omega(t) - \omega^*(t)$, викликане впливом даного джерела збурень, будемо називати еквівалентною похибкою датчика. Таким чином, тракт перетворення вимірюваної величини вважаємо ідеальним, а єдиним джерелом похибок вважаємо датчик первинної інформації. Так як вплив різних джерел має різну фізичну

природу і свої закономірності, то необхідно попередньо вивчити способи переходу до еквівалентних похибок.

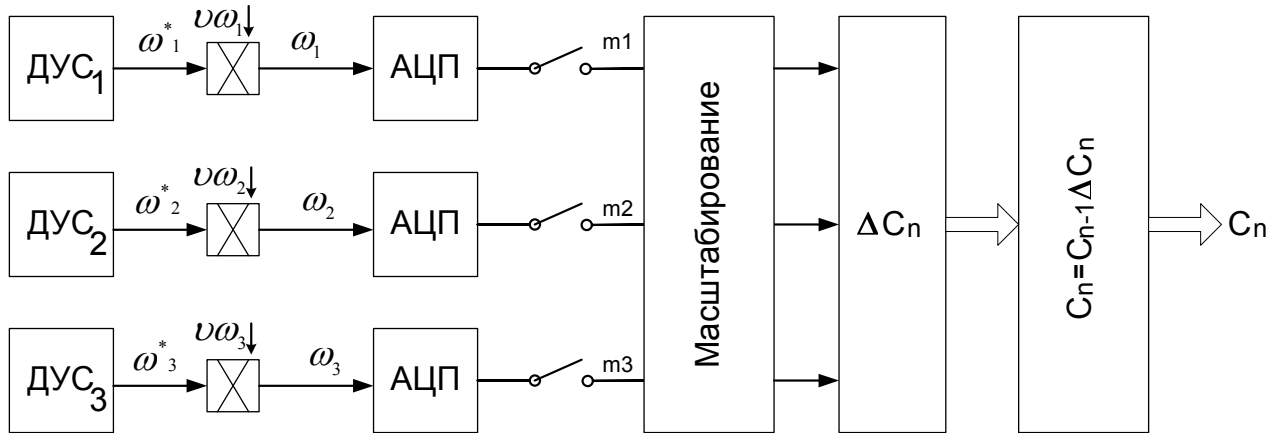


Рис. 2.1. Схема визначення орієнтації у БСО

2.2.1. Похибки через відхилення осей чутливості від розрахункових положень

Нехай $(\bar{e}_1^*, \bar{e}_2^*, \bar{e}_3^*)$ - розрахункові орти, уздовж яких по наших припущеннях спрямовані осі чутливості гіроскопів, а $(\bar{e}_1, \bar{e}_2, \bar{e}_3)$ - орти, що визначають реальне розташування осей чутливості. Положення ортів $(\bar{e}_1, \bar{e}_2, \bar{e}_3)$ відносно $(\bar{e}_1^*, \bar{e}_2^*, \bar{e}_3^*)$ будемо описувати матрицею напрямних косинусів:

$$\begin{aligned} \begin{pmatrix} \bar{e}_1 \\ \bar{e}_2 \\ \bar{e}_3 \end{pmatrix} &= \delta L \begin{pmatrix} \bar{e}_1^* \\ \bar{e}_2^* \\ \bar{e}_3^* \end{pmatrix} = \begin{pmatrix} \delta l_{11}\bar{e}_1^* + \delta l_{12}\bar{e}_2^* + \delta l_{13}\bar{e}_3^* \\ \delta l_{21}\bar{e}_1^* + \delta l_{22}\bar{e}_2^* + \delta l_{23}\bar{e}_3^* \\ \delta l_{31}\bar{e}_1^* + \delta l_{32}\bar{e}_2^* + \delta l_{33}\bar{e}_3^* \end{pmatrix} \\ \delta l_{ij} &= (\bar{e}_i \cdot \bar{e}_j^*) \end{aligned}$$

У загальному випадку матриця δL не ортогональна.

Вектор кутової швидкості $\bar{\omega}^*$ в проєкціях на осі $(\bar{e}_1^*, \bar{e}_2^*, \bar{e}_3^*)$ дорівнює:

$$\bar{\omega}^* = \omega_1^* \bar{e}_1^* + \omega_2^* \bar{e}_2^* + \omega_3^* \bar{e}_3^*$$

Реальні показання датчиків визначаються співвідношеннями:

$$\omega_1 = (\bar{e}_1 \cdot \bar{\omega}^*), \quad \omega_2 = (\bar{e}_2 \cdot \bar{\omega}^*), \quad \omega_3 = (\bar{e}_3 \cdot \bar{\omega}^*).$$

Підставляючи в ці вирази вектори \bar{e}_j і $\bar{\omega}^*$, записані в СК E , і враховуючи, що СК E^* ортогональна, одержуємо:

$$\begin{aligned} \omega_1 &= \delta l_{11}\omega_1^* + \delta l_{12}\omega_2^* + \delta l_{13}\omega_3^* \\ \omega_2 &= \delta l_{21}\omega_1^* + \delta l_{22}\omega_2^* + \delta l_{23}\omega_3^* \\ \omega_3 &= \delta l_{31}\omega_1^* + \delta l_{32}\omega_2^* + \delta l_{33}\omega_3^* \end{aligned}$$

Звідси одержуємо еквівалентні похибки датчиків, обумовлені невідповідністю осей чутливості датчиків своїм розрахунковим положенням:

$$\delta\omega_1 = \omega_1 - \omega_1^* + (\delta l_{11} - 1)\omega_1^* + \delta l_{12}\omega_2^* + \delta l_{13}\omega_3^*$$

$$\delta\omega_2 = \omega_2 - \omega_2^* + \delta l_{21}\omega_1^* + (\delta l_{22} - 1)\omega_2^* + \delta l_{23}\omega_3^*$$

$$\delta\omega_3 = \omega_3 - \omega_3^* + \delta l_{31}\omega_1^* + \delta l_{32}\omega_2^* + (\delta l_{33} - 1)\omega_3^*$$

У матричній формі маємо рівняння:

$$\delta\bar{\omega} = (\delta L - E)\bar{\omega}^*.$$

2.2.2. Похибки аналого-цифрового перетворювача .

Аналого-цифровий перетворювач (АЦП) перетворює напругу на виході гіроскопа, пропорційно проекції кутової швидкості, у двійковий код. Нехай $U^*(t)$ – сигнал на виході АЦП, x – ціна одного розряду АЦП. Тоді на виході АЦП маємо сигнал:

$$n(t) = \left[\frac{U^*(t)}{x} \right],$$

де [...] – ціла частина числа.

Кутова швидкість визначається в ЕОМ шляхом масштабування результату вимірювання отриманого в дискретний момент t_i :

$$\omega(t_i) = n(t_i)x.$$

Таким чином, у результаті аналого-цифрового перетворення одержуємо похибку, що еквівалентна похибці датчика:

$$\delta\omega(t_i) = \omega(t_i) - \omega^*(t_i) = x \left[\frac{U^*}{x} \right] - x \frac{U^*}{x} = -x \left(\frac{U^*}{x} - \left[\frac{U^*}{x} \right] \right) = -x \left\{ \frac{U^*}{x} \right\},$$

де $\left\{ \frac{U^*}{x} \right\}$ – дробова частина числа $\frac{U^*}{x}$.

У загальному випадку $\left\{ \frac{U^*}{x} \right\}$ є випадковою величиною, рівномірно розподіленою в інтервалі $[0,1]$, а, отже, $\delta\omega(t_i)$ рівномірно розподілена в інтервалі $[-x,0]$.

Математичне чекання $\delta\omega(t_i)$ дорівнює $H|\delta\omega| = -0,5x$.

2.2.3. Похибки алгоритмів визначення орієнтації.

Відповідно до описаної вище методики кожен чисельний алгоритм визначення орієнтації неявно містить у собі дві операції над результатами вимірів:

– відновлення (апроксимацію) функції $\bar{\omega}(t)$ на інтервалі інтегрування $(t_{n-1} \dots t_n)$ по її обчислених значеннях у дискретні моменти часу;

– інтегрування кінематичного рівняння, використовуючи отриману після апроксимації функцію $\bar{\omega}(t)$.

Обидві ці операції здійснюються наближеними методами, тому є джерелами методичних похибок визначення орієнтації. У першій операції джерелом похибок є похибки апроксимації. Ясно, що ця похибка зводиться до еквівалентної похибки датчиків:

$$\delta\omega(t) = \tilde{\omega}(t) - \omega^*(t),$$

де $\tilde{\omega}(t)$ – апроксимуюча функція.

Джерелом похибок другої операції є необхідність обмеження числа членів матрицанту (або його аналогів для інших кінематичних параметрів). У цьому випадку можливість приведення похибки до еквівалентної похибки датчиків не очевидна і задача вимагає додаткового дослідження (далі буде показано, що не приводиться).

Розглянемо похибку апроксимації кутової швидкості. На інтервалі інтегрування $(t_i \dots t_{i+1})$ кутова швидкість $\bar{\omega}^*(t)$ може бути подана рядом Тейлора:

$$\omega^*(\tau) = \omega_i^* + \dot{\omega}_i^* \tau + \frac{1}{2} \ddot{\omega}_i^* \tau^2 + \dots + \frac{1}{n!} \omega_i^{(n)*} \tau^n + \dots \tau = t - t_i.$$

У дійсності кутова швидкість апроксимується деякою кривою $\tilde{\omega}(\tau)$, яку можна представити у вигляді:

$$\tilde{\omega}(\tau) = \tilde{\omega}_i + \tilde{\omega}_i' \tau + \frac{1}{2} \tilde{\omega}_i'' \tau^2 + \dots + \frac{1}{n!} \tilde{\omega}_i^{(n)} \tau^n.$$

Тоді похибка апроксимації дорівнює:

$$\begin{aligned} \delta\omega(\tau) = \omega(\tau) - \tilde{\omega}(\tau) &= (\omega_i - \tilde{\omega}_i) + (\dot{\omega}_i - \tilde{\omega}_i') \tau + \frac{1}{2} (\ddot{\omega}_i - \tilde{\omega}_i'') \tau^2 + \dots + \\ &+ \frac{1}{n!} (\tilde{\omega}_i^{(n)} - \omega_i^{(n)*}) \tau^n - \frac{1}{(1+n)!} \cdot \omega_i^{(n+1)*} \tau^{n+1} - \dots = \sum_{j=0}^n \delta\tilde{\omega}_i \frac{\tau^j}{j!} - \sum_{j=n+1}^{\infty} \tilde{\omega}_i \frac{\tau^j}{j!}. \end{aligned}$$

Перша складова обумовлена похибками оцінки похідних кутової швидкості в момент t_i за результатами дискретних вимірів, а друга – обмеженням кількості членів ряду. Вид функції $\delta\omega(\tau)$ залежить від $\omega^*(t)$ і обчислюється для кожного конкретного випадку. Через те, що апроксимація кутової швидкості виконується не на всьому інтервалі $[0 \dots t]$ одночасно, а кусочно на інтервалах $(t_{i-1} \dots t_i)$, то функція $\delta\omega(t)$ має кусковий характер з можливими розривами першого роду на стиках.

3. Кінематичні рівняння орієнтації і загальні шляхи побудови алгоритмів їх чисельного інтегрування

Як раніше відзначалося, найбільш небезпечним джерелом похибок є «швидкий» канал БНС – канал визначення поточних значень кутів орієнтації об'єкта. Саме тут, через великий обсяг інформації, яка оброблюється за одиницю часу, можливо найбільш швидко нагромадження похибок. Тому канал орієнтації – безплатформова система орієнтації (БСО) – є основним об'єктом вивчення у цьому навчальному посібнику. Далі головною задачею буде дослідження похибок БСО, джерелом яких є сам процес чисельного визначення поточних значень кутів орієнтації об'єкта за відомими (вимірними) значеннями проєкцій кутової швидкості основи на осі системи координат, пов'язаної з цією основою. Найважливіша мета такого дослідження – встановити найбільш ефективний у пропонованих умовах (як із точки зору його простоти (мінімізації необхідних обчислювальних ресурсів), швидкодії, так і з погляду мінімуму можливих похибок, обумовлених обмеженими можливостями самого чисельного методу (так званих похибок методу)) метод (алгоритм) обробки вимірюваної інформації, що надходить.

До найбільш важливих характеристик методів (алгоритмів) чисельного розв'язування кінематичних рівнянь орієнтації з цієї точки зору відносяться:

- вид вимірювальної інформації, що надходить на вхід пристрою обробки інформації;
- затримка в обробці інформації;
- точність методу, яка характеризується зазвичай порядком методу;
- вид рівнянь орієнтації, до яких застосовується метод.

Задачу розробки алгоритмів чисельного інтегрування рівнянь орієнтації зручніше розглядати у спрощеному варіанті цих рівнянь, коли кутове положення рухомого об'єкта визначається безпосередньо в інерціальній системі координат. Це значно спрощує аналіз похибок і моделювання алгоритмів, а основні висновки, зроблені при цьому, повністю зберігають свою слушність і при більш опосередкованому завданні кутового положення об'єкта.

Нехай $\mathbf{I}(\mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3)$ – нерухома, а $\mathbf{E}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ – обертова пов'язана з рухомим об'єктом праві ортогональні системи координат (СК), де $\mathbf{i}_j, \mathbf{e}_j, (j = 1, 2, 3)$ їхні орти. Існують різні системи кінематичних параметрів, що описують орієнтацію СК \mathbf{E} відносно СК \mathbf{I} : напрямні косинуси, параметри Родріґа-Гамільтона, кути Ейлера-Крилова, координати вектора кінцевого повороту, параметри Кейлі-Клейна. Це різноманіття параметрів можна розділити на три принципово різні групи – кутові, матричні і векторні параметри.

3.1. Кутові параметри

До цієї групи кінематичних параметрів, що описують орієнтацію, відносять кути Ейлера-Крилова. Це подання засноване на тому, що довільний поворот твердого тіла навколо нерухомої точки можна подати у вигляді трьох послідовних плоских поворотів навколо трьох різних осей, що проходять через цю точку.

Кути Ейлера-Крилова введемо так, як це було у п. 1.4.1 при завданні положення об'єктові системи відліку відносно щодо навігаційної системи координат. Практично це означає, що для вирішення задачі розробки чисельних алгоритмів БСО задля її спрощення припускається, що Земля є нерухомою (навігаційна система координат є інерціальною). У відповідності з цим послідовність поворотів, що переводить інерціальну систему у об'єктову, приймемо такою (рис. 3.1):

- перший поворот навколо осі \bar{i}_2 на кут курсу $(-\psi)$;
- другий поворот навколо нового положення осі \bar{i}_3 на кут тангажа θ ;
- третій поворот навколо осі \bar{e}_1 на кут крена γ .

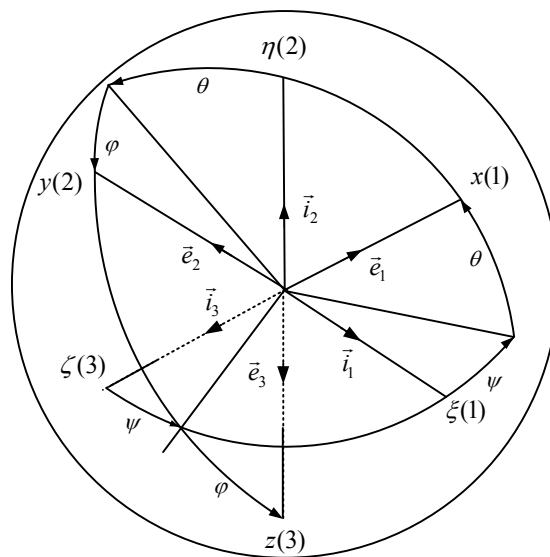


Рис. 3.1. Кути Ейлера-Крилова

Матриця перетворення C_I^E у цьому випадку має вигляд:

$$C_I^E = \begin{bmatrix} \cos\theta\cos\psi & -\sin\gamma\sin\psi - \cos\gamma\cos\psi\sin\theta & -\cos\gamma\sin\psi + \sin\gamma\cos\psi\sin\theta \\ \sin\theta & \cos\theta\cos\gamma & -\sin\gamma\cos\theta \\ \cos\theta\sin\psi & \sin\gamma\cos\psi - \cos\gamma\sin\psi\sin\theta & \cos\gamma\cos\psi + \sin\gamma\sin\psi\sin\theta \end{bmatrix}$$

Кінематичні рівняння для цих кутів Ейлера-Крилова мають вигляд:

$$\left\{ \begin{array}{l} \frac{d\psi}{dt} = -(\omega_2 \cdot \cos \gamma - \omega_3 \cdot \sin \gamma) / \cos \vartheta \\ \frac{d\theta}{dt} = \omega_3 \cdot \cos \gamma + \omega_2 \cdot \sin \gamma \\ \frac{d\gamma}{dt} = \omega_1 - (\omega_2 \cdot \cos \gamma - \omega_3 \cdot \sin \gamma) \operatorname{tg} \vartheta \end{array} \right. , \quad (3.1)$$

де ω_1 , ω_2 і ω_3 - проєкції вектора абсолютної швидкості обертання рухомого об'єкта на відповідні осі системи координат \mathbf{E} . Рівняння слухні, якщо $\varphi \neq (2k+1)\frac{\pi}{2}$.

3.2. Матричні параметри

При матричному способі орієнтація задається дев'ятьма параметрами c_{jk} ($j = 1, 2, 3$), що є косинусами кутів між ортами рухомої і нерухомої систем координат:

$$C_I^E = \left\| \begin{array}{ccc} (\mathbf{e}_1 \mathbf{i}_1) & (\mathbf{e}_2 \mathbf{i}_1) & (\mathbf{e}_3 \mathbf{i}_1) \\ (\mathbf{e}_1 \mathbf{i}_2) & (\mathbf{e}_2 \mathbf{i}_2) & (\mathbf{e}_3 \mathbf{i}_2) \\ (\mathbf{e}_1 \mathbf{i}_3) & (\mathbf{e}_2 \mathbf{i}_3) & (\mathbf{e}_3 \mathbf{i}_3) \end{array} \right\| = \|c_{jk}\|,$$

де $c_{jk} = (\mathbf{i}_j \cdot \mathbf{e}_k) = \cos(\mathbf{i}_j, \mathbf{e}_k)$.

Кінематичне рівняння, що описує змінювання матриці в часі в залежності від кутової швидкості обертання СК \mathbf{E} відносно СК \mathbf{I} має вигляд:

$$\frac{dC_I^E}{dt} = C_I^E (\boldsymbol{\Omega}_I^E \times)_E, \quad C_I^E(t_0) = C_0. \quad (3.2)$$

$$(\boldsymbol{\Omega}_I^E \times)_E = \left\| \begin{array}{ccc} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{array} \right\|, \quad (\omega_1, \omega_2, \omega_3)^T = (\boldsymbol{\omega}_I^E)_E$$

З огляду витрат ресурсів ЕОМ на розв'язування кінематичного рівняння матричний спосіб неекономічний. Це пов'язано з тим, що з дев'яти елементів матриці незалежними є тільки три. На елементи матриці накладені шість рівнянь зв'язку, обумовлені ортогональністю осей координат. Ці рівняння зв'язку мають вигляд:

$$\sum_{i=1}^3 c_{ij} c_{ik} = \sum_{i=1}^3 c_{ji} c_{ki} = \delta_{jk} = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases}.$$

3.3. Векторні параметри

Параметри Родріга-Гамільтона, Кейлі-Клейна, координати вектора орієнтації, координати вектора кінцевого повороту відносяться до групи векторних параметрів і усі вони ґрунтуються на відомій теоремі Ейлера:

- будь-який обертовий рух твердого тіла еквівалентний плоскому обертанню навколо деякої осі і може бути заданий скінченним поворотом навколо цієї осі.

З цієї теореми випливає, що орієнтація рухомої СК щодо нерухомої в будь-який момент може бути описана деяким одиничним вектором \mathbf{e} , перпендикулярним площині скінченного повороту, і кутом χ , на який здійснюється поворот. Різні кінематичні параметри, що відносяться до групи векторних, є деякими функціями \mathbf{e} і χ .

Вектор кінцевого повороту $\boldsymbol{\theta}$ по напрямку збігається з вектором Ейлера \mathbf{e} , по модулю дорівнює $2\operatorname{tg} \frac{\chi}{2} \mathbf{e}$.

Кінематичне рівняння, що описує змінювання вектора кінцевого повороту в часі, має вигляд:

$$\frac{d\boldsymbol{\theta}}{dt} = \boldsymbol{\omega} + \frac{1}{2}\boldsymbol{\theta} \times \boldsymbol{\omega} + \frac{1}{4}(\boldsymbol{\omega} \cdot \boldsymbol{\theta})\boldsymbol{\theta}; \quad \boldsymbol{\theta}(t_0) = \boldsymbol{\theta}_0.$$

Це рівняння нелінійне і вироджується при $\chi = (2k + 1)\pi$, ($k = 0, 1, 2$).

Як кінематичні параметри використовуються проекції (координати) вектора скінченного повороту на осі рухомої СК \mathbf{E} .

Вектор орієнтації $\boldsymbol{\chi}$ по напрямку також збігається з вектором Ейлера, але по величині дорівнює куту ейлерового повороту χ .

$$\boldsymbol{\chi} = \chi^e.$$

Кінематичне рівняння для $\bar{\boldsymbol{\chi}}$ має вигляд:

$$\frac{d\bar{\boldsymbol{\chi}}}{dt} = \boldsymbol{\omega} + \frac{1}{2}(\bar{\boldsymbol{\chi}} \times \boldsymbol{\omega}) + \frac{1}{\chi^2} \left(1 - \frac{1}{2} \cdot \frac{\chi \sin \chi}{1 - \cos \chi}\right) \bar{\boldsymbol{\chi}} \times (\bar{\boldsymbol{\chi}} \times \boldsymbol{\omega}); \quad \bar{\boldsymbol{\chi}}(t_0) = \bar{\boldsymbol{\chi}}_0.$$

Це рівняння нелінійне і вироджується при $\chi = 2\pi k$, ($k = 0, 1, \dots$).

Як кінематичні параметри використовують проекції вектора орієнтації на осі рухомої СК \mathbf{E} .

Параметри Родріга-Гамільтона, які складають собою власний кватерніон перетворення базису \mathbf{I} в базис \mathbf{E} , пов'язані з параметрами ейлерового повороту співвідношенням:

$$\Lambda = \cos \frac{\chi}{2} + \sin \frac{\chi}{2} \mathbf{e},$$

тобто скалярна частина кватерніона дорівнює $\lambda_0 = \cos \frac{\chi}{2}$, а векторна $\boldsymbol{\lambda} = \sin \frac{\chi}{2} \mathbf{e}$.

Кінематичне рівняння для параметрів Родріга-Гамільтона має вигляд:

$$\frac{d\Lambda}{dt} = \frac{1}{2} \Lambda \circ \bar{\boldsymbol{\omega}}_E, \quad (3.3)$$

$$\Lambda = \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix}; \quad \overline{\omega}_E = \begin{pmatrix} 0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}; \quad (\omega_1, \omega_2, \omega_3)^T = (\omega_I^E)_E.$$

◦ – позначка кватерніонного добутку, правило виконання якого визначається виразом:

$$N = \Lambda \circ M \Rightarrow \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} \lambda_0 \mu_0 - \lambda_1 \mu_1 - \lambda_2 \mu_2 - \lambda_3 \mu_3 \\ \lambda_0 \mu_1 + \lambda_1 \mu_0 + \lambda_2 \mu_3 - \lambda_3 \mu_2 \\ \lambda_0 \mu_2 + \lambda_2 \mu_0 + \lambda_3 \mu_1 - \lambda_1 \mu_3 \\ \lambda_0 \mu_3 + \lambda_3 \mu_0 + \lambda_1 \mu_2 - \lambda_2 \mu_1 \end{pmatrix}.$$

Кінематичне рівняння для параметрів Родріга-Гамільтона лінійне і не вироджується. На параметри накладене одне рівняння зв'язку:

$$\lambda_0^2 + \lambda_1^2 + \lambda_2^2 + \lambda_3^2 = 1.$$

3.4. Попередні висновки

З описаних кінематичних параметрів найбільший інтерес являють собою напрямні косинуси і параметри Родріга-Гамільтона. Це пов'язано насамперед з лінійністю їхніх кінематичних рівнянь і відсутністю яких-небудь обмежень на кутові рухи, при яких ці рівняння мали б особливості. З погляду витрат ресурсу ЕОМ перевагу мають параметри Родріга-Гамільтона. Виграш у порівнянні з напрямними косинусами досягає по оцінкам деяких авторів 30%. Однак при розрахунках варто враховувати одне істотне зауваження. Якщо інформація про орієнтацію використовується для перетворення вектора з однієї системи координат в іншу (а це й є головним призначенням каналу орієнтації), то ця операція у випадку напрямних косинусів вимагає значно менших витрат, ніж подвійний кватерніонний добуток, якщо знати параметри Родріга-Гамільтона. Тому зазвичай при необхідності роблять перехід від параметрів Родріга-Гамільтона до напрямних косинусів і після цього здійснюють перетворення вектора. Ясно, що при необхідності високої частоти перетворення вектора, переваги параметрів Родріга-Гамільтона можуть виявитися суто удаваними.

Окрім цих параметрів становлять інтерес координати вектора орієнтації. Хоча вони мають обмеження і є нелінійними, проте використання їх у якості проміжних при обчисленні напрямних косинусів і параметрів Родріга-Гамільтона може дати значний ефект.

З огляду на все це розглянемо далі чисельні методи розв'язування кінематичних рівнянь для напрямних косинусів, параметрів Родріга-Гамільтона. Для порівняння розглянемо також інтегрування кінематичних рівнянь у кутах Ейлера.

3.5. Особливості здійснювання чисельного інтегрування рівнянь орієнтації

Джерелом вимірювальної інформації в розглянутому випадку є гіроскопічні тахеометри – пристрої, що здійснюють вимірювання проєкцій кутової швидкості об'єкта ω на осі системи координат, жорстко пов'язаної з цим об'єктом. Тому, на перший погляд, вхідна (вимірювальна) інформація надходить на вхід пристрою обробки інформації у вигляді проєкцій $\omega_1, \omega_2, \omega_3$ кутової швидкості об'єкта. Однак у реальних умовах вихідна інформація з гіротахеометрів має частотно-модульований характер (частотний вихід). Здійснюючи вимірювання не частоти цього сигналу, а кількості імпульсів за час, рівний кванту часу вимірювання, можна одержати інформацію про інтеграли від зазначених проєкцій кутових швидкостей за час цього кванта вимірювання. Під тривалістю кванта вимірювання (кроком вимірювання, опитування) розумітимемо час між двома наступними вимірюваннями. Таким чином, у цьому випадку елементами вхідної інформації для пристрою обробки слід розуміти інтеграли за крок вимірювання від відповідних проєкцій кутових швидкостей:

$$\theta_{1,n} = \int_{t_{n-1}}^{t_n} \omega_1(\tau) d\tau, \quad \theta_{2,n} = \int_{t_{n-1}}^{t_n} \omega_2(\tau) d\tau, \quad \theta_{3,n} = \int_{t_{n-1}}^{t_n} \omega_3(\tau) d\tau.$$

Отже, можливі два види вимірювальної інформації, що надходить:

- проєкції $\omega_1, \omega_2, \omega_3$ кутових швидкостей об'єкта;
- інтеграли $\theta_1, \theta_2, \theta_3$ за крок вимірювання від цих проєкцій – прирости квазікоординат.

Точність методу чисельного інтегрування кінематичних рівнянь будемо характеризувати величиною показника ступеня при кроці обробки інформації у виразі залежності похибки методу від кроку.

З цієї позиції виділимо:

- 1) методи першого порядку – коли похибка обмеження відповідного методу прямо пропорційна кроку інтегрування;
- 2) методи другого порядку – похибка обмеження пропорційна квадрату кроку інтегрування;
- 3) методи третього порядку – похибка пропорційна кубові кроку;
- 4) методи четвертого порядку – похибка пропорційна четвертому ступеню кроку;
- 5) методи більш високого порядку за четвертий.

Головною задачею чисельного інтегрування кінематичних рівнянь орієнтації є одержання значень параметрів орієнтації у наступний момент часу на основі використання інформації про значення цих же параметрів у кілька попередніх моментів часу поділених інтервалами опитування. У подальшому інтервали опитування (вимірювання) ототожнюватимемо з кроком чисельного інтегрування.

Як відомо з теорії чисельного інтегрування диференціальних рівнянь, відомі дві групи методів чисельного інтегрування, які одержали назви однокрокових і багатокрокових. Усі ці методи використовують не виміряну інформацію

про поточні значення деяких величин, що входять в диференційні рівняння, а спираються на знання точних функцій правих частин диференційних рівнянь у формі Коши від змінних стану і часу. На основі цього знання розраховуються значення правих частин диференційних рівнянь у наступні моменти часу.

У випадку, коли залежності деяких величин у правих частинах диференційних рівнянь від часу є невідомими (як це є у розглядуваному випадку орієнтації рухомого об'єкту), таких підхід неможливий. Неможливо передбачити значення проекції кутової швидкості об'єкту у наступний момент часу. Тому у цьому випадку можливо розрахувати значення параметрів орієнтації лише у той момент часу, для якого вже здійснено вимірювання. Праві частини диференційних рівнянь у формі Коши можуть бути обчислені лише у дискретні моменти часу, які відповідають крокам вимірювання на основі вимірної інформації.

Однокрокові методи використовують знання функцій в правих частинах диференційних рівнянь задля визначення значень похідних від змінних стану у декількох наступних моментах часу, які зазвичай є частинами кроку інтегрування. Таке одержання інформації при поточному вимірюванні у дискретні моменти часу не уявляється можливим.

Багатокрокові ж методи використовують інформацію у декількох попередніх моментах часу, поділених однаковими інтервалами. Тому вони можуть бути застосовані і для розв'язування задачі чисельного інтегрування кінематичних рівнянь орієнтації на борті рухомого об'єкту.

Багатокрокові методи є методами прогнозу-корекції і складаються з двох етапів: прогнозу і корекції. На кожному з них розрахунки наступних значень змінних стану проводиться за різними формулами, деякі з яких приведені в таблиці 3.1. Алгоритм, що приведений як алгоритм четвертого порядку, зазвичай називають методом Адамса-Башфорта.

Таблиця 3.1. Формули багатокрокових методів прогнозу-корекції

Порядок методу	Формула прогнозу	Формула корекції
2	$y_{n+1} = y_{n-1} + 2h \cdot Z(t_n, y_n)$	$y_{n+1} = y_n + \frac{h}{2} [Z(t_{n+1}, y_{n+1}^*) + Z(t_n, y_n)]$
3	$y_{n+1} = y_n + \frac{h}{12} [23 \cdot Z(t_n, y_n) - 16 \cdot Z(t_{n-1}, y_{n-1}) + 5 \cdot Z(t_{n-2}, y_{n-2})]$	$y_{n+1} = y_n + \frac{h}{12} [5 \cdot Z(t_{n+1}, y_{n+1}^*) - 8 \cdot Z(t_n, y_n) - Z(t_{n-1}, y_{n-1})]$
4	$y_{n+1} = y_n + \frac{h}{24} [55Z(t_n, y_n) - 59Z(t_{n-1}, y_{n-1}) + 37Z(t_{n-2}, y_{n-2}) - 9Z(t_{n-3}, y_{n-3})]$	$y_{n+1} = y_n + \frac{h}{24} [9Z(t_{n+1}, y_{n+1}^*) + 19Z(t_n, y_n) - 5Z(t_{n-1}, y_{n-1}) + Z(t_{n-1}, y_{n-1})]$

Методи інтегрування відрізняються один від одного ще й у зв'язку з тією обставиною, що вони інтегрують різні рівняння. У випадку кінематичних рівнянь орієнтації, як було показано, об'єктами інтегрування можуть бути:

- кінематичні рівняння Ейлера;
- матричне рівняння орієнтації Пуассона (через матрицю напрямних косинусів);
- кватерніонне рівняння вектора орієнтації;
- рівняння змінювання вектора орієнтації.

У першому випадку інтегруються рівняння (3.1). При цьому вихідними величинами (які обчислюються безпосередньо) алгоритму є саме кути орієнтації ψ , θ і φ .

В другому випадку інтегрується матричне рівняння (рівняння Пуассона) (3.2):

$$\frac{dC}{dt} = C(\Omega \times),$$

$$\text{де } C \text{ – матриця напрямних косинусів і } (\Omega \times) = \begin{vmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{vmatrix}.$$

Вихідною величиною алгоритму є матриця C напрямних косинусів.

У третьому випадку інтегрується кватерніонне рівняння (3.3):

$$\frac{d\Lambda}{dt} = \frac{1}{2} \Lambda \circ \bar{\omega}_E,$$

де

$$\Lambda = \begin{vmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{vmatrix}, \quad \bar{\omega}_E = \begin{vmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{vmatrix},$$

\circ – позначка кватерніонного добутку. У ньому визначається кватерніон Λ , зв'язаний із шуканими кутами орієнтації відомими формулами.

В останньому випадку інтегрується векторне рівняння:

$$\frac{d\chi}{dt} = \omega + \frac{1}{2}(\chi \times \omega) + \frac{1}{\chi^2} \left(1 - \frac{1}{2} \cdot \frac{\chi \sin \chi}{1 - \cos \chi}\right) \chi \times (\chi \times \omega). \quad (3.4)$$

Виходом є проекції вектора орієнтації χ на осі, пов'язані з об'єктом.

У безплатформовій інерціальній навігаційній системі основною задачею безплатформової системи орієнтації є перепроеціювання на осі навігаційної системи відліку безпосередньо вимірюваних проекцій (на осі зв'язаної системи) абсолютного прискорення об'єкта. Для здійснення цієї операції в кожен момент часу необхідно знати матрицю C напрямних косинусів. Тому, основною задачею БСО в складі БНС є визначення поточного значення цієї матриці.

Звідси випливає, що тільки методи інтегрування матричного рівняння орієнтації вирішують безпосередньо задачу БСО в БНС. Всі інші методи ма-

ють бути доповнені операціями обчислення матриці C по обчислених вихідних параметрах відповідного алгоритму.

Зокрема, алгоритм, що використовує кінематичні рівняння Ейлера має бути доповнений обчисленням матриці C відповідно до формул:

$$C = \begin{bmatrix} \cos \theta \cos \psi & -\sin \gamma \sin \psi - \cos \gamma \cos \psi \sin \theta & -\cos \gamma \sin \psi + \sin \gamma \cos \psi \sin \theta \\ \sin \theta & \cos \theta \cos \gamma & -\sin \gamma \cos \theta \\ \cos \theta \sin \psi & \sin \gamma \cos \psi - \cos \gamma \sin \psi \sin \theta & \cos \gamma \cos \psi + \sin \gamma \sin \psi \sin \theta \end{bmatrix}. \quad (3.5)$$

Методи інтегрування кватерніонного рівняння (3.2) мають бути доповнені аналогічними обчисленнями по формулах:

$$C = \begin{bmatrix} \lambda_0^2 + \lambda_1^2 - \lambda_2^2 - \lambda_3^2 & 2(\lambda_1\lambda_2 - \lambda_0\lambda_3) & 2(\lambda_0\lambda_2 + \lambda_1\lambda_3) \\ 2(\lambda_0\lambda_3 + \lambda_1\lambda_2) & \lambda_0^2 + \lambda_2^2 - \lambda_1^2 - \lambda_3^2 & 2(\lambda_2\lambda_3 - \lambda_0\lambda_1) \\ 2(\lambda_1\lambda_3 - \lambda_0\lambda_2) & 2(\lambda_0\lambda_1 + \lambda_2\lambda_3) & \lambda_0^2 + \lambda_3^2 - \lambda_1^2 - \lambda_2^2 \end{bmatrix}. \quad (3.6)$$

4. Алгоритми методу прогноз-корекції

Для інтегрування рівнянь (3.1) – (3.4) у випадку, коли вихідними величинами вимірювачів є проекції кутових швидкостей і необхідно визначити кути орієнтації без затримки, можливе застосування класичних багатокрокових методів типу Адамса-Башфорта. Деяким ускладненням у цьому випадку є визначення значень кутів орієнтації на перших попередніх кроках при розгоні цих методів. Полегшуюча обставина – одержання значень проекцій кутових швидкостей на попередніх кроках при розгоні не дає ускладнень (вони безпосередньо вимірюються у попередні моменти часу).

4.1. Розробка алгоритмів

На відміну від постановки задачі чисельного інтегрування у чисельній математиці, де вважається заданими праві частини диференційних рівнянь як явні функції часу і шуканих змінних стану, у випадку вимірювання параметрів руху об'єкта на його борті праві частини визначаються лише у дискретні моменти часу як явні функції значень змінних стану у попередні моменти часу і неявні функції часу, якими є вимірювальні проекції кутової швидкості. В усьому решті чисельне інтегрування на борті повністю є ідентичним чисельному інтегруванню багатокроковими методами. Тому наведені у таблиці 3.1 методи можуть бути майже без змінювання застосовані на борті рухомого об'єкта (див. таблицю 4.1).

Таблиця 4.1. Формули методів прогнозу-корекції для БСО

Порядок методу	Формула прогнозу	Формула корекції
2	$y_{n+1} = y_{n-1} + 2h \cdot Z(\tilde{\omega}_n, y_n)$	$y_{n+1} = y_n + \frac{h}{2} [Z(\tilde{\omega}_{n+1}, y_{n+1}^*) + Z(\tilde{\omega}_n, y_n)]$
3	$y_{n+1} = y_n + \frac{h}{12} [23 \cdot Z(\tilde{\omega}_n, y_n) - 16 \cdot Z(\tilde{\omega}_{n-1}, y_{n-1}) + 5 \cdot Z(\tilde{\omega}_{n-2}, y_{n-2})]$	$y_{n+1} = y_n + \frac{h}{12} [5 \cdot Z(\tilde{\omega}_{n+1}, y_{n+1}^*) - 8 \cdot Z(\tilde{\omega}_n, y_n) - Z(\tilde{\omega}_{n-1}, y_{n-1})]$
4	$y_{n+1} = y_n + \frac{h}{24} [55Z(\tilde{\omega}_n, y_n) - 59Z(\tilde{\omega}_{n-1}, y_{n-1}) + 37Z(\tilde{\omega}_{n-2}, y_{n-2}) - 9Z(\tilde{\omega}_{n-3}, y_{n-3})]$	$y_{n+1} = y_n + \frac{h}{24} [9Z(\tilde{\omega}_{n+1}, y_{n+1}^*) + 19Z(t_n, y_n) - 5Z(t_{n-1}, y_{n-1}) + Z(t_{n-1}, y_{n-1})]$

Як бачимо, єдиною особливістю і відмінністю від класичних методів прогноз-корекції є те, що замість часу у правих частинах рівнянь фігурує вимірю-

ваний вектор $\tilde{\omega}$, який складається з трьох його проекцій на осі об'єктові системи координат.

З необхідності здійснення корекції також впливає, що здійсненність методів прогноз корекції можлива лише за умови, що є вимірним вектор $\tilde{\omega}$ у той момент часу, для якого потрібно визначити параметри орієнтації.

4.2. Мета моделювання і засоби її досягнення

Головною метою моделювання алгоритмів методів чисельного розв'язування рівнянь орієнтації є відшукування похибок визначення параметрів орієнтації, перевірка їх залежності від кроку інтегрування, а також визначення засобів зменшення похибок алгоритмів, їх стійкості, обсягу потрібних обчислювальних ресурсів.

Для визначення похибок програма моделювання має забезпечити:

- 1) розрахунок "точних" значень параметрів орієнтації у відповідні моменти часу, тобто імітування руху об'єкта;
- 2) імітування роботи бортових вимірювачів проекцій кутової швидкості об'єкта;
- 3) розрахунок на кожному кроці вимірювання (чисельного інтегрування) нових (на наступному кроці) значень параметрів орієнтації за досліджуваними алгоритмами;
- 4) обчислювання різниці між "точними" і одержаними внаслідок чисельного інтегрування значеннями параметрів орієнтації (похибок алгоритмів) і приведення їх до уніфікованої форми;
- 5) виведення розрахованих значень похибок у графічній формі.

У подальшому обмежимося вивченням лише похибок, обумовлених самими алгоритмами, тому при моделюванні вважатимемо похибки вимірювання відсутніми. Інакше кажучи, припускатимемо, що вимірні значення проекцій повністю збігаються з істинними значеннями відповідних проекцій кутової швидкості основи.

Програмне моделювання, яке подається далі, розроблене для середовища Matlab мовою цього середовища.

4.2.1. Процедури обчислення параметрів поточного руху основи

"Генерування" кутового руху об'єкта і формування як значень параметрів орієнтації у дискретні моменти модельного часу, так й значень проекцій кутової швидкості, забезпечується у подальшому за допомогою процедури *UgDvObm.m*. Ця процедура за заданим значенням моменту часу обчислює поточні значення кутів ψ , ϑ і γ . При цьому припускається, що кути повороту об'єкта у загальному випадку змінюються у часі за законами:

$$\psi(t) = \dot{\psi}_o t + \psi_m \cdot \sin(\omega_\psi t + \varepsilon_\psi);$$

$$\vartheta(t) = \dot{\vartheta}_o t + \vartheta_m \cdot \sin(\omega_\vartheta t + \varepsilon_\vartheta);$$

$$\gamma(t) = \dot{\gamma}_o t + \gamma_m \cdot \sin(\omega_\gamma t + \varepsilon_\gamma),$$

де $\dot{\psi}_o$, $\dot{\vartheta}_o$, $\dot{\gamma}_o$ – сталі складові похідних за часом від відповідних кутів; ψ_m , ϑ_m , γ_m – амплітуди гармонійних складових змінювання відповідних кутів; ω_ψ , ω_θ , ω_γ – колові частоти, а ε_ψ , ε_ϑ , ε_γ – початкові фази цих коливань. Усі вказані сталі величини мають бути задані в основній програмі і сформовані у єдиний масив D(12), а останній об'явлений як глобальний.

Окрім того, обчислюються похідні від кутів за формулами:

$$\dot{\psi}(t) = \dot{\psi}_o + \psi_m \omega_\psi \cdot \cos(\omega_\psi t + \varepsilon_\psi);$$

$$\dot{\vartheta}(t) = \dot{\vartheta}_o + \vartheta_m \omega_\vartheta \cdot \cos(\omega_\vartheta t + \varepsilon_\vartheta);$$

$$\dot{\gamma}(t) = \dot{\gamma}_o + \gamma_m \cdot \cos(\omega_\gamma t + \varepsilon_\gamma).$$

Нарешті, на основі цих обчислень, розраховуються поточні значення проєкції кутової швидкості на об'єктові осі за формулами

$$\omega_1(t) = \dot{\gamma}(t) - \dot{\psi}(t) \sin \vartheta(t);$$

$$\omega_2(t) = \dot{\vartheta}(t) \sin \gamma(t) - \dot{\psi}(t) \cos \vartheta(t) \cdot \cos \gamma(t);$$

$$\omega_3(t) = \dot{\vartheta}(t) \cos \gamma(t) + \dot{\psi}(t) \cos \vartheta(t) \cdot \sin \gamma(t).$$

Текст процедури поданий нижче:

```
function x=UgDvObm(t)
% Процедура обчислення параметрів поточного кутового руху ОСНОВИ
%
% Вхідні параметри
%   t - поточний час;
%   Kt, Km, omK, eK - параметри руху з кута курсу
%   ( psi(t)= Kt*t + Km*sin(omK*t+eK) );
%   Tt, Tm, omT, eT - параметри руху з кута тангажа teta
%   ( teta(t) = Tt*t + Tm*sin(omT*t+eT) );
%   Gt, Gm, omG, eG - параметри руху з кута крена gamma
%   ( gamma(t) = Gt*t + Gm*sin(omG*t+eG) );
%
% Выходные параметры
%   x(1) - кут курсу (psi); x(2) - кут тангажа (teta); x(3) - кут крена (gamma);
%   x(4) - кутова швидкість курсу; x(5) - тангажа; x(6) - крена;
%   x(7) - проєкція кутової швидкості на вісь X; x(8) - на вісь Y; x(9) - на вісь Z;
%   OMx = x(7) = x(6) - x(4)*sin(x(2))
%   OMy = x(8) = - x(4)*cos(x(2))*cos(x(3))+ x(5)*sin(x(3))
%   OMz = x(9) = x(4)*cos(x(2))*sin(x(3))+ x(5)*cos(x(3))
%   x(10) - проєкція кутового прискорення на ось X; x(11) - на вісь Y; x(12) - на вісь Z;

% Лазарев Ю. Ф. 2-03-2006
global D2
for i=1:3,
    f(i)=D2(i+6)*t+D2(i+9);
    x(i)=D2(i)*t+D2(i+3)*sin(f(i)); % Обчислення кутів
    x(i+3)=D2(i)+D2(i+6)*D2(i+3)*cos(f(i)); % Обчислення кутових швидкостей
end
% Обчислення проєкцій кутової швидкості
sg=sin(x(3)); cg=cos(x(3)); st=sin(x(2)); ct=cos(x(2));
x(7) = x(6) - x(4)*st;
x(8) = - x(4)*ct*cg+ x(5)*sg;
x(9) = x(4)*ct*sg+ x(5)*cg;
% Кінець процедури
```

4.2.2. Процедури методів прогноз-корекції

Для того, щоб промоделювати поведження трьох алгоритмів прогноз-корекції, формули яких наведені у таблиці 4.1, створені процедури-функції *prkr2*, *prkr3* і *AdBash*, які втілюють ці алгоритми. Тексти їх наведені нижче.

```
function yout = prkr2(yfun,h,t,y1,y,omo1,omo,om)
%   Интегрування Кінематичних рівнянь Безплатформової Системи Орієнтації,
%   праві частини яких задані процедурою урfun, методом прогноз-корекції 2-го порядку
%   при вимірюванні проєкцій кутової швидкості об'єкта
%   Вхідні змінні:
%   h   - крок інтегрування;
%   t   - поточний момент часу;
%   y   - вектор значень фазових змінних у поточний момент часу t
%   y1  - вектор значень фазових змінних у попередній момент часу t-h
%   omo - вектор вимірюваних значень проєкцій кутової швидкості у
%   поточний момент часу t
%   omo1 - вектор вимірюваних значень проєкцій кутової швидкості у
%   попередній момент часу t-h
%   om  - вимірюваних значень проєкцій кутової швидкості у
%   наступний момент часу t+h
%   урfun - рядок, який містить ім'я п/п правих частин ЗДР БСО.
%   Звернення: z = fun(t,y,omo), де урfun = 'fun'.
%   t   - поточний час
%   y   - поточне значення вектора фазових змінних
%   omo - поточне вимірюване значення вектора кутової швидкості об'єкта
%   z   - обчислені значення похідних z(i) = dy(i)/dt.
%   Вихідні змінні:
%   yout - обчислене значення вектора фазових змінних у момент часу t+h

%   Y. F. Lazarev,   11-03-2006
%   П Р О Г Н О З
%   Розрахунок значення похідних від к фазових змінних для поточного моменту часу t
z = feval(yfun, t, y,omo);
%   Розрахунок прогнозованого значення вектора фазових змінних
%   у наступний момент часу t+h
ур=y1 + 2*h*z;
%   К О Р Е К Ц І Я
for i=1:4
    zp=feval(yfun, t+h, ур,om);
    ур=у + h/2*(zp+z);
end;
%   Розрахунок остаточних значень вектора фазових змінних
%   у наступний момент часу t+h
yout = ур;
=====

function yout = prkr3(yfun,h,t,y2,y1,y,omo2,omo1,omo,om)
%   Интегрування Кінематичних рівнянь Безплатформової Системи Орієнтації,
%   праві частини яких задані процедурою урfun, методом прогноз-корекції 3-го порядку
%   при вимірюванні проєкцій кутової швидкості об'єкта
%   Вхідні змінні:
%   h   - крок інтегрування;
%   t   - поточний момент часу;
%   y   - вектор значень фазових змінних у поточний момент часу t
%   y1  - вектор значень фазових змінних у попередній момент часу t-h
%   y2  - вектор значень фазових змінних у попередній момент часу t-2h
%   omo - вектор вимірюваних значень проєкцій кутової швидкості у
%   поточний момент часу t
%   omo1 - вектор вимірюваних значень проєкцій кутової швидкості у
%   попередній момент часу t-h
```

```

%      omo2 - вектор виміряних значень проекцій кутової швидкості у
%      попередній момент часу t-2h
%      om   - виміряних значень проекцій кутової швидкості у
%      наступний момент часу t+h
%      ypfun - рядок, який містить ім'я п/п правих частин ЗДР БСО.
%      Звернення: z = fun(t,y,omo), де ypfun = 'fun'.
%      t    - поточний час
%      y    - поточне значення вектора фазових змінних
%      omo  - поточне виміряне значення вектора кутової швидкості об'єкта
%      z    - обчислені значення похідних z(i) = dy(i)/dt.
%      Вихідні змінні:
%      yout - обчислене значення вектора фазових змінних у момент часу t+h

%      Yu. F. Lazarev,   11-03-2006
%      П Р О Г Н О З
%      Расчет промежуточных значений производных
z = feval(ypfun, t, y,omo);
z1 = feval(ypfun, t-h, y1,omo1);
z2 = feval(ypfun, t-2*h, y2,omo2);
%      Расчет предсказанного значения
yp=y + h/12*(23*z-16*z1+5*z2);
%      К О Р Р Е К Ц И Я
for i=1:4,
    zp=feval(ypfun, t+h, yp,om);
    yp=y + h/12*(5*zp+8*z-z1);
end;
%      Расчет новых значений вектора фазовых переменных
yout = yp;
=====

function yout = AdBash(ypfun,h,t,y3,y2,y1,y,omo3,omo2,omo1,omo,om)
%      Интегрирование ОДУ правые части которых заданы
%      процедурой ypfun, методом Прогноза-коррекции 4-го порядка
%      (метод Адамса-Башфорта)
%      (по четырем предыдущим точкам t).
%      Входные переменные:
%      h      - шаг интегрирования;
%      t      - предшествующий момент времени.
%      y3,y2,y1,y  - векторы значений фазовых переменных
%                  в четырех предшествующих точках.
%      omo3,omo2,omo1,omo - векторы измеренных значений
%                  в четырех предшествующих точках.
%      om - вектор измеренных значений в последующей точке.
%      ypfun - Строка, содержащая имя п/п правых частей ОДУ.
%      Вызов: z = fun(t,y,omo), где ypfun = 'fun'.
%      t      - Текущее время.
%      y      - текущее значение вектора фазовых переменных.
%      omo    - текущее измеренное значение .
%      z      - рассчитанные значения производных; z(i) = dy(i)/dt.
%      Выходные переменные:
%      yout - Расчитанное значение вектора фазовых переменных
%            через шаг интегрирования.

%      Y. F. Lazarev, 274-67-31.
%      Copyright (c) 1998 by The PSON, NTU of Ukraine "KPI" .
%      П Р О Г Н О З
%      Расчет промежуточных значений производных
z = feval(ypfun, t, y,omo);
z1 = feval(ypfun, t-h, y1,omo1);
z2 = feval(ypfun, t-2*h, y2,omo2);
z3 = feval(ypfun, t-3*h, y3,omo3);
%      Расчет предсказанного значения

```

```

ур=y + h/24*(55*z - 59*z1 + 37*z2 -9*z3);

% КОРРЕКЦІЯ
for i=1:4,
    zp=feval(yfun, t+h, ур,om);
    ур=y + h/24*(9*zp + 19*z - 5*z1 + z2);
end;
% Расчет новых значений вектора фазовых переменных
уout = ур;

```

4.2.3. Процедури правих частин рівнянь орієнтації

Усі подані у цьому поділі алгоритми розраховані на застосування вимірювачів, які видають інформацію про значення проекцій кутової швидкості основи. Алгоритми апробовані на трьох видах рівнянь орієнтації

- 1) кінематичних рівняннях Ейлера;
- 2) матричному рівнянні Пуассона;
- 3) кватерніонному рівнянні.

Задля цього були створені процедури-функції *KUE*, *Poisson* і *KRKvat* обчислення правих частин диференційних рівнянь цих трьох видів, тексти яких наводяться далі. Вхідними величинами цих процедур є значення часу, змінних стану і вектора кутової швидкості у цей момент часу. Вихідна величина - вектор (матриця) значень правих частин відповідних рівнянь орієнтації

```

function z=KUE(t,y,omo)
% Процедура вычисления правых частей Кинематических Уравнений Эйлера
% psi' = [omz*sin(gama) - оmy*cos(gama)]/cos(teta);
% teta' = omz*cos(gama) - оmy*sin(gama);
% gama' = omx+[omz*sin(gama) - оmy*cos(gama)]/tg(teta);
% y(1) = psi; y(2) = teta; y(3) = gama
% оmo(1) = omx; оmo(2) = оmy; оmo(3) = omz;

```

```

% Лазарев Ю. Ф. 1-06-2006
omm=omo(3)*sin(y(3))-omo(2)*cos(y(3));
z(1)= omm/cos(y(2));
z(2)=omo(3)*cos(y(3))+omo(2)*sin(y(3));
z(3)=omo(1)+omm*tan(y(2));
=====

```

```

function z = Poisson(t,y,om)
% Програма z = Poisson(t,y,om) обчислює праві частини "z"
% матричних рівнянь Пуассона:
% z = y*(om*),
% де "y" - матриця напрямних косинусів;
% (om*) - кососиметрична матриця, яка відповідає
% вектору-рядку "om" проекцій кутової швидкості.

```

```

% Ю.Ф.Лазарев, 22 вересня 1998р.
ksom = vect2ksm(om);
z = y*ksom;
=====

```

```

function z = KRKvat(t,Qv,om)
% Програма z = KRKvat(t,Qv,om) обчислює кватерніон правої частини "z"
% кінематичних рівнянь у кватерніонах:

```

```

%          z = 1/2*Qv*omkv,
% де Qv - кватерніон кутової орієнтації;
% omkv- вектор кватерніон відповідний до вектора кутової швидкості.

% Ю.Ф.Лазарев, Бобровицька Я. 13 березня 2006р
% 1-06-2006
V=[0 om];
omkv=quatern(V);
z = Qv*omkv/2;

```

4.3. Опис програмних засобів

Керуюча програма *PK_KUE_om_upr.m* ініціює роботу сукупності програмних одиниць, яка забезпечує обчислення похибок чисельного інтегрування кінематичних рівнянь Ейлера трьома алгоритмами методу прогнозу-корекції – другого, третього і четвертого порядків.

Використовуються наступні процедури й функції:

- 1) *idbsosn1.m* – процедура змінювання значень вхідних даних для програми *PK_KUE_om_upr.m*;
- 2) *ugdvoibm.m* – функція обчислення параметрів поточного кутового руху основи (кутів, кутових швидкостей, кутових прискорень);
- 3) *prkr2.m* – процедура чисельного інтегрування методом прогноз-корекції другого порядку (по двох попередніх точках);
- 4) *prkr3.m* – процедура чисельного інтегрування методом прогноз-корекції третього порядку (по трьох попередніх точках);
- 5) *AdBach.m* – процедура чисельного інтегрування методом Адамса-Башфорта четвертого порядку (по чотирьох попередніх точках)
- 6) *KUE.m* – функція обчислення правих частин кінематичних рівнянь Ейлера;
- 7) *grafik91.m* – процедура побудови графіків 9 величин як функцій від однієї величини на трьох графічних підвікнах в одному графічному вікні один під одним;
- 8) *dngrbso1m.m* – процедура виведення на поле графіків розрахункових даних у текстовій формі.

Текст головної (керуючої) програми *PK_KUE_om_upr.m* наведений нижче.

```

% PK_KUE_om_upr
% Програма дослідження похибок алгоритму чисельного інтегрування
% 1) кінематичних рівнянь орієнтації у кутах Ейлера
% 2) багатокроковими методами другого, третього і четвертого (Адамса-Башфорта)
%   порядку
% 3) шляхом використання вимірювальної інформації про проекції кутової
%   швидкості основи
% Використовуються наступні процедури й функції:
% idbsosn1.m - процедура змінювання значень вхідних даних для
%   програми PK_KUE_om_upr
% ugdvoibm.m - функція обчислення параметрів поточного кутового руху
%   основи (кутів, кутових швидкостей, кутових прискорень)
% prkr2.m - процедура чисельного інтегрування методом Прогнозу-корекції
%   2-го порядку (по двох попередніх точках)

```

```

% prkr3.m - процедура чисельного інтегрування методом Прогнозу-коррекції
%          3-го порядку (по трьох попередніх точках)
% AdBach.m - процедура чисельного інтегрування методом Адамса-Башфорта
%          4-го порядку (по чотирьох попередніх точках)
% KUE.m - функція обчислення правих частин Кинематических Уравнений Эйлера
% grafik91.m - процедура побудови графіків 9 величин як функцій від однієї величини
%            на трьох підграфіках один під одним
% dngrbso1m.m - процедура виведення на поле графіків розрахункових
%            даних у текстовій формі

% Лазарев Ю. Ф., Бобровицька Я. 3-02-2006 р.
echo off, clc, clear all
global sprogram sname
global D2
global D5
sprogram='PK-KUE-om-upr';      clc
disp(' Ви увійшли в програму PK_KUE_om_upr,')
disp('яка інтегрує кінематичні рівняння Ейлера')
disp('багатокроковими алгоритмами прогноз-корекції')
disp('і обчислює похибки цього чисельного інтегрування')
disp(' ')
k0=menu('Що робити?','Продовжити роботу','Закінчити роботу');
if k0==1                          % Вхід у програму
    sname='Бобровицька Я.';
    % Введення "вшитих" значень первісних параметрів руху основи
    ps0=0;          te0=0;          fi0=0;
    psm=0.1;      tem=psm;      fim=psm;
    omps=2*pi/10;  omte=omps;  omfi=omps;
    eps=0;        ete=0;        efi=0;
    % параметрів інтегрування;
    h=0.2;        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 для ПОВТОРУ ЕКСПЕРИМЕНТІВ
    while k0==1
        idbso1; % БЛОК ЗМІНЮВАННЯ ДАНИХ;
        % Визначення мінімального Ti и максимального Ta періодів
        qwe=[D2(7) D2(8) D2(9)]; % Формування масиву частот кутових коливань
        Ti = 2*pi/max(qwe); Ta=Ti;
        for i=1:3,
            if D2(6+i)>0
                T1=2*pi/D2(6+i);
                if T1>Ta Ta=T1; end
                if T1<Ti Ti=T1; end
            end
        end
        % Визначення величини кроку інтегрування h
        h = D4(1);
        % Визначення моменту Tn0 часу початку формування
        % вектора масивів даних вимірювань
        Tn0 = Ta*D4(2);
        % Визначення кінцевого моменту часу tfinal інтегрування
        tfinal= Ta*(D4(2)+D4(3)); D5=[h Tn0 tfinal];
        % БЛОК ІНТЕГРУВАННЯ
        %%%%%%%%%%%%%%%
        t=0; % Завдання початкового моменту часу
        % Завдання вектора початкових умов у попередніх точках
        x=ugdvoibm(t-3*h); % Параметри руху ОСНОВИ 3 точки потому
        ys3=[x(1) x(2) x(3)]; om3=[x(7) x(8) x(9)];
        x=ugdvoibm(t-2*h); % Параметри руху ОСНОВИ 2 точки потому
        ys2=[x(1) x(2) x(3)]; om2=[x(7) x(8) x(9)];
    end
end

```

```

x=ugdvobm(t-h); % Параметри руху ОСНОВИ 1 точки потому
ys1=[x(1) x(2) x(3)]; om1=[x(7) x(8) x(9)];
x=ugdvobm(t); % Параметри руху ОСНОВИ у поточній точці
ys=[x(1) x(2) x(3)]; om=[x(7) x(8) x(9)];
x=ugdvobm(t+h); % Параметри руху ОСНОВИ у наступній точці
om_1=[x(7) x(8) x(9)];
y2s1=ys1;y2s=ys; % для 2-ого порядку
y3s2=ys2;y3s1=ys1; y3s=ys; % для 3-ого порядку
y4s3=ys3;y4s2=ys2; y4s1=ys1; y4s=ys; % для 4-ого порядку
imas=1; % Лічильник елементів вихідних масивів
% Цикл інтегрування
%%%%%%%%%%
disp('Почекайте (йде обчислення)')
while (imas-1)*h<tfinal
    t=(imas-1)*h;
    %%%%%%%%%% Метод прогнозу-корекції другого порядку
    y2 = prkr2('KUE',h,t,y2s1,y2s,om1,om,om_1);
    y2s1=y2s; y2s = y2;
    %%%%%%%%%% Метод прогнозу-корекції третього порядку
    y3 = prkr3('KUE',h,t,y3s2,y3s1,y3s,om2,om1,om,om_1);
    y3s2=y3s1; y3s1=y3s; y3s = y3;
    %%%%%%%%%% Метод Адамса-Башфорта
    y4 = AdBash('KUE',h,t,y4s3,y4s2,y4s1,y4s,om3,om2,om1,om,om_1);
    y4s3=y4s2; y4s2=y4s1; y4s1=y4s; y4s = y4;
    om3=om2; om2=om1; om1=om; om=om_1;
    x=ugdvobm(t+2*h); % Поточне положення основи
    om_1=[x(7) x(8) x(9)];
    x=ugdvobm(t+h);
    % для метода другого порядку
    dpsi2(imas)=y2s(1)-x(1); % Масиви похибок
    dteta2(imas)=y2s(2)-x(2); % оцінювання
    dfi2(imas)=y2s(3)-x(3); % кутів повороту основи
    % для метода третього порядку
    dpsi3(imas)=y3s(1)-x(1); % Масиви похибок
    dteta3(imas)=y3s(2)-x(2); % оцінювання
    dfi3(imas)=y3s(3)-x(3); % кутів повороту основи
    % для Адамса-Башфорта
    dpsi4(imas)=y4s(1)-x(1); % Масиви похибок
    dteta4(imas)=y4s(2)-x(2); % оцінювання
    dfi4(imas)=y4s(3)-x(3); % кутів повороту основи
    tt(imas)=t; % Массив моментів времени
    imas=imas+1;
end
% Побудова Графіків
%%%%%%%%%%
stit1='ПОХИБКИ БСО (кутові мінути) - алгоритми ПРОГНОЗУ-КОРЕКЦІЇ';
syllab1='Метод 2-го порядку'; sxlab1=''; stit2='';
syllab2='Метод 3-го порядку'; sxlab2=''; stit3='';
syllab3='Метод Адамса-Башфорта'; sxlab3='Час, с';
grafik91(tt,dpsi2*180*60/pi,dteta2*180*60/pi,dfi2*180*60/pi,stit1,syllab1,sxlab1,...
    dpsi3*180*60/pi,dteta3*180*60/pi,dfi3*180*60/pi,stit2,syllab2,sxlab2,...
    dpsi4*180*60/pi,dteta4*180*60/pi,dfi4*180*60/pi,stit3,syllab3,sxlab3);
dngrbso1m; % 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')

```

Програма використовує кілька допоміжних програм.

Програма idbsosn1.m забезпечує діалогове змінювання значень вхідних даних для програми РК_KUE_om_upr.m. Її текст слідує далі.

```
% idbsosn1
% Процедура изменения значений входных данных
% для программы РК_KUE_om_upr.m.

% Lazarev Yu. F., 13-03-2006

global D1 D2 D3 D4
% Формирование вектора строковых переменных;
% для упрощения процедуры ввода новых значений данных;
sD2 = ['Kt ='; 'Tt ='; 'Gt ='; 'Km ='; 'Tm ='; 'Gm ='; 'omK='; 'omT='; 'omG='; 'eK ='; ...
'eT ='; 'eG ='];
sD4 = [' h ='; 'Nt0 ='; 'Ndtf ='];
% ВЫВЕДЕНИЕ ЗАСТАВКИ на экран
clc, disp(' Інтегруються кінематичні рівняння Ейлера БСО')
disp(' ')
disp(' за наступних значень параметрів: ')
disp(' ')
disp(' руху ОСНОВИ ')
disp([sprintf(' Kt = %g ',D2(1)) sprintf(' Km = %g ',D2(4)) ...
sprintf(' omK = %g ',D2(7)) sprintf(' eK = %g ',D2(10))])
disp([sprintf(' Tt = %g ',D2(2)) sprintf(' Tm = %g ',D2(5)) ...
sprintf(' omT = %g ',D2(8)) sprintf(' eT = %g ',D2(11))])
disp([sprintf(' Gt = %g ',D2(3)) sprintf(' Gm = %g ',D2(6)) ...
sprintf(' omG = %g ',D2(9)) sprintf(' eG = %g ',D2(12))])
disp(' ')
disp(' інтегрування')
disp([sprintf(' h = %g ',D4(1)) sprintf(' Nt0 = %g ',D4(2)) ...
sprintf(' Ndtf = %g ',D4(3))]);

k=1
while k<3
% Выведение на экран меню выбора параметров
k= menu('Що змінити?',...
'Параметри руху ОСНОВИ',...
'Параметри інтегрування',...
'Ничого змінювати не потрібно');
% Ввод новых значений параметров
if k==1,
n=0;
while n<5
% Выведение на экран меню выбора параметров
n= menu('Що змінити?',...
['Сталі складові кут.швидкості ОСНОВИ'],...
['Амплітуди кутів хитавиці'],...
['Частоти хитавиці'],...
['Фази хитавиці'],...
'Ничого змінювати не потрібно');
if n==1,
ert=0;
while ert<4
% Выведение на экран меню выбора параметров
ert= menu('Пост. складові кут.швидкості ОСНОВИ',...
['Курсу Kt = ' num2str(D2(1))],...
['Тангажа Tt = ' num2str(D2(2))],...
['Крена Gt = ' num2str(D2(3))],...
'Ничого змінювати не потрібно');
if ert<4,
D2(ert)=input(['Введіть нове значення '...
sD2(ert,:)]);
```

```

        end
    end
end
if n==2,
    ert=0;
    while ert<4
        % Выведение на экран меню выбора параметров
        ert= menu('Амплітуди кутів',...
            ['Курсу   Km = ' num2str(D2(4))],...
            ['Тангажа   Tm = ' num2str(D2(5))],...
            ['Крена     Gm = ' num2str(D2(6))],...
            'Ничого змінювати не потрібно');
        if ert<4,
            D2(ert+3)=input(['Введіть нове значення '...
                sD2(ert+3,:)]);
        end
    end
end
if n==3,
    ert=0;
    while ert<4
        % Выведение на экран меню выбора параметров
        ert= menu('Частоты хитавиці (у радіанах у секунду)',...
            ['Курсу   omK = ' num2str(D2(7))],...
            ['Тангажа   omT = ' num2str(D2(8))],...
            ['Крена     omG = ' num2str(D2(9))],...
            'Ничого змінювати не потрібно');
        if ert<4,
            D2(ert+6)=input(['Введіть нове значення '...
                sD2(ert+6,:)]);
        end
    end
end
if n==4,
    ert=0;
    while ert<4
        % Выведение на экран меню выбора параметров
        ert= menu('Фази хитавиці (у радіанах)',...
            ['Курсу   eK = ' num2str(D2(10))],...
            ['Тангажа   eT = ' num2str(D2(11))],...
            ['Крена     eG = ' num2str(D2(12))],...
            'Ничого змінювати не потрібно');
        if ert<4,
            D2(ert+9)=input(['Введіть нове значення '...
                sD2(ert+9,:)]);
        end
    end
end
end
end
end
if k==2,
    n=0
    while n<4
        % Выведение на экран меню выбора параметров интегрирования
        n= menu('Що змінити?',...
            ['Крок інтегрування h = ' num2str(D4(1))],...
            ['Кількість максим. періодів до початку з'єму...
                інформації Nt0 = ' num2str(D4(2))],...
            ['Кількість максим. періодів часу з єму інформації ...
                Nddf= ' num2str(D4(3))],...
            'Ничого змінювати не потрібно');
        if n<4

```

```

                                D4(n)=input(['Введіть нове значення ' sD4(n,:)]);
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end
% КОНЕЦ процедуры изменения данных
%   Конеч БЛОКА ИЗМЕНЕНИЯ ПАРАМЕТРОВ

```

Процедура `grafik91.m` здійснює виведення даних, що подаються на неї як вхідні, у графічне вікно у вигляді графіків 9 величин як функцій від однієї величини на трьох графічних підвікнах один під одним. Далі наведений текст програми.

```

function y=grafik91(x,y1,y2,y3,stit1,sylab1,sxlab1,...
    y4,y5,y6,stit2,sylab2,sxlab2,...
    y7,y8,y9,stit3,sylab3,sxlab3)
% Построение Графика
figure
subplot(3,4,1:3)
plot(x,y1,'+k',x,y2,'.k',x,y3,'k'), grid
set(gca,'FontSize',10);
title(stit1), ylabel(sylab1), xlabel(sxlab1)
subplot(3,4,5:7)
plot(x,y4,'+k',x,y5,'.k',x,y6,'k'), grid
set(gca,'FontSize',10);
ylabel(sylab2), xlabel(sxlab2)
legend('\psi','\theta','\gamma',0)
subplot(3,4,9:11);
plot(x,y7,'+k',x,y8,'.k',x,y9,'k'), grid
set(gca,'FontSize',10);
ylabel(sylab3), xlabel(sxlab3)
zoom on

```

Наступна процедура `dngrbso1m.m` виводить на поле графіків розрахункові дані у текстовій формі – вид кінематичних рівнянь орієнтації, що інтегруються, вид залежностей від часу кутів повороту основи, значення параметрів руху основи.

```

% DnGrBSO1m
% Процедура выведения на поле графиков Данных
% программы расчета погрешностей БСО

% Лазарев Ю.Ф., Бобровицька Я. 02-02-2006
global D1 D2 D3 D4 D5 sprogram sname
subplot(3,4,[4 8 12]); axis('off');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h1=text(-0.9,1.05,'Інтегруються Кінематичні Рівняння Ейлера','FontSize',14);
h1=text(-0.1,1.0,'\psi' = [\Omega_zsin\gamma - ...
    \Omega_y cos\gamma]/cos\theta','FontSize',12);
h1=text(-0.1,.95,'\theta' = \Omega_z cos\gamma + \Omega_y sin\gamma','FontSize',12);
h1=text(-0.1,.9,'\gamma' = \Omega_x + [\Omega_zsin\gamma - ...
    \Omega_y cos\gamma]tg\theta','FontSize',12);
h1=text(-0.3,.85,'Вимірюються проекції кутової швидкості \Omega');
h1=text(-0.3,.8,'\ОСНОВА рухається за законом:','FontSize',12);
h1=text(-0.1,.75,'\psi = Kt*t + Km*sin(omK*t+eK)');

```

```

h1=text(-0.1,.7,'\theta = Tt*t + Tm*sin(omT*t+eT)');
h1=text(-0.1,.65,'\gamma = Gt*t+Gm*sin(omG*t+eG)');
h1=text(0,0.6,'Параметри руху основи:', 'FontSize',12);
%%%%%%%%%%
h1=text(-0.2,0.55,sprintf('Kt = %g ',D2(1)));
h1=text(-0.2,0.5,sprintf('Tt = %g ',D2(2)));
h1=text(-0.2,0.45,sprintf('Gt = %g ',D2(3)));
h1=text(0.7,0.55,sprintf('Km = %g ',D2(4)));
h1=text(0.7,0.5,sprintf('Tm = %g ',D2(5)));
h1=text(0.7,0.45,sprintf('Gm = %g ',D2(6)));
h1=text(-0.2,0.4,sprintf('omK = %g ',D2(7)));
h1=text(-0.2,0.35,sprintf('omT = %g ',D2(8)));
h1=text(-0.2,0.3,sprintf('omG = %g ',D2(9)));
h1=text(0.7,0.4,sprintf('eK = %g ',D2(10)));
h1=text(0.7,0.35,sprintf('eT = %g ',D2(11)));
h1=text(0.7,0.3,sprintf('eG = %g ',D2(12)));
%%%%%%%%%%
h1=text(-0.2,0.2,sprintf('Крок інтегрування: h = %g ',D5(1)), 'FontSize',12);
h1=text(-0.2,0.15,'-----');
tm=fix(clock); Tv=tm(4:6);
h1=text(0,0.1,['Програма ' sprogram]);
h1=text(-0.2,0.05,['Виконав ' sname]);
h1=text(-0.2,0.0,[sprintf(' %g : ',Tv) ' ' date]);
h1=text(-0.2,-0.05,'-----');
h1=text(-0.1,-0.1,'Ukraine, KPI, cath. PSON');

```

Решта програм описані раніше.

Керуюча програма *PK_Poisson_om_upr.m* забезпечує запуск і взаємодію комплексу програмних засобів, які здійснюють чисельне інтегрування рівнянь орієнтації у вигляді матричного рівняння Пуассона методами прогноз-корекції.

У склад цього комплексу входять:

- 1) *idbso2.m* – процедура змінювання значень вхідних даних для програми *PK_Poisson_om_upr*;
- 2) *ugdvoibm.m* – функція обчислення параметрів поточного кутового руху основи (кутів, кутових швидкостей, кутових прискорень);
- 3) *prkr2.m* – процедура чисельного інтегрування методом прогноз-корекції 2-го порядку (по двох попередніх точках);
- 4) *prkr3.m* – процедура чисельного інтегрування методом прогноз-корекції 3-го порядку (по трьох попередніх точках);
- 5) *AdBach.m* – процедура чисельного інтегрування методом Адамса-Башфорта 4-го порядку (по чотирьох попередніх точках);
- 6) *Poisson.m* – функція обчислення правих частин рівняння Пуассона;
- 7) *grafik91.m* – процедура побудови графіків 9 величин як функцій від однієї величини;
- 8) *DnGrBSO2.m* – процедура виведення на поле графіків розрахункових даних у текстовій формі.

Текст керуючої програми наводиться нижче.

```

% PK_Poisson_om_upr
% Програма дослідження похибок алгоритму чисельного інтегрування
% 1) кінематичного рівняння орієнтації Пуассона
% 2) багатокроковими методами другого, третього і четвертого (Адамса-Башфорта)
%   порядку

```

```

% 3) шляхом використання вимірювальної інформації про проекції кутової
% швидкості основи
% Використовуються наступні процедури й функції:
% idbso2.m - процедура змінювання значень вхідних даних для
% програми BSOint2m_upr
% ugdvobm.m - функція обчислення параметрів поточного кутового руху
% основи (кутів, кутових швидкостей, кутових прискорень)
% prkr2.m - процедура чисельного інтегрування методом Прогнозу-коррекції
% 2-го порядку (по двох попередніх точках)
% prkr3.m - процедура чисельного інтегрування методом Прогнозу-коррекції
% 3-го порядку (по трьох попередніх точках)
% AdBach.m - процедура чисельного інтегрування методом Адамса-Башфорта
% 4-го порядку (по чотирьох попередніх точках)
% Poisson.m - функція обчислення правих частин Уравнений Пуассона
% grafik91.m - процедура побудови графіків 9 величин як функцій від однієї величини
% на трьох підграфіках один під одним
% DnGrBSO2.m - процедура виведення на поле графіків розрахункових
% даних у текстовій формі

% Лазарев Ю. Ф., Бобровицька Я. 12-03-2006 р.
echo off, clc, clear all
global sprogram sname
global D2
global D5
sprogram='PK-Poisson-om-upr';
clc
disp(' Ви увійшли в програму PK_Poisson_om_upr,')
disp(' яка інтегрує кінематичні рівняння Пуассона')
disp('багатокроковими алгоритмами прогноз-корекції')
disp('і обчислює похибки цього чисельного інтегрування'), disp(' ')
k0=menu('Що робити?','Продовжити роботу','Закінчити роботу');
if k0==1 % Вхід у програму
    sname='Бобровицька Я.';
    % Введення "вшитих" значень первісних параметрів руху основи
    ps0=0; te0=0; fi0=0;
    psm=0.1; tem=psm; fim=psm;
    omps=2*pi/10; omte=omps; omfi=omps;
    eps=0; ete=0; efi=0;
    % параметрів інтегрування;
    h=0.2; 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 для ПОВТОРУ ЕКСПЕРИМЕНТІВ
    while k0==1
        idbso2; % БЛОК ЗМІНЮВАННЯ ДАНИХ;
        % Визначення мінімального Ti і максимального Ta періодів
        qwe=[D2(7) D2(8) D2(9)]; % Формування масиву частот кутових коливань
        Ti = 2*pi/max(qwe); Ta=Ti;
        for i=1:3,
            if D2(6+i)>0
                T1=2*pi/D2(6+i);
                if T1>Ta Ta=T1; end
                if T1<Ti Ti=T1; end
            end
        end
        % Визначення величини кроку інтегрування h
        h = D4(1);
        % Визначення моменту Tn0 часу початку формування
        % вектора масивів даних вимірювань
        Tn0 = Ta*D4(2);
        % Визначення кінцевого моменту часу tfinal інтегрування

```

```

tfinal= Ta*(D4(2)+D4(3));
D5=[h Tn0 tfinal];
% БЛОК ИНТЕГРУВАННЯ
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t=0; % Завдання початкового моменту часу
% Завдання вектора початкових умов у попередніх точках
x=ugdvobm(t-3*h); % Параметри руху ОСНОВИ у момент часу t-3*h
ys3=[-x(1) x(2) x(3)]; om3=[x(7) x(8) x(9)];
x=ugdvobm(t-2*h); % Параметри руху ОСНОВИ у момент часу t-2*h
ys2=[-x(1) x(2) x(3)]; om2=[x(7) x(8) x(9)];
x=ugdvobm(t-h); % Параметри руху ОСНОВИ у момент часу t-h
ys1=[-x(1) x(2) x(3)]; om1=[x(7) x(8) x(9)];
x=ugdvobm(t); % Параметри руху ОСНОВИ у поточний момент часу t
ys=[-x(1) x(2) x(3)]; om=[x(7) x(8) x(9)];
x=ugdvobm(t+h); % Параметри руху ОСНОВИ у наступний момент часу t+h
ys_1=[-x(1) x(2) x(3)]; om_1=[x(7) x(8) x(9)];
kc=[2,3,1];
% Обчислення матриць напрямних косинусів
C2s1=ug2mnk(ys1,kc); C2s=ug2mnk(ys,kc); % для метода 2-го порядку
C3s2=ug2mnk(ys2,kc); C3s1=C2s1; C3s=C2s; % для метода 3-ого порядку
C4s3=ug2mnk(ys3,kc); C4s2=C3s2; C4s1=C2s1; C4s=C2s;% для 4-ого порядку
imas=1; % Лічильник елементів вихідних масивів
% Цикл інтегрування
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('Почекайте (йде обчислення)')
while (imas-1)*h<tfinal
    t=(imas-1)*h;
    %%%%%%%%% Метод прогнозу-корекції другого порядку
    C2 = prkr2('Poisson',h,t,C2s1,C2s,om1,om,om_1);
    C2s1=C2s; C2s = C2;
    %%%%%%%%% Метод прогнозу-корекції третього порядку
    C3 = prkr3('Poisson',h,t,C3s2,C3s1,C3s,om2,om1,om,om_1);
    C3s2=C3s1; C3s1=C3s; C3s = C3;
    %%%%%%%%% Метод Адамса-Башфорта
    C4 = AdBash('Poisson',h,t,C4s3,C4s2,C4s1,C4s,om3,om2,om1,om,om_1);
    C4s3=C4s2; C4s2=C4s1; C4s1=C4s; C4s = C4;
    om3=om2; om2=om1; om1=om; om=om_1;
    x=ugdvobm(t+h);
    % для метода другого порядку
    y2=MNK2Ug(C2,kc);
    psi2(imas)=y2(1); %
    teta2(imas)=y2(2); % оцінки
    fi2(imas)=y2(3); % кутів повороту основи
    dps2(imas)=y2(1)+x(1); % Масиви похибок
    dteta2(imas)=y2(2)-x(2); % оцінювання
    dfi2(imas)=y2(3)-x(3); % кутів повороту основи
    % для метода третього порядку
    y3=MNK2Ug(C3,kc);
    dps3(imas)=y3(1)+x(1); % Масиви похибок
    dteta3(imas)=y3(2)-x(2); % оцінювання
    dfi3(imas)=y3(3)-x(3); % кутів повороту основи
    % для Адамса-Башфорта
    y4=MNK2Ug(C4,kc);
    dps4(imas)=y4(1)+x(1); % Масиви похибок
    dteta4(imas)=y4(2)-x(2); % оцінювання
    dfi4(imas)=y4(3)-x(3); % кутів повороту основи
    x=ugdvobm(t+2*h); % Параметри руху ОСНОВИ у наступний момент часу t+2h
    ys_1=[x(1) x(2) x(3)]; om_1=[x(7) x(8) x(9)];
    tt(imas)=t; % Массив моментів часу
    imas=imas+1;
end
% Побудова Графіків

```

```

%%%%%%%%%%
stit1='ПОХИБКИ БСО (кутові мінути)'; sylab1='Метод 2-го порядку'; sxlab1=' ';
stit2=' '; sylab2='Метод 3-го порядку'; sxlab2=' ';
stit3=' '; sylab3='Метод Адамса-Башфорта '; sxlab3='Час, с';
grafik91(tt,dpsi2*180*60/pi,dteta2*180*60/pi,dfi2*180*60/pi,stit1,sylab1,sxlab1,...
    dpsi3*180*60/pi,dteta3*180*60/pi,dfi3*180*60/pi,stit2,sylab2,sxlab2,...
    dpsi4*180*60/pi,dteta4*180*60/pi,dfi4*180*60/pi,stit3,sylab3,sxlab3);
DnGrBSO2; % Виведення на поле графіків розрахункових даних
% Кінець процедури виведення графіків
k0=menu('Що робити?','Продовжити роботу','Закінчити роботу');
end
end % Вихід з циклу програми
clear global
disp('Кінець програми')

```

Головною відмінністю цієї керуючої програми від попередньої є необхідність початкового переходу від обчислених значень кутів орієнтації до значення відповідної матриці напрямних косинусів і у подальшому, після завершення кроку чисельного інтегрування – зворотній перехід від матриці напрямних косинусів до значень кутів орієнтації. Перша операція здійснюється за допомогою функції *Ug2MNK*, а друга – за допомогою процедури *MNK2Ug*. Тексти обох процедур наведені у додатку.

Програма *PK_Quatern_om_upr* здійснює керування сукупністю програм, які забезпечують чисельне інтегрування кінематичного рівняння у кватерніонах методами прогнозу-корекції і обчислення похибок цього інтегрування.

Використовуються наступні процедури й функції:

- 1) *idbsosn1.m* – процедура змінювання значень вхідних даних для програми *PK_Quatern_om_upr*;
- 2) *UgDvOb.m* – функція обчислення параметрів поточного кутового руху основи (кутів, кутових швидкостей, кутових прискорень);
- 3) *prkr2.m* – процедура чисельного інтегрування методом прогноз-корекції 2-го порядку (по двох попередніх точках);
- 4) *prkr3.m* – процедура чисельного інтегрування методом прогноз-корекції 3-го порядку (по трьох попередніх точках);
- 5) *AdBach.m* – процедура чисельного інтегрування методом Адамса-Башфорта 4-го порядку (по чотирьох попередніх точках);
- 6) *KRKvat.m* – функція обчислення правої частини кватерніонного рівняння орієнтації;
- 7) *grafik91.m* – процедура побудови графіків 9 величин як функцій від однієї величини;
- 8) *dngrbsoUgSk.m* – процедура виведення на поле графіків розрахункових даних у текстовій формі.

Далі наведений текст керуючої програми.

```

% PK_Quatern_om_upr
% Програма дослідження похибок алгоритма чисельного інтегрування
% 1) кінематичних рівнянь орієнтації у кватерніонах
% 2) багатокроківими методами другого, третього і четвертого (Адамса-Башфорта)
%    порядку
% 3) шляхом використання вимірювальної інформації про проекції кутової
%    швидкості основи

```

```

% Використовуються наступні процедури й функції:
% idbsosn1.m - процедура змінювання значень вхідних даних для
%           програми PK_Quatern_om_upr
% UgDvOb.m - функція обчислення параметрів поточного кутового руху
%           основи (кутів, кутових швидкостей, кутових прискорень)
% prkr2.m - процедура чисельного інтегрування методом Прогнозу-коррекції
%           2-го порядку (по двох попередніх точках)
% prkr3.m - процедура чисельного інтегрування методом Прогнозу-коррекції
%           3-го порядку (по трьох попередніх точках)
% AdBach.m - процедура чисельного інтегрування методом Адамса-Башфорта
%           4-го порядку (по чотирьох попередніх точках)
% KRKvat.m - пр-ра обчислення правої частини кватерніонного рівняння орієнтації
% grafik91.m - процедура побудови графіків 9 величин як функцій від однієї величини
%           на трьох підграфіках один під одним
% dngrbsoUgSk.m - процедура виведення на поле графіків розрахункових
%           даних у текстовій формі

% Лазарев Ю. Ф., Бобровицька Я. 21-03-2006 р.
% 2-06-2006
echo off, clc, clear all
global sprogram sname
global D2
global D5
sprogram='PK-Quatern-om-upr'; clc
disp(' Ви увійшли в програму PK_Quatern_om_upr,')
disp('яка інтегрує кінематичні рівняння в кватерніонах')
disp('методами прогноз-корекції')
disp('по вимірюваних проекціях кутової швидкості основи')
disp('і обчислює похибки цього чисельного інтегрування'), disp(' ')
k0=menu('Що робити?','Продовжити роботу','Закінчити роботу');
if k0==1 % Вхід у програму
    sname='Бобровицька Я.';
    % Введення "вшитих" значень первісних параметрів руху основи
    ps0=0; te0=0; fi0=0;
    psm=0.1; tem=psm; fim=psm;
    omps=2*pi/10; omte=omps; omfi=omps;
    eps=0; ete=0; efi=0;
    % параметрів інтегрування;
    h=0.2; 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 для ПОВТОРУ ЕКСПЕРИМЕНТІВ
    while k0==1
        idbsosn1; % БЛОК ЗМІНЮВАННЯ ДАНИХ;
        % Визначення мінімального Ti і максимального Ta періодів
        qwe=[D2(7) D2(8) D2(9)]; % Формування масиву частот кутових коливань
        Ti = 2*pi/max(qwe); Ta=Ti;
        for i=1:3,
            if D2(6+i)>0
                T1=2*pi/D2(6+i);
                if T1>Ta Ta=T1; end
                if T1<Ti Ti=T1; end
            end
        end
        % Визначення величини кроку інтегрування h
        h = D4(1);
        % Визначення моменту Tn0 часу початку формування
        % вектора масивів даних вимірювань
        Tn0 = Ta*D4(2);
        % Визначення кінцевого моменту часу tfinal інтегрування
        tfinal= Ta*(D4(2)+D4(3));
    end
end

```



```

D5=[h Tn0 tfinal];
% БЛОК ИНТЕГРУВАННЯ
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t=0; % Завдання початкового моменту часу
% Завдання вектора початкових умов у попередніх точках
x=Ugdvobm(t-3*h); % Параметри руху ОСНОВИ 3 точки потому
ys3=[-x(1) x(2) x(3)]; om3=[x(7) x(8) x(9)];
Ls3=Ug2quat(-x(1),2,x(2),3,x(3),1);
x=Ugdvobm(t-2*h); % Параметри руху ОСНОВИ 2 точки потому
ys2=[-x(1) x(2) x(3)]; om2=[x(7) x(8) x(9)];
Ls2=Ug2quat(-x(1),2,x(2),3,x(3),1);
x=Ugdvobm(t-h); % Параметри руху ОСНОВИ 1 точки потому
ys1=[-x(1) x(2) x(3)]; om1=[x(7) x(8) x(9)];
Ls1=Ug2quat(-x(1),2,x(2),3,x(3),1);
x=Ugdvobm(t); % Параметри руху ОСНОВИ у поточній точці
ys=[-x(1) x(2) x(3)]; om=[x(7) x(8) x(9)];
Ls=Ug2quat(-x(1),2,x(2),3,x(3),1);
x=Ugdvobm(t+h); % Параметри руху ОСНОВИ у наступній точці
om_1=[x(7) x(8) x(9)];
kv=[2 3 1];
L2s1=Ls1; L2s=Ls; % для 2-ого порядку
L3s2=Ls2; L3s1=Ls1; L3s=Ls; % для 3-ого порядку
L4s3=Ls3; L4s2=Ls2; L4s1=Ls1; L4s=Ls; % для 4-ого порядку
imas=1; % Лічильник елементів вихідних масивів
% Цикл інтегрування
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('Почекайте (йде обчислення)')
while (imas-1)*h<tfinal
    t=(imas-1)*h;
    %%%%%%%%% Метод прогнозу-корекції другого порядку
    L2 = prkr2('KRKvat',h,t,L2s1,L2s,om1,om,om_1);
    L2s1=L2s; L2s = L2;
    %%%%%%%%% Метод прогнозу-корекції третього порядку
    L3 = prkr3('KRKvat',h,t,L3s2,L3s1,L3s,om2,om1,om,om_1);
    L3s2=L3s1; L3s1=L3s; L3s = L3;
    %%%%%%%%% Метод Адамса-Башфорта
    L4 = AdBash('KRKvat',h,t,L4s3,L4s2,L4s1,L4s,om3,om2,om1,om,om_1);
    L4s3=L4s2; L4s2=L4s1; L4s1=L4s; L4s = L4;
    om3=om2; om2=om1; om1=om; om=om_1;
    x=Ugdvobm(t+2*h); % Поточне положення основи
    om_1=[x(7) x(8) x(9)];
    x=Ugdvobm(t+h);
    K=x(1)*180*60/pi; T=x(2)*180*60/pi; G=x(3)*180*60/pi;
    % для метода другого порядку
    Yg2=Quat2Ug(L2,kv);
    K2=Yg2(1)*180*60/pi; T2=Yg2(2)*180*60/pi; G2=Yg2(3)*180*60/pi;
    dps2(imas)=Yg2(1)+x(1); % Масиви похибок
    dteta2(imas)=Yg2(2)-x(2); % оцінювання
    dfi2(imas)=Yg2(3)-x(3); % кутів повороту основи
    % для метода третього порядку
    Yg3=Quat2Ug(L3,kv);
    K3=Yg3(1)*180*60/pi; T3=Yg3(2)*180*60/pi; G3=Yg3(3)*180*60/pi;
    dps3(imas)=Yg3(1)+x(1); % Масиви похибок
    dteta3(imas)=Yg3(2)-x(2); % оцінювання
    dfi3(imas)=Yg3(3)-x(3); % кутів повороту основи
    % для Адамса-Башфорта
    Yg4=Quat2Ug(L4,kv);
    K4=Yg4(1)*180*60/pi; T4=Yg4(2)*180*60/pi; G4=Yg4(3)*180*60/pi;
    dps4(imas)=Yg4(1)+x(1); % Масиви похибок
    dteta4(imas)=Yg4(2)-x(2); % оцінювання
    dfi4(imas)=Yg4(3)-x(3); % кутів повороту основи
    tt(imas)=t; % Массив моментів в часі

```

```

    imas=imas+1;
end
% Побудова Графіків
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
stit1='ПОХИБКИ БСО (кутові мінути) - методи ПРОГНОЗ-КОРЕКЦІЇ';
syllab1='Метод 2-го порядку';          sxlab1=' ';          stit2=' ';
syllab2='Метод 3-го порядку';          sxlab2=' ';          stit3=' ';
syllab3='Метод Адамса-Башфорта ';     sxlab3='Час, с';
grafik91(tt,dpsi2*180*60/pi,dteta2*180*60/pi,dfi2*180*60/pi,stit1,syllab1,sxlab1,...
         dpsi3*180*60/pi,dteta3*180*60/pi,dfi3*180*60/pi,stit2,syllab2,sxlab2,...
         dpsi4*180*60/pi,dteta4*180*60/pi,dfi4*180*60/pi,stit3,syllab3,sxlab3);
dngrbsoUgSk; % 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')

```

У цій програмі, на відміну від попередніх, використовуються функції – процедури *Ug2Quat* і *Quat2Ug* переходу від кутів орієнтації до кватерніону і навпаки. Окрім того, при застосуванні функції *KRKvat* використовується операція кватерніонного добутку, тобто застосовується новий клас об'єктів - кватерніонів, бібліотека якого описується у додатку.

4.4. Результати моделювання

Результати роботи описаних вище комплексів програм наводяться далі. Для випробування обрано режим синхронних коливань основи з усіх трьох кутів Ейлера з періодом $T=10$ с і з амплітудою $0,1$ рад. $\approx 5,73^\circ = 343,8'$. Розглянуто два варіанти цього режиму:

- 1) синфазні коливання з усіх трьох кутів;
- 2) коливання з куту тангажа зсунуті по фазі на 90° по відношенню до коливань з кутів курса і крена.

Для кожного варіанта режиму обчислені поточні значення похибок визначення усіх трьох кутів орієнтації при чисельному інтегруванні трьома методами прогнозу-корекції (другого, третього і четвертого порядку) за двох значень кроку інтегрування $h=0,2$ с і $h=0,02$ с.

Отже, для кожного з видів рівнянь орієнтації проведені 4 експерименти. Результати цих експериментів наведені у графічній формі:

- на рис. 4.1...4.4 подані похибки вищевказаних алгоритмів для кінематичних рівнянь Ейлера;
- на рис. 4.5...4.8 подані результати для кінематичного рівняння Пуассона;
- рис. 4.9...4.12 подають результати для кватерніонного рівняння орієнтації.

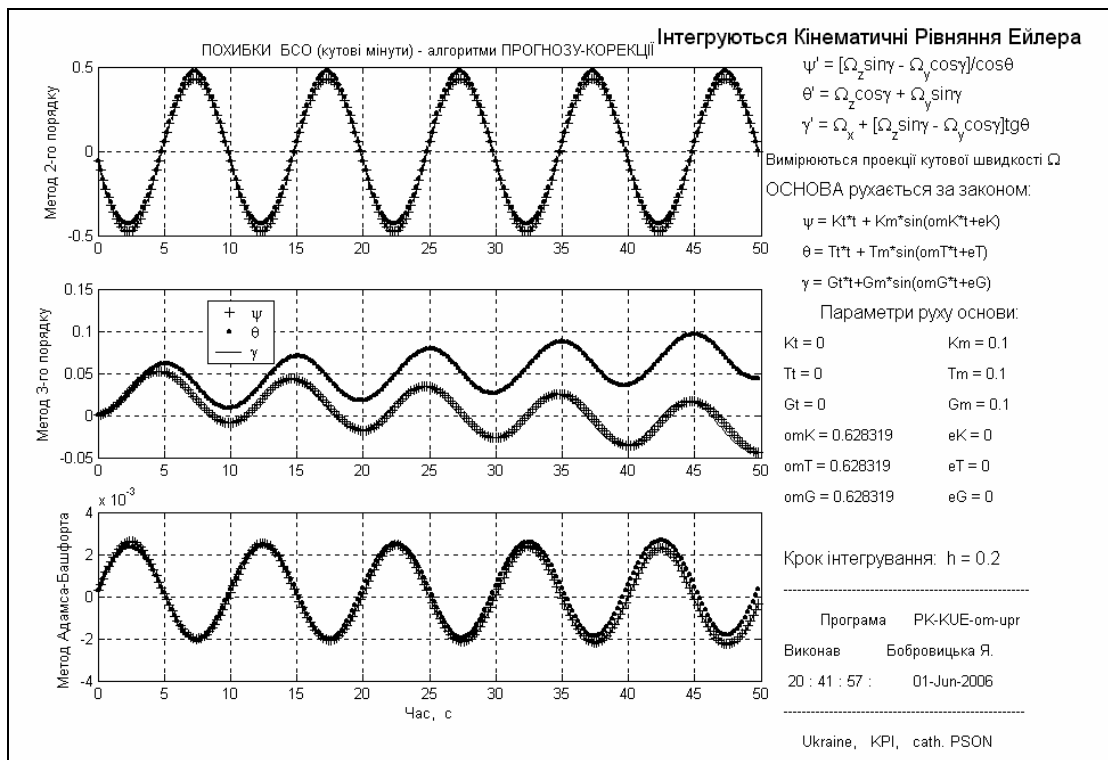


Рис. 4.1. Похибки застосування алгоритмів прогноз-корекції для інтегрування кінематичних рівнянь Ейлера ($h=0.2c$) при синхронних і синфазних коливаннях основи з трьох кутів

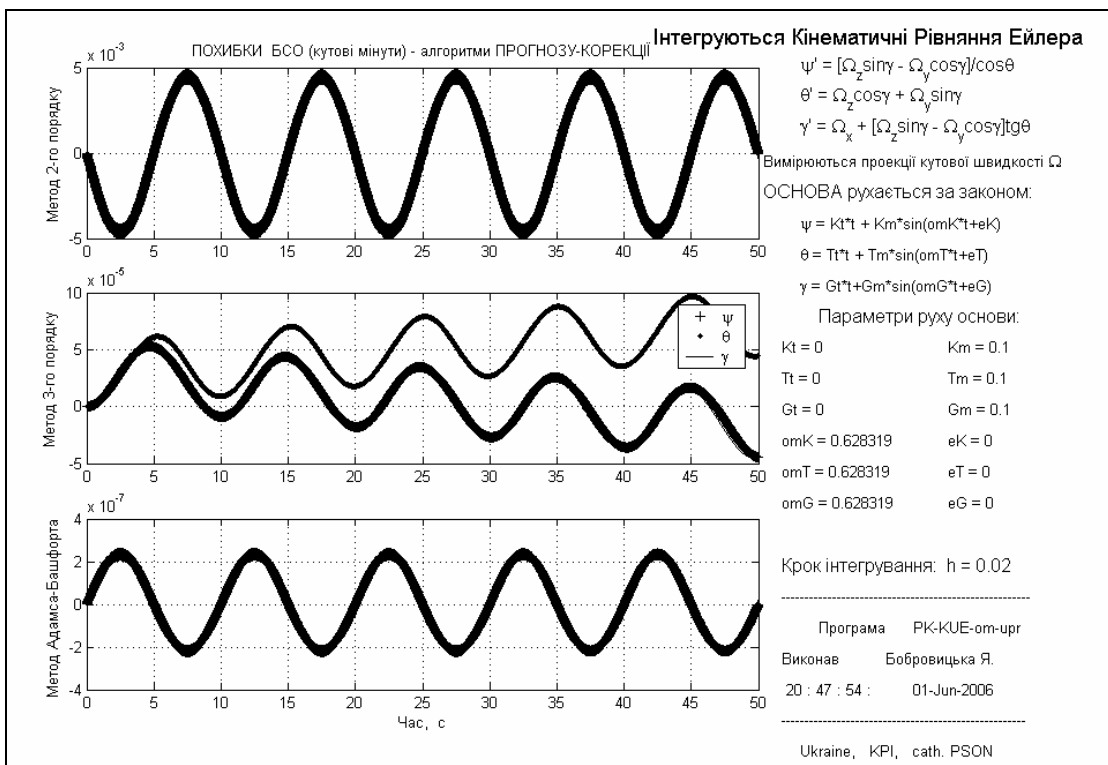


Рис. 4.2. Похибки застосування алгоритмів прогноз-корекції для інтегрування кінематичних рівнянь Ейлера ($h=0.02c$) при синхронних і синфазних коливаннях основи з трьох кутів

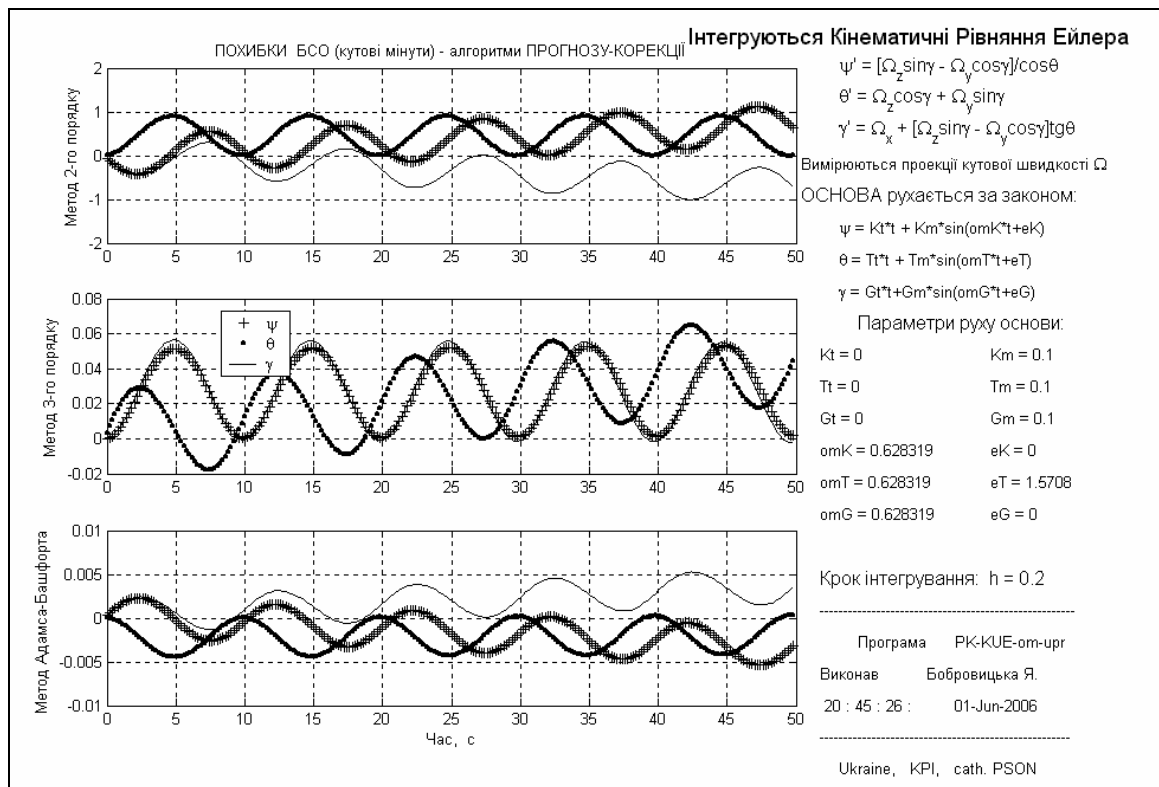


Рис. 4.3. Похибки застосування алгоритмів прогноз-корекції для інтегрування кінематичних рівнянь Ейлера ($h=0.2c$) при синхронних коливаннях основи з зсувом фаз 90°

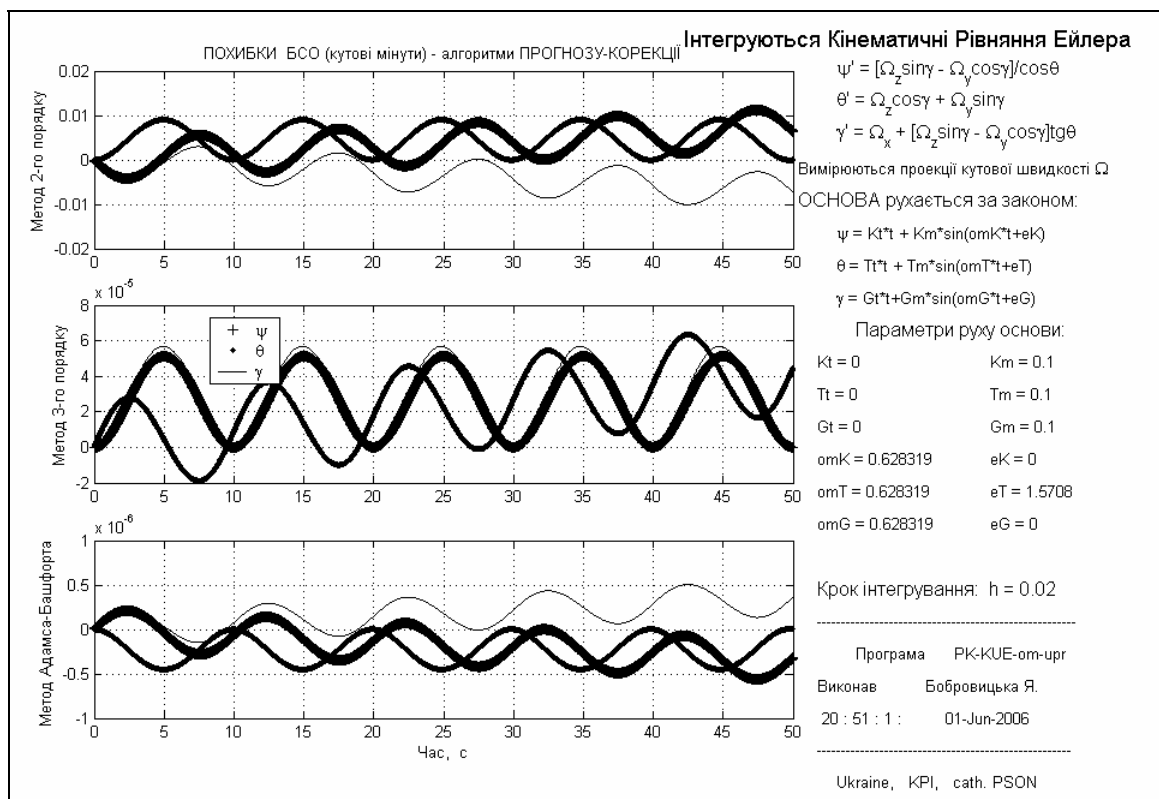


Рис. 4.4. Похибки застосування алгоритмів прогноз-корекції для інтегрування кінематичних рівнянь Ейлера ($h=0.2c$) при синхронних коливаннях основи з зсувом фаз 90°

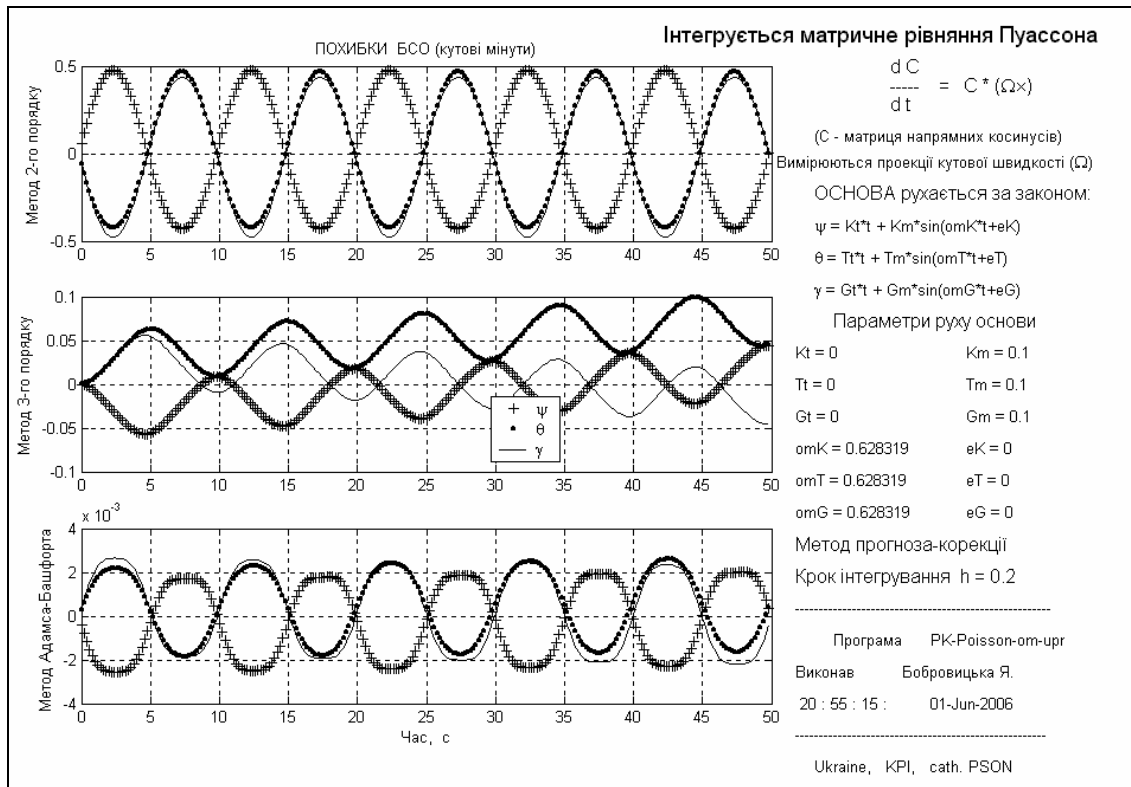


Рис. 4.5. Похибки застосування алгоритмів прогноз-корекції для інтегрування матричного рівняння Пуассона ($h=0.2c$) при синхронних і синфазних коливаннях основи з трьох кутів

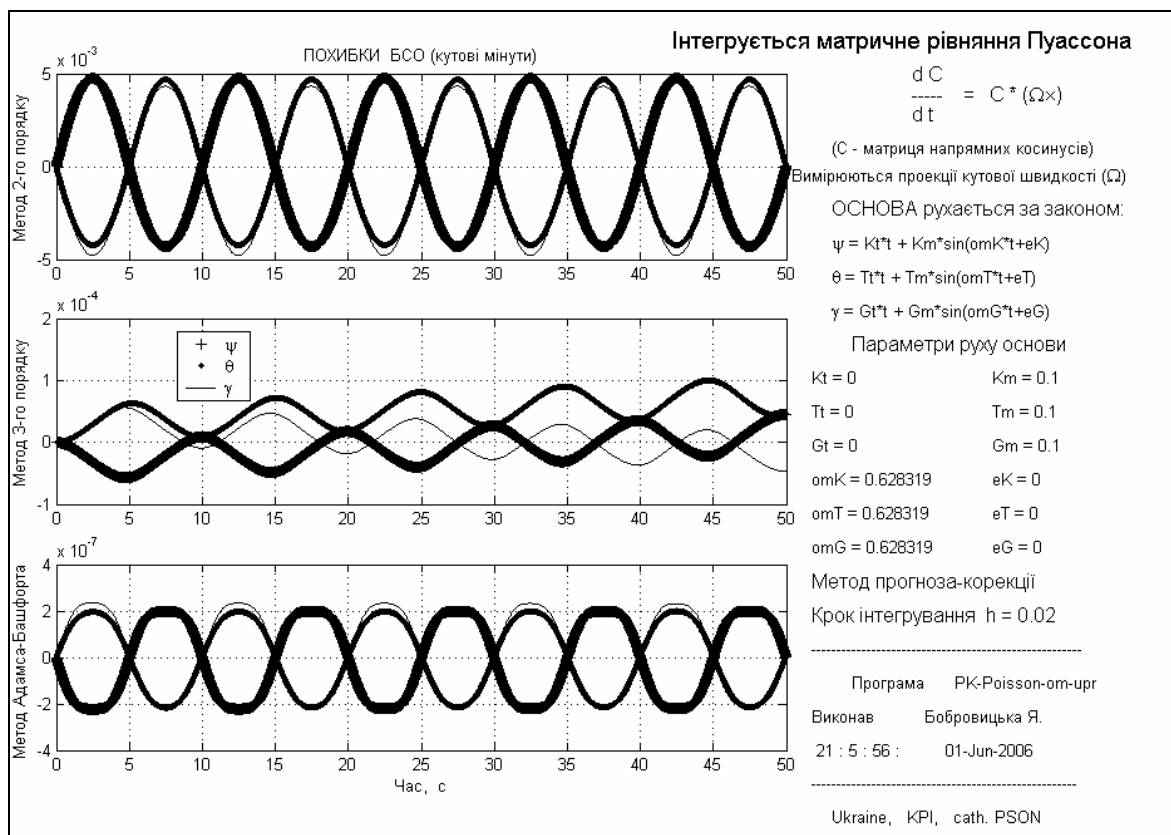


Рис. 4.6. Похибки застосування алгоритмів прогноз-корекції для інтегрування матричного рівняння Пуассона ($h=0.02c$) при синхронних і синфазних коливаннях основи з трьох кутів

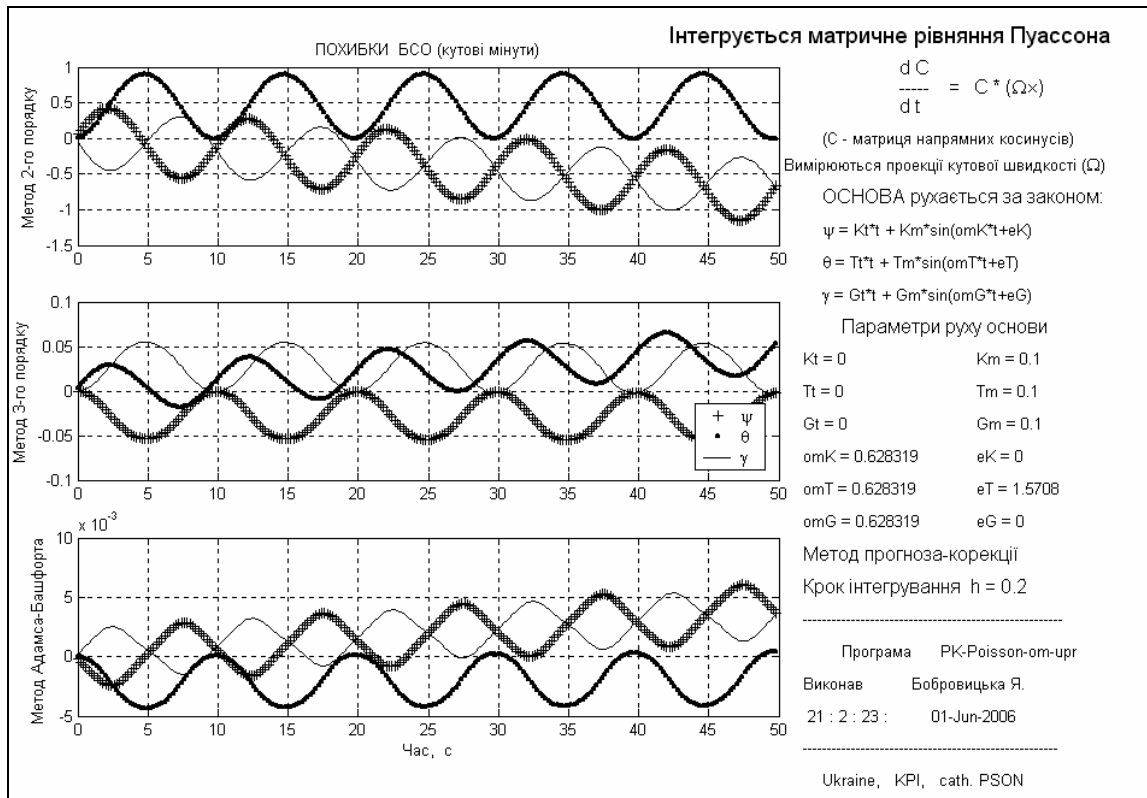


Рис. 4.7. Похибки застосування алгоритмів прогноз-корекції для інтегрування матричного рівняння Пуассона ($h=0.2c$) при синхронних коливаннях основи з зсувом фаз 90°

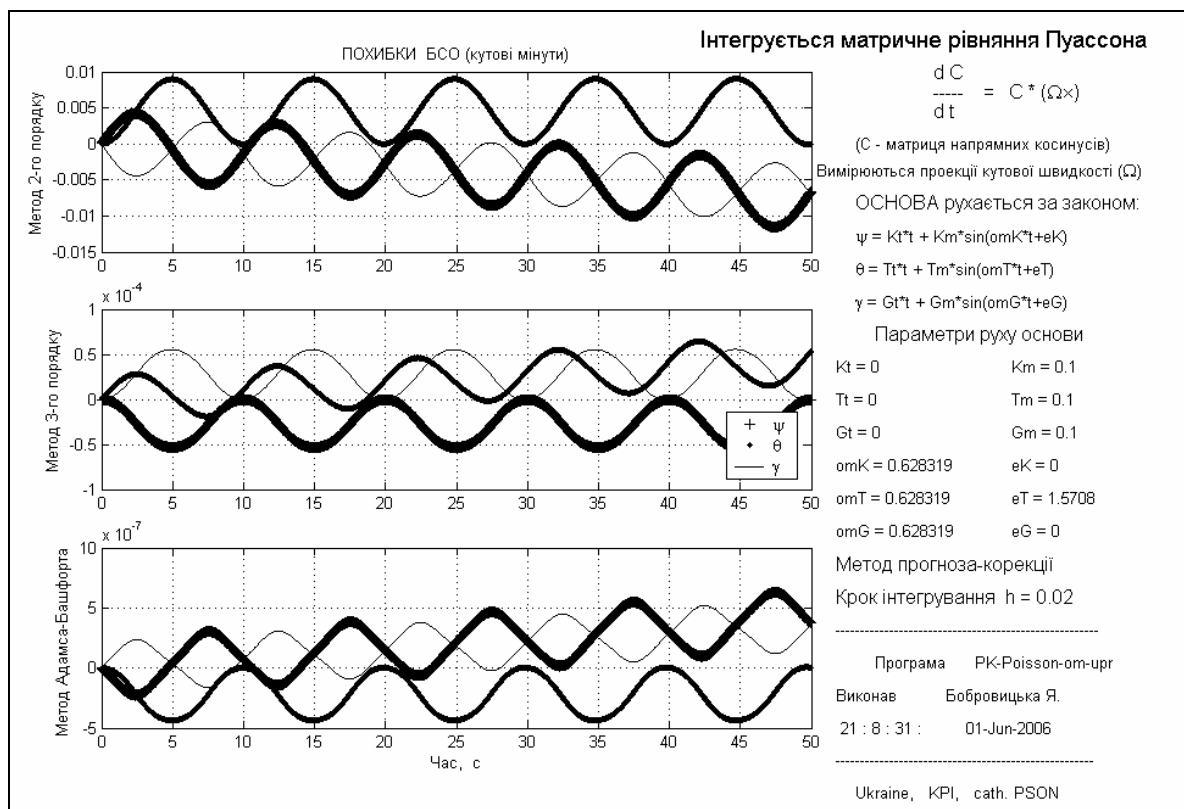


Рис. 4.8. Похибки застосування алгоритмів прогноз-корекції для інтегрування матричного рівняння Пуассона ($h=0.02c$) при синхронних коливаннях основи з зсувом фаз 90°

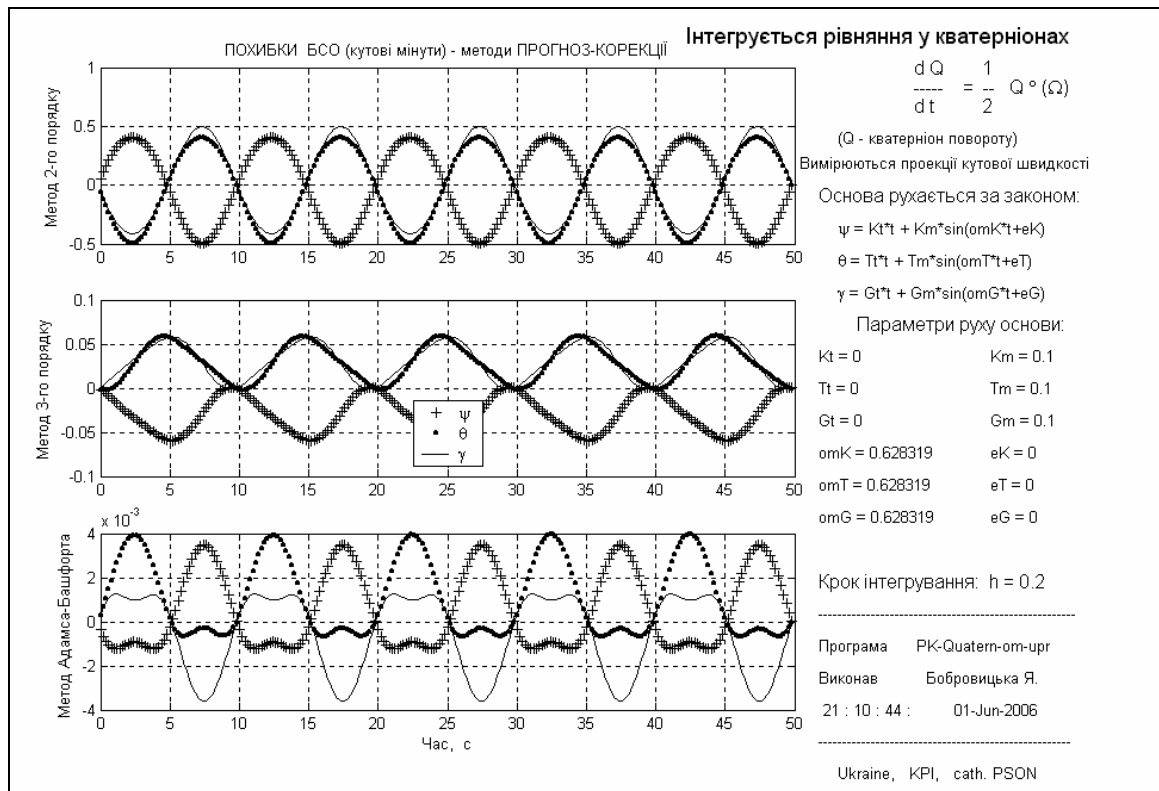


Рис. 4.9. Похибки застосування алгоритмів прогноз-корекції для інтегрування кінематичного рівняння у кватерніонах ($h=0.2c$) при синхронних і синфазних коливаннях основи з трьох кутів

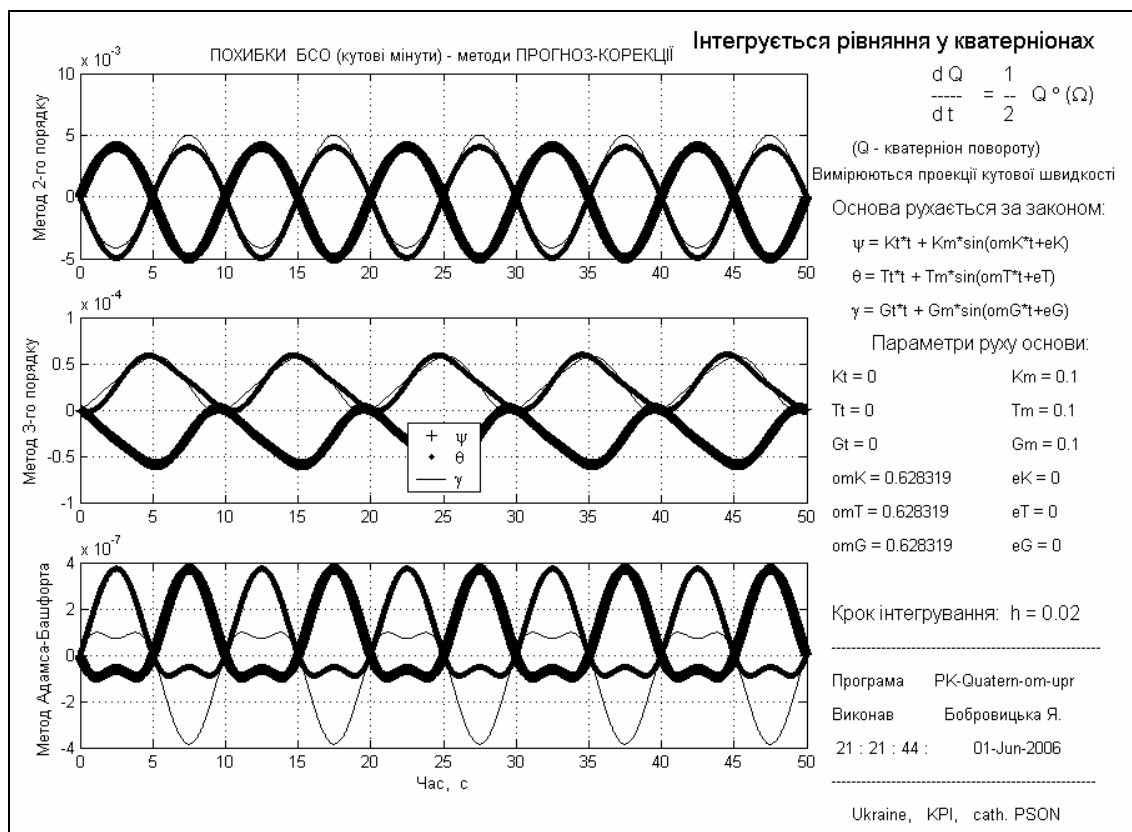


Рис. 4.10. Похибки застосування алгоритмів прогноз-корекції для інтегрування кінематичного рівняння у кватерніонах ($h=0.02c$) при синхронних і синфазних коливаннях основи з трьох кутів

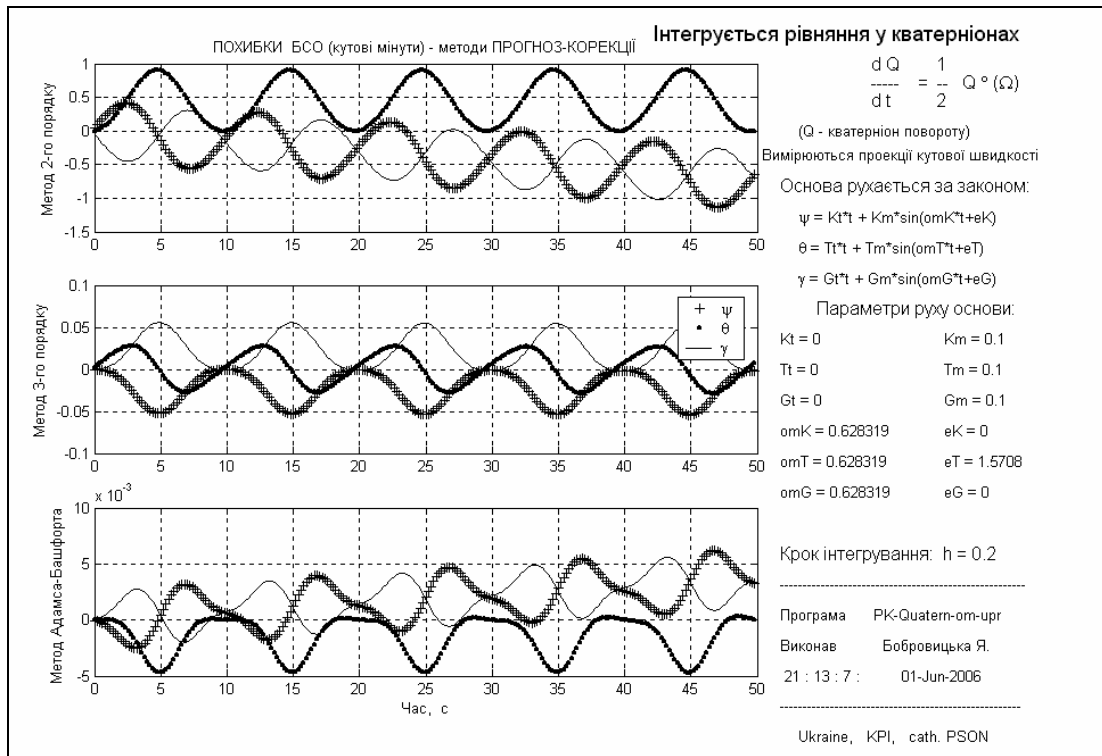


Рис. 4.11. Похибки застосування алгоритмів прогноз-корекції для інтегрування кінематичного рівняння у кватерніонах (h=0.2с) при синхронних коливаннях основи з зсувом фаз 90°

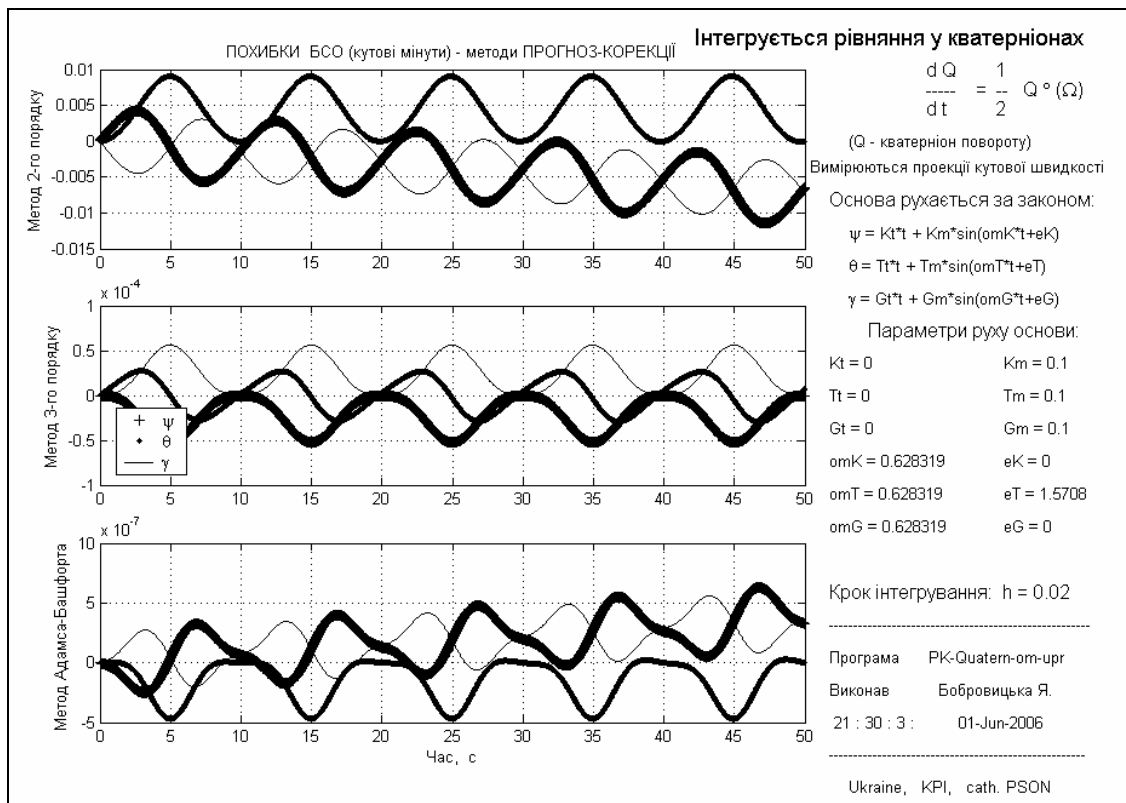


Рис. 4.12. Похибки застосування алгоритмів прогноз-корекції для інтегрування кінематичного рівняння у кватерніонах (h=0.02с) при синхронних коливаннях основи з зсувом фаз 90°

Як показують графіки одержаних результатів, похибка алгоритму змінюється з часом з періодом хитавиці основи і незмінною амплітудою. Незмінність амплітуд похибок свідчить про стійкість розглянутих алгоритмів. При цьому у деяких випадках виникає дрейф похибки, тобто постійне зростання середнього значення похибки з деякою незмінною швидкістю. Дрейф похибки є найбільш небезпечною складовою похибки алгоритму, оскільки призводить до неперервного накопичення похибки з часом. Тому головними параметрами похибки будемо вважати амплітуду (піврозмах) її коливання і величину дрейфу з відповідного куту.

Результати моделювання зведені у таблиці 4.2 і 4.3, де наведені приблизні значення амплітуд похибок алгоритмів і величини дрейфів похибок.

Таблиця 4.2. Параметри похибок алгоритмів прогноз-корекції при синфазній хитавиці основи

Вид руху основи	Крок	Порядок методу	Вид параметра	Рівняння Ейлера	Рівняння Пуассона	Кватерніонне рівняння
Синфазні і синхронні коливання основи з трьох кутів з періодом 10 с і амплітудою 0,1 радіан	0,2с	2	Амплітуда	$(5 \cdot 10^{-1})'$	$(5 \cdot 10^{-1})'$	$(5 \cdot 10^{-1})'$
			Дрейф з ψ	0	0	0
			Дрейф з ϑ	0	0	0
			Дрейф з γ	0	0	0
		3	Амплітуда	$(2,5 \cdot 10^{-2})'$	$(2,5 \cdot 10^{-2})'$	$(2,5 \cdot 10^{-2})'$
			Дрейф з ψ	$-(1 \cdot 10^{-3})'/c$	$(1 \cdot 10^{-3})'/c$	0
			Дрейф з ϑ	$(1 \cdot 10^{-3})'/c$	$(1 \cdot 10^{-3})'/c$	0
			Дрейф з γ	0	0	0
		4	Амплітуда	$(2,5 \cdot 10^{-3})'$	$(2,5 \cdot 10^{-3})'$	$(2,5 \cdot 10^{-3})'$
			Дрейф з ψ	0	0	0
			Дрейф з ϑ	0	0	0
			Дрейф з γ	0	0	0
	0,02с	2	Амплітуда	$(5 \cdot 10^{-3})'$	$(5 \cdot 10^{-3})'$	$(5 \cdot 10^{-3})'$
			Дрейф з ψ	0	0	0
			Дрейф з ϑ	0	0	0
			Дрейф з γ	0	0	0
		3	Амплітуда	$(2,5 \cdot 10^{-5})'$	$(2,5 \cdot 10^{-5})'$	$(2,5 \cdot 10^{-5})'$
			Дрейф з ψ	$-(1 \cdot 10^{-6})'/c$	$(1 \cdot 10^{-6})'/c$	0
			Дрейф з ϑ	$(1 \cdot 10^{-6})'/c$	$(1 \cdot 10^{-6})'/c$	0
			Дрейф з γ	0	0	0
		4	Амплітуда	$(2,5 \cdot 10^{-7})'$	$(2 \cdot 10^{-7})'$	$(2 \cdot 10^{-7})'$
			Дрейф з ψ	0	0	0
			Дрейф з ϑ	0	0	0
			Дрейф з γ	0	0	0

			Дрейф з γ	0	0	0
--	--	--	------------------	---	---	---

Таблиця 4.3. Параметри похибок алгоритмів прогноз-корекції при еліптичній хитавиці основи

Вид руху основи	Крок	Порядок методу	Вид параметра	Рівняння Ейлера	Рівняння Пуассона	Кватерніонне рівняння
Синфазні коливання основи з трьох кутів зі зсувом фаз 90° з періодом 10 с і амплітудою 0,1 радіан	0,2с	2	Амплітуда	$(5 \cdot 10^{-1})'$	$(5 \cdot 10^{-1})'$	$(5 \cdot 10^{-1})'$
			Дрейф з ψ	$(1 \cdot 10^{-2})'/c$	$-(1,2 \cdot 10^{-2})'/c$	$-(1,4 \cdot 10^{-2})'/c$
			Дрейф з ϑ	0	0	0
			Дрейф з γ	$-(1 \cdot 10^{-2})'/c$	$-(1,2 \cdot 10^{-2})'/c$	$-(1,4 \cdot 10^{-2})'/c$
		3	Амплітуда	$(2,5 \cdot 10^{-2})'$	$(2,6 \cdot 10^{-2})'$	$(2,6 \cdot 10^{-2})'$
			Дрейф з ψ	0	0	0
			Дрейф з ϑ	$(1 \cdot 10^{-3})'/c$	$(1 \cdot 10^{-3})'/c$	0
			Дрейф з γ	0	0	0
		4	Амплітуда	$(2,5 \cdot 10^{-3})'$	$(2,5 \cdot 10^{-3})'$	$(2,5 \cdot 10^{-3})'$
			Дрейф з ψ	$(6 \cdot 10^{-5})'/c$	$(6 \cdot 10^{-5})'/c$	$(6 \cdot 10^{-5})'/c$
			Дрейф з ϑ	0	0	0
			Дрейф з γ	$-(6 \cdot 10^{-5})'/c$	$(6 \cdot 10^{-5})'/c$	$(6 \cdot 10^{-5})'/c$
	0,02с	2	Амплітуда	$(5 \cdot 10^{-3})'$	$(5 \cdot 10^{-3})'$	$(5 \cdot 10^{-3})'$
			Дрейф з ψ	$(8 \cdot 10^{-5})'/c$	$-(1,4 \cdot 10^{-4})'/c$	$-(1,4 \cdot 10^{-4})'/c$
			Дрейф з ϑ	0	0	0
			Дрейф з γ	$-(8 \cdot 10^{-5})'/c$	$-(1,4 \cdot 10^{-4})'/c$	$-(1,4 \cdot 10^{-4})'/c$
		3	Амплітуда	$(2,5 \cdot 10^{-5})'$	$(2,5 \cdot 10^{-5})'$	$(2,5 \cdot 10^{-5})'$
			Дрейф з ψ	0	0	0
			Дрейф з ϑ	$(8 \cdot 10^{-7})'/c$	$(1 \cdot 10^{-6})'/c$	0
			Дрейф з γ	0	0	0
		4	Амплітуда	$(2,5 \cdot 10^{-7})'$	$(2 \cdot 10^{-7})'$	$(2 \cdot 10^{-7})'$
			Дрейф з ψ	$(6 \cdot 10^{-9})'/c$	$(6 \cdot 10^{-9})'/c$	$(6 \cdot 10^{-9})'/c$
			Дрейф з ϑ	0	0	0
			Дрейф з γ	$-(6 \cdot 10^{-9})'/c$	$(6 \cdot 10^{-9})'/c$	$(6 \cdot 10^{-9})'/c$

Попередній аналіз одержаних внаслідок моделювання похибок алгоритмів прогноз-корекції приводить до наступних висновків.

1. Похибки методу певного порядку практично не залежать від виду рівнянь орієнтації. Зокрема, амплітуди коливань похибки є практично однаковими для будь яких рівнянь орієнтації. Те саме можна казати й про дрейфи похибки з усіх трьох кутів.

2. Усі промодельовані алгоритми відповідають власним порядкам. При зменшенні кроку інтегрування у 10 разів (з 0,2 до 0,02 с) амплітуда і дрейфи алгоритму другого порядку (функція $prkr2$) зменшуються за величиною у 10^2 разів, для алгоритму методу третього порядку (функція $prkr3$) – у 10^3 разів, а для алгоритму методу четвертого порядку (функція $AdVash$) – у 10^4 разів.
3. При синфазних коливаннях основи з усіх трьох кутів Ейлера дрейфи похибок алгоритмів парного порядку відсутні. Дрейфи похибок виникають з кутів ψ і ϑ при застосуванні алгоритмів непарного порядку. Виключення складають алгоритми, прикладені до кватерніонного рівняння орієнтації. У цьому випадку дрейфи похибок відсутні.
4. При еліптичній хитавиці основи амплітуди змінювання і дрейфи похибок практично не змінюються у порівнянні з такими при синфазних коливаннях, але змінюються кути, за якими здійснюються дрейфи. При цьому аналогічні дрейфи виникають і при застосуванні кватерніонного рівняння орієнтації.
5. При частоті опитування вимірювачів 5 Гц відносна амплітуда похибок найменш точного методу другого порядку не перевищує 0,15%.

4.5. Висновки

1. Розроблені на основі методів прогноз-корекції алгоритми є стійкими, можуть бути застосовані для будь яких видів рівнянь орієнтації і забезпечують достатню точність визначення параметрів орієнтації.

2. Для підвищення точності визначення параметрів орієнтації дієвим є шлях підвищення частоти опитування вимірювачів, причому ця дієвість підвищується за рахунок застосування методів більш високого порядку. Так, застосування частоти вимірювання 50 Гц у випадку використання методу четвертого порядку дозволяє зменшити відносну амплітуду похибки до величини $7,3 \cdot 10^{-8}\%$ (у порівнянні з 0,15% при частоті опитування 5 Гц для методу другого порядку).

5. Алгоритми методу Пікара для рівняння Пуассона

Тепер розглянемо рівняння орієнтації у формі матричного рівняння Пуассона. Побудуємо алгоритми другого, третього і четвертого порядків на основі застосування до цього матричного рівняння метода послідовних наближень Пікара.

5.1. Розроблення алгоритмів

5.1.1. Постановка задачі

Задача полягає в одержанні алгоритму розв'язування матричного диференціального рівняння:

$$\frac{dC}{dt} = C(\Omega \times), \quad C(t_0) = C_0, \quad (5.1)$$

де

$$(\Omega \times) = \begin{vmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{vmatrix}. \quad (5.2)$$

При цьому інформація про кутову швидкість надходить у дискретні моменти часу $t_i (i = 1, 2, \dots)$ з інтервалами h (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 квазі-координат $\Delta\theta_j(t_i)$ (при використанні інтегруючих гіротахometrів):

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

Інтегруючи (1) почленно у межах від t_0 до t , одержимо

$$C(t) = C(t_0) + \int_{t_0}^t C(\xi) \cdot [\Omega(\xi) \times] d\xi. \quad (5.3)$$

Розв'язок рівняння (5.3) шукатимемо у вигляді добутку початкового значення матриці напрямних косинусів на деяку невідому матрицю – функцію часу, вираз якої й потрібно відшукати:

$$C(t) = C(t_0) \cdot P(t). \quad (5.4)$$

Підставимо (5.4) у (5.3):

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

звідки випливає

$$P(t) = E + \int_{t_0}^t P(\xi) \cdot [\Omega(\xi) \times] d\xi, \quad (5.5)$$

де позначено E одиничну матрицю розміром (3×3) . Тепер рівняння (3) можна подати у вигляді (5.4).

Розглянемо алгоритм відшукування 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 (5.4), дійдемо

$$C_{n+1} = C_n \cdot P_{n+1}, \quad (5.6)$$

причому, у відповідності до (5), P_{n+1} є розв'язком рівняння

$$P(t) = E + \int_{t_n}^t P(\xi) \cdot [\Omega(\xi) \times] d\xi \quad (5.7)$$

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

Рівняння (7.7), за методом Пікара, розв'язується послідовними наближеннями.

5.1.2. Побудова наближених розв'язків допоміжного рівняння

У нульовому наближенні прийmemo $P_{(0)}(t) = E$. Підставивши це у праву частину (7), одержимо розв'язок цього рівняння у першому наближенні:

$$P_{(1)}(t) = E + \int_{t_n}^t P_{(0)}(\xi) \cdot [\Omega(\xi) \times] d\xi = E + \int_{t_n}^t [\Omega(\xi) \times] d\xi. \quad (5.8)$$

Для відшукування розв'язку у другому наближенні, за методом Пікара, достатньо підставити у праву частину рівняння (7) результат попереднього наближення матриці P :

$$\begin{aligned} P_{(2)}(t) &= E + \int_{t_n}^t P_{(1)}(\xi) \cdot [\Omega(\xi) \times] d\xi = E + \int_{t_n}^t \left(E + \int_{t_n}^{\xi} [\Omega(\eta) \times] d\eta \right) \cdot [\Omega(\xi) \times] d\xi = \\ &= E + \int_{t_n}^t [\Omega(\xi) \times] d\xi + \int_{t_n}^t \left(\int_{t_n}^{\xi} [\Omega(\eta) \times] d\eta \right) \cdot [\Omega(\xi) \times] d\xi. \end{aligned} \quad (5.9)$$

Використовуючи цей результат, аналогічним чином відшукаємо третє наближення

$$\begin{aligned} P_{(3)}(t) &= E + \int_{t_n}^t P_{(2)}(\xi) \cdot [\Omega(\xi) \times] d\xi = \\ &= E + \int_{t_n}^t \left[E + \int_{t_n}^{\xi} [\Omega(\eta) \times] d\eta + \int_{t_n}^{\xi} \left(\int_{t_n}^{\eta} [\Omega(\zeta) \times] d\zeta \right) \cdot [\Omega(\eta) \times] d\eta \right] \cdot [\Omega(\xi) \times] d\xi = \\ &= E + \int_{t_n}^t [\Omega(\xi) \times] d\xi + \int_{t_n}^t \left(\int_{t_n}^{\xi} [\Omega(\eta) \times] d\eta \right) \cdot [\Omega(\xi) \times] d\xi + \\ &+ \int_{t_n}^t \left[\int_{t_n}^{\xi} \left(\int_{t_n}^{\eta} [\Omega(\zeta) \times] d\zeta \right) \cdot [\Omega(\eta) \times] d\eta \right] \cdot [\Omega(\xi) \times] d\xi. \end{aligned} \quad (5.10)$$

і четверте наближення:

$$\begin{aligned}
P_{(4)}(t) &= E + \int_{t_n}^t P_{(3)}(\xi) \cdot [\Omega(\xi) \times] d\xi = \\
&= E + \int_{t_n}^t [\Omega(\xi) \times] d\xi + \int_{t_n}^t \left(\int_{t_n}^{\xi} [\Omega(\eta) \times] d\eta \right) \cdot [\Omega(\xi) \times] d\xi + \\
&+ \int_{t_n}^t \left[\int_{t_n}^{\xi} \left(\int_{t_n}^{\eta} [\Omega(\zeta) \times] d\zeta \right) \cdot [\Omega(\eta) \times] d\eta \right] \cdot [\Omega(\xi) \times] d\xi + \\
&+ \int_{t_n}^t \left\{ \int_{t_n}^{\xi} \left[\int_{t_n}^{\eta} \left(\int_{t_n}^{\zeta} [\Omega(v) \times] dv \right) \cdot [\Omega(\zeta) \times] d\zeta \right] \cdot [\Omega(\eta) \times] d\eta \right\} \cdot [\Omega(\xi) \times] d\xi. \quad (5.11)
\end{aligned}$$

5.1.3. Уточнення розв'язків для поліноміальної апроксимації кутової швидкості

При розробці алгоритмів чисельного інтегрування рівняння (5.7) на борті рухомого об'єкта за допомогою бортових цифрових обчислювачів і вимірювачів слід взяти до уваги, що усі підінтегральні функції часу у виразах (5.8)...(5.11) у цьому випадку є невідомими. Відомими є лише значення цих функцій у декількох попередніх моментах часу. Щоб обчислити за цими значеннями значення інтегралів, потрібно апроксимувати вектор $\boldsymbol{\omega} = \mathbf{i}\omega_x + \mathbf{j}\omega_y + \mathbf{k}\omega_z$ проєкцій кутової швидкості об'єкта на потрібному інтервалі часу як явну функцію часу. Наприклад, для того, щоб побудувати алгоритми чисельного інтегрування рівняння Пуассона не вище за четвертий порядок мализни відносно малої величини кроку інтегрування, достатньо цей вектор в діапазоні від t_n до t_{n+1} (від $\tau = 0$ до $\tau = h$) подати як поліном третього порядку відносно часу τ :

$$\boldsymbol{\omega} = \boldsymbol{\omega}_o + 2\boldsymbol{\varepsilon} \cdot \tau + 3\mathbf{a} \cdot \tau^2 + 4\mathbf{b} \cdot \tau^3, \quad (5.12)$$

де $\boldsymbol{\omega}_o$, $\boldsymbol{\varepsilon}$, \mathbf{a} і \mathbf{b} - постійні вектори, значення яких визначається шляхом поліноміального інтерполювання по вимірних значеннях кутової швидкості (чи інтеграла від неї на кількох попередніх кроках інтегрування).

$$\begin{aligned}
\int_{t_n}^t [\Omega(\xi) \times] d\xi &= \int_{t_n}^t [\boldsymbol{\omega}(\xi) \times] d\xi = \int_0^{\tau} [(\boldsymbol{\omega}_o \times) + 2(\boldsymbol{\varepsilon} \times) \cdot \xi + 3(\mathbf{a} \times) \cdot \xi^2 + 4(\mathbf{b} \times) \cdot \xi^3] d\xi = \\
&= (\boldsymbol{\omega}_o \times) \cdot \tau + (\boldsymbol{\varepsilon} \times) \cdot \tau^2 + (\mathbf{a} \times) \cdot \tau^3 + (\mathbf{b} \times) \cdot \tau^4. \quad (5.13)
\end{aligned}$$

Позначення $(\mathbf{x} \times)$ має зміст кососиметричної матриці, що відповідає вектору $\mathbf{x}\{x_1, x_2, x_3\}$:

$$(\mathbf{x} \times) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}.$$

Використовуючи отриманий вираз першого інтеграла, можна відшукати й подвійний інтеграл. При обчисленні його нехтуватимемо членами, які містять як множник величину τ у степені, вище за четвертий:

$$\begin{aligned}
& \int_{t_n}^t \left(\int_{t_n}^{\xi} [\Omega(\eta) \times] d\eta \right) \cdot [\Omega(\xi) \times] d\xi = \\
& = \int_0^{\tau} [(\mathbf{\omega}_o \times) \cdot \xi + (\mathbf{\epsilon} \times) \cdot \xi^2 + (\mathbf{a} \times) \cdot \xi^3 + (\mathbf{b} \times) \cdot \xi^4] [(\mathbf{\omega}_o \times) + \\
& \quad + 2(\mathbf{\epsilon} \times) \cdot \xi + 3(\mathbf{a} \times) \cdot \xi^2 + 4(\mathbf{b} \times) \cdot \xi^3] d\xi \approx \\
& \approx \frac{1}{2} (\mathbf{\omega}_o \times)^2 \cdot \tau^2 + \frac{1}{3} [2(\mathbf{\omega}_o \times)(\mathbf{\epsilon} \times) + (\mathbf{\epsilon} \times)(\mathbf{\omega}_o \times)] \cdot \tau^3 + \\
& \quad + \frac{1}{4} [3(\mathbf{\omega}_o \times)(\mathbf{a} \times) + 2(\mathbf{\epsilon} \times)^2 + (\mathbf{a} \times)(\mathbf{\omega}_o \times)] \tau^4. \tag{5.14}
\end{aligned}$$

Наступний (потрійний) інтеграл матиме вигляд:

$$\begin{aligned}
& \int_0^{\tau} \left[\int_0^{\xi} \left(\int_0^{\eta} [\Omega(\zeta) \times] d\zeta \right) [\Omega(\zeta) \times] d\eta \right] [\Omega(\xi) \times] d\xi \approx \\
& \approx \int_0^{\tau} \left[\frac{1}{2} (\mathbf{\omega}_o \times)^2 \cdot \xi^2 + \frac{1}{3} [2(\mathbf{\omega}_o \times)(\mathbf{\epsilon} \times) + (\mathbf{\epsilon} \times)(\mathbf{\omega}_o \times)] \cdot \xi^3 + \frac{1}{4} [3(\mathbf{\omega}_o \times)(\mathbf{a} \times) + 2(\mathbf{\epsilon} \times)^2 + (\mathbf{a} \times)(\mathbf{\omega}_o \times)] \xi^4 \right] \cdot \\
& \quad \cdot [(\mathbf{\omega}_o \times) + 2(\mathbf{\epsilon} \times) \cdot \xi + 3(\mathbf{a} \times) \cdot \xi^2 + 4(\mathbf{b} \times) \cdot \xi^3] d\xi \approx \\
& \approx \frac{1}{6} (\mathbf{\omega}_o \times)^3 \tau^3 + \frac{1}{4} \left[(\mathbf{\omega}_o \times)^2 (\mathbf{\epsilon} \times) + \frac{2}{3} (\mathbf{\omega}_o \times)(\mathbf{\epsilon} \times)(\mathbf{\omega}_o \times) + \frac{1}{3} (\mathbf{\epsilon} \times)(\mathbf{\omega}_o \times)^2 \right] \tau^4. \tag{5.15}
\end{aligned}$$

Нарешті четвертий інтеграл набуде наступного виду:

$$\begin{aligned}
& \int_0^{\tau} \left\{ \int_0^{\xi} \left[\int_0^{\eta} \left(\int_0^{\zeta} [\Omega(v) \times] dv \right) [\Omega(\zeta) \times] d\zeta \right] [\Omega(\eta) \times] d\eta \right\} [\Omega(\xi) \times] d\xi \approx \\
& = \int_0^{\tau} \left\{ \frac{1}{6} (\mathbf{\omega}_o \times)^3 \xi^3 + \frac{1}{4} \left[(\mathbf{\omega}_o \times)^2 (\mathbf{\epsilon} \times) + \frac{2}{3} (\mathbf{\omega}_o \times)(\mathbf{\epsilon} \times)(\mathbf{\omega}_o \times) + \frac{1}{3} (\mathbf{\epsilon} \times)(\mathbf{\omega}_o \times)^2 \right] \xi^4 \right\} \cdot \\
& \quad \cdot [(\mathbf{\omega}_o \times) + 2(\mathbf{\epsilon} \times) \cdot \xi + 3(\mathbf{a} \times) \cdot \xi^2 + 4(\mathbf{b} \times) \cdot \xi^3] d\xi \approx \frac{1}{24} (\mathbf{\omega}_o \times)^4 \tau^4. \tag{5.16}
\end{aligned}$$

У підсумку, з точністю до малих четвертого порядку величина P_{n+1} у відповідності до (5.11) визначиться формулою

$$\begin{aligned}
P_{n+1} = & E + \int_0^h [\Omega(\xi) \times] d\xi + \int_0^h \left(\int_0^{\xi} [\Omega(\eta) \times] d\eta \right) \cdot [\Omega(\xi) \times] d\xi + \\
& + \int_0^h \left[\int_0^{\xi} \left(\int_0^{\eta} [\Omega(\zeta) \times] d\zeta \right) \cdot [\Omega(\eta) \times] d\eta \right] \cdot [\Omega(\xi) \times] d\xi + \\
& + \int_0^h \left\{ \int_0^{\xi} \left[\int_0^{\eta} \left(\int_0^{\zeta} [\Omega(v) \times] dv \right) \cdot [\Omega(\zeta) \times] d\zeta \right] \cdot [\Omega(\eta) \times] d\eta \right\} \cdot [\Omega(\xi) \times] d\xi =
\end{aligned}$$

$$\begin{aligned}
&= E + (\boldsymbol{\omega}_o \times) \cdot h + (\boldsymbol{\varepsilon} \times) \cdot h^2 + (\mathbf{a} \times) \cdot h^3 + (\mathbf{b} \times) \cdot h^4 + \frac{1}{2} (\boldsymbol{\omega}_o \times)^2 h^2 + \\
&+ \frac{1}{3} [2(\boldsymbol{\omega}_o \times)(\boldsymbol{\varepsilon} \times) + (\boldsymbol{\varepsilon} \times)(\boldsymbol{\omega}_o \times)] h^3 + \frac{1}{4} [3(\boldsymbol{\omega}_o \times)(\mathbf{a} \times) + 2(\boldsymbol{\varepsilon} \times)^2 + (\mathbf{a} \times)(\boldsymbol{\omega}_o \times)] h^4 + \\
&+ \frac{1}{6} (\boldsymbol{\omega}_o \times)^3 h^3 + \frac{1}{4} \left[(\boldsymbol{\omega}_o \times)^2 (\boldsymbol{\varepsilon} \times) + \frac{2}{3} (\boldsymbol{\omega}_o \times)(\boldsymbol{\varepsilon} \times)(\boldsymbol{\omega}_o \times) + \frac{1}{3} (\boldsymbol{\varepsilon} \times)(\boldsymbol{\omega}_o \times)^2 \right] h^4 + \frac{1}{24} (\boldsymbol{\omega}_o \times)^4 h^4.
\end{aligned}$$

Остаточню маємо:

$$\begin{aligned}
P_{n+1} &= E + (\boldsymbol{\omega}_o \times) \cdot h + [(\boldsymbol{\varepsilon} \times) + \frac{1}{2} (\boldsymbol{\omega}_o \times)^2] \cdot h^2 + \\
&+ \left\{ (\mathbf{a} \times) + \frac{1}{3} [2(\boldsymbol{\omega}_o \times)(\boldsymbol{\varepsilon} \times) + (\boldsymbol{\varepsilon} \times)(\boldsymbol{\omega}_o \times)] + \frac{1}{6} (\boldsymbol{\omega}_o \times)^3 \right\} \cdot h^3 + \\
&+ \left\{ (\mathbf{b} \times) + \frac{1}{4} [3(\boldsymbol{\omega}_o \times)(\mathbf{a} \times) + 2(\boldsymbol{\varepsilon} \times)^2 + (\mathbf{a} \times)(\boldsymbol{\omega}_o \times)] + \right. \\
&\left. + \frac{1}{4} \left[(\boldsymbol{\omega}_o \times)^2 (\boldsymbol{\varepsilon} \times) + \frac{2}{3} (\boldsymbol{\omega}_o \times)(\boldsymbol{\varepsilon} \times)(\boldsymbol{\omega}_o \times) + \frac{1}{3} (\boldsymbol{\varepsilon} \times)(\boldsymbol{\omega}_o \times)^2 \right] + \frac{1}{24} (\boldsymbol{\omega}_o \times)^4 \right\} h^4. \quad (5.17)
\end{aligned}$$

Залишаючи члени до другого порядку мализни включно, одержуємо формулу для методу Пікара другого порядку:

$$P_{n+1} = E + (\boldsymbol{\omega}_o \times) \cdot h + [(\boldsymbol{\varepsilon} \times) + \frac{1}{2} (\boldsymbol{\omega}_o \times)^2] \cdot h^2. \quad (5.18)$$

Аналогічно одержимо формулу для методу Пікара третього порядку:

$$\begin{aligned}
P_{n+1} &= E + (\boldsymbol{\omega}_o \times) \cdot h + [(\boldsymbol{\varepsilon} \times) + \frac{1}{2} (\boldsymbol{\omega}_o \times)^2] \cdot h^2 + \\
&+ \left\{ (\mathbf{a} \times) + \frac{1}{3} [2(\boldsymbol{\omega}_o \times)(\boldsymbol{\varepsilon} \times) + (\boldsymbol{\varepsilon} \times)(\boldsymbol{\omega}_o \times)] + \frac{1}{6} (\boldsymbol{\omega}_o \times)^3 \right\} \cdot h^3. \quad (5.19)
\end{aligned}$$

5.1.4. Визначення коефіцієнтів інтерполяції для випадку вимірювань кутової швидкості

Нехай безпосередньо вимірюються значення проекцій кутової швидкості об'єкта, тобто вектор $\boldsymbol{\omega}$, на кожному кроці інтегрування. Значення цього вектору на поточному кроці ($t = t_n$, $\tau = 0$) безпосередньо вимірюється. Позначимо його $\boldsymbol{\omega}(0) = \boldsymbol{\omega}_o$.

Метод другого порядку. При побудові алгоритму другого порядку достатньо в апроксимації (12) залишити лише члени до другого порядку мализни (виключно) відносно малої величини τ :

$$\boldsymbol{\omega} = \boldsymbol{\omega}_o + 2\boldsymbol{\varepsilon} \cdot \tau, \quad (5.20)$$

На наступному кроці ($t = t_{n+1}$, $\tau = h$), у відповідності до (20) матимемо:

$$\boldsymbol{\omega}_1 = \boldsymbol{\omega}(h) = \boldsymbol{\omega}_o + 2\boldsymbol{\varepsilon} \cdot h.$$

Звідси визначаємо вектор $\boldsymbol{\varepsilon}$:

$$\boldsymbol{\varepsilon} = \frac{1}{2h} \nabla \boldsymbol{\omega}_1, \quad (5.21)$$

де позначено

$$\nabla \omega_1 = \omega_1 - \omega_o \quad (5.22)$$

- різниця першого порядку вимірюваних кутових швидкостей для кроку інтегрування $n+1$.

Тепер одержаних значень ω_o і ε достатньо, щоб реалізувати обчислення згідно формули (18) алгоритму другого порядку.

Метод третього порядку. Тепер потрібно апроксимувати кутову швидкість поліномом другого порядку

$$\omega = \omega_o + 2\varepsilon \cdot \tau + 3\mathbf{a} \cdot \tau^2. \quad (5.23)$$

У відповідності до нього на наступному кроці ($n+1$) значення кутової швидкості визначаються співвідношенням

$$\omega_1 = \omega_o + 2\varepsilon \cdot h + 3\mathbf{a} \cdot h^2,$$

а на попередньому ($n-1$, $t = t_{n-1}$, $\tau = -h$) – у такий спосіб:

$$\omega_{-1} = \omega_o - 2\varepsilon \cdot h + 3\mathbf{a} \cdot h^2.$$

Віднімаючи результати сусідніх вимірювань, дійдемо такої системи рівнянь для визначення невідомих ε і \mathbf{a} :

$$\begin{cases} 2h\varepsilon + 3h^2\mathbf{a} = \omega_1 - \omega_o = \nabla \omega_1 \\ 2h\varepsilon - 3h^2\mathbf{a} = \omega_o - \omega_{-1} = \nabla \omega_o \end{cases}, \quad (5.24)$$

де позначено: $\nabla \omega_o = \omega_o - \omega_{-1}$ - різниця першого порядку для кроку n . Віднімаючи з першого рівняння друге і вводячи поняття різниці другого порядку для кроку $n+1$

$$\nabla^2 \omega_1 = \nabla \omega_1 - \nabla \omega_o, \quad (5.25)$$

одержимо вираз для коефіцієнта \mathbf{a} :

$$\mathbf{a} = \frac{1}{6h^2} \nabla^2 \omega_1. \quad (5.26)$$

З першого рівняння (24) знайдемо вираз для другого невідомого коефіцієнта ε :

$$\varepsilon = \frac{1}{2h} \left(\nabla \omega_1 - \frac{1}{2} \nabla^2 \omega_1 \right). \quad (5.27)$$

Знання цих коефіцієнтів достатньо для обчислення за формулою (19) алгоритму методу третього порядку.

Метод четвертого порядку. У цьому випадку вектор кутової швидкості апроксимується поліномом третього порядку (12). Підставляючи в нього значення часу $\tau = h, 0, -h, -2h$ можна одержати три рівняння у різницях першого порядку:

$$\begin{cases} 2h\varepsilon + 3h^2\mathbf{a} + 4h^3\mathbf{b} = \boldsymbol{\omega}_1 - \boldsymbol{\omega}_o = \nabla\boldsymbol{\omega}_1 \\ 2h\varepsilon - 3h^2\mathbf{a} + 4h^3\mathbf{b} = \boldsymbol{\omega}_o - \boldsymbol{\omega}_{-1} = \nabla\boldsymbol{\omega}_o \\ 2h\varepsilon - 9h^2\mathbf{a} + 28h^3\mathbf{b} = \boldsymbol{\omega}_{-1} - \boldsymbol{\omega}_{o-2} = \nabla\boldsymbol{\omega}_{-1} \end{cases} .$$

Віднімаючи одне з одного сусідні рівняння, одержимо два рівняння у різницях другого порядку:

$$\begin{cases} 6h^2\mathbf{a} = \nabla\boldsymbol{\omega}_1 - \nabla\boldsymbol{\omega}_o = \nabla^2\boldsymbol{\omega}_1 \\ 6h^2\mathbf{a} - 24h^3\mathbf{b} = \nabla\boldsymbol{\omega}_o - \nabla\boldsymbol{\omega}_{-1} = \nabla^2\boldsymbol{\omega}_o \end{cases}$$

Віднімаючи з першого рівняння друге, знаходимо коефіцієнт \mathbf{b} :

$$24h^3\mathbf{b} = \nabla^2\boldsymbol{\omega}_1 - \nabla^2\boldsymbol{\omega}_o = \nabla^3\boldsymbol{\omega}_1; \quad \Rightarrow \quad \mathbf{b} = \frac{1}{24h^3} \nabla^3\boldsymbol{\omega}_1. \quad (5.28)$$

З першого рівняння одержуємо вираз для коефіцієнта \mathbf{a} :

$$\mathbf{a} = \frac{1}{6h^2} \nabla^2\boldsymbol{\omega}_1. \quad (5.29)$$

Нарешті з першого рівняння у різницях першого порядку відшукаємо коефіцієнт ε :

$$2h\varepsilon = \nabla\boldsymbol{\omega}_1 - 3h^2\mathbf{a} - 4h^3\mathbf{b}; \quad \Rightarrow \quad \varepsilon = \frac{1}{2h} \left(\nabla\boldsymbol{\omega}_1 - \frac{1}{2} \nabla^2\boldsymbol{\omega}_1 - \frac{1}{6} \nabla^3\boldsymbol{\omega}_1 \right). \quad (5.30)$$

Знання значень цих коефіцієнтів достатньо для проведення обчислень за алгоритмом (17) четвертого порядку.

5.1.5. Визначення коефіцієнтів інтерполяції для випадку вимірювань вектору псевдокоординат

Тепер перейдемо до випадку вимірювання на кожному кроці величин псевдокоординат, тобто інтегралів від проекцій кутової швидкості об'єкта на координатні осі:

$$\Theta(t_n, \tau) = \int_{t_n}^{t_n+\tau} \boldsymbol{\omega}(\xi) d\xi. \quad (5.31)$$

Безпосередньо на кожному кроці вимірюється величина

$$\Delta\Theta_{n+1} = \Theta(t_n, h) - \Theta(t_n, 0) = \int_{t_n}^{t_n+h} \boldsymbol{\omega}(\xi) d\xi. \quad (5.32)$$

Покладаючи попередню залежність (5.12) вектора кутової швидкості від часу, прийmemo, що залежність вектора псевдокоординат від часу має вид:

$$\Theta(t_n, \tau) = \boldsymbol{\omega}_o \cdot \tau + \varepsilon \cdot \tau^2 + \mathbf{a} \cdot \tau^3 + \mathbf{b} \cdot \tau^4. \quad (5.33)$$

Розпишемо значення цієї величини у моменти часу $\tau = h, 0, -h, -2h$

$$\Theta(t_n, h) = \Theta_1 = \boldsymbol{\omega}_o \cdot h + \varepsilon \cdot h^2 + \mathbf{a} \cdot h^3 + \mathbf{b} \cdot h^4;$$

$$\Theta(t_n, 0) = \Theta_o = 0;$$

$$\Theta(t_n, -h) = \Theta_{-1} = -\boldsymbol{\omega}_o \cdot h + \varepsilon \cdot h^2 - \mathbf{a} \cdot h^3 + \mathbf{b} \cdot h^4;$$

$$\Theta(t_n, -2h) = \Theta_{-2} = -2\omega_o \cdot h + 4\varepsilon \cdot h^2 - 8\mathbf{a} \cdot h^3 + 16\mathbf{b} \cdot h^4;$$

$$\Theta(t_n, -3h) = \Theta_{-3} = -3\omega_o \cdot h + 9\varepsilon \cdot h^2 - 27\mathbf{a} \cdot h^3 + 81\mathbf{b} \cdot h^4.$$

Через те, що на кожному кроці безпосередньо вимірюється кількість імпульсів протягом відповідного кроку, тобто різниці між сусідніми значеннями цієї величини, перейдемо до рівнянь відносно таких різниць:

$$\left\{ \begin{array}{l} \omega_o + h\varepsilon + h^2\mathbf{a} + h^3\mathbf{b} = \frac{1}{h} \nabla \Theta_1 = \frac{1}{h} (\Theta_1 - \Theta_0) \\ \omega_o - h\varepsilon + h^2\mathbf{a} - h^3\mathbf{b} = \frac{1}{h} \nabla \Theta_o = \frac{1}{h} (\Theta_o - \Theta_{-1}) \\ \omega_o - 3h\varepsilon + 7h^2\mathbf{a} - 15h^3\mathbf{b} = \frac{1}{h} \nabla \Theta_{-1} = \frac{1}{h} (\Theta_{-1} - \Theta_{-2}) \\ \omega_o - 5h\varepsilon + 19h^2\mathbf{a} - 65h^3\mathbf{b} = \frac{1}{h} \nabla \Theta_{-2} = \frac{1}{h} (\Theta_{-2} - \Theta_{-3}) \end{array} \right. \quad (5.34)$$

Ці рівняння і є основою для обчислення невідомих коефіцієнтів ω_o , ε , \mathbf{a} і \mathbf{b} .

Метод другого порядку. Для побудови метода другого порядку достатньо припустити наступну залежність від часу

$$\Theta = \omega_o \cdot \tau + \varepsilon \cdot \tau^2. \quad (5.35)$$

Тому замість рівнянь (34) можна розглядати впливаючи з них рівняння

$$\left\{ \begin{array}{l} \omega_o + h\varepsilon = \frac{1}{h} \nabla \Theta_1 \\ \omega_o - h\varepsilon = \frac{1}{h} \nabla \Theta_o \end{array} \right. ,$$

розв'язуючи які, знайдемо

$$\omega_o = \frac{1}{h} (\nabla \Theta_1 - \frac{1}{2} \nabla^2 \Theta_1); \quad \varepsilon = \frac{1}{2h^2} \nabla^2 \Theta_1 = \frac{1}{2h^2} (\nabla \Theta_1 - \nabla \Theta_o). \quad (5.36)$$

Метод третього порядку. Тепер припускатимемо наступну залежність від часу

$$\Theta = \omega_o \cdot \tau + \varepsilon \cdot \tau^2 + \mathbf{a} \cdot \tau^3. \quad (5.37)$$

Рівняння (34) замінюються на такі:

$$\left\{ \begin{array}{l} \omega_o + h\varepsilon + h^2\mathbf{a} = \frac{1}{h} \nabla \Theta_1 \\ \omega_o - h\varepsilon + h^2\mathbf{a} = \frac{1}{h} \nabla \Theta_o \\ \omega_o - 3h\varepsilon + 7h^2\mathbf{a} = \frac{1}{h} \nabla \Theta_{-1} \end{array} \right. \quad (5.38)$$

Віднімаючи сусідні рівняння, дійдемо двох рівнянь відносно різниць другого порядку

$$\begin{cases} 2h\varepsilon = \frac{1}{h}(\nabla\Theta_1 - \nabla\Theta_o) = \frac{1}{h}\nabla^2\Theta_1 \\ 2h\varepsilon - 6h^2\mathbf{a} = \frac{1}{h}(\nabla\Theta_o - \nabla\Theta_{-1}) = \frac{1}{h}\nabla^2\Theta_o \end{cases} \quad (5.39)$$

розв'язуючи які, знайдемо

$$\varepsilon = \frac{1}{2h^2}\nabla^2\Theta_1 = \frac{1}{2h^2}(\nabla\Theta_1 - \nabla\Theta_o); \quad \mathbf{a} = \frac{1}{6h^3}\nabla^3\Theta_1 = \frac{1}{6h^3}(\nabla^2\Theta_1 - \nabla^2\Theta_o). \quad (5.40)$$

У відповідності до першого рівняння (38), одержимо значення останнього коефіцієнта

$$\omega_o = \frac{1}{h}(\nabla\Theta_1 - \frac{1}{2}\nabla^2\Theta_1 - \frac{1}{6}\nabla^3\Theta_1). \quad (5.41)$$

Метод четвертого порядку. Побудуємо на основі рівнянь (34) рівняння у різницях другого порядку

$$\begin{cases} 2h\varepsilon + 2h^3\mathbf{b} = \frac{1}{h}(\nabla\Theta_1 - \nabla\Theta_o) = \frac{1}{h}\nabla^2\Theta_1 \\ 2h\varepsilon - 6h^2\mathbf{a} + 14h^3\mathbf{b} = \frac{1}{h}(\nabla\Theta_o - \nabla\Theta_{-1}) = \frac{1}{h}\nabla^2\Theta_o \\ 2h\varepsilon - 12h^2\mathbf{a} + 50h^3\mathbf{b} = \frac{1}{h}(\nabla\Theta_{-1} - \nabla\Theta_{-2}) = \frac{1}{h}\nabla^2\Theta_{-1} \end{cases} \quad (5.42)$$

Аналогічно побудуємо рівняння у різницях третього порядку

$$\begin{cases} 6h^2\mathbf{a} - 12h^3\mathbf{b} = \frac{1}{h}(\nabla^2\Theta_1 - \nabla^2\Theta_o) = \frac{1}{h}\nabla^3\Theta_1 \\ 6h^2\mathbf{a} - 36h^3\mathbf{b} = \frac{1}{h}(\nabla^2\Theta_o - \nabla^2\Theta_{-1}) = \frac{1}{h}\nabla^3\Theta_o \end{cases} \quad (5.43)$$

З (43) випливає:

$$\mathbf{b} = \frac{1}{24h^4}\nabla^4\Theta_1; \quad \mathbf{a} = \frac{1}{6h^3}(\nabla^3\Theta_1 + \frac{1}{2}\nabla^4\Theta_1). \quad (5.44)$$

Перше рівняння (42) дозволяє визначити коефіцієнт ε :

$$\varepsilon = \frac{1}{2h^2}(\nabla^2\Theta_1 - \frac{1}{12}\nabla^4\Theta_1),$$

а перше рівняння (34) - коефіцієнт ω_o :

$$\omega_o = \frac{1}{h}(\nabla\Theta_1 - \frac{1}{2}\nabla^2\Theta_1 - \frac{1}{6}\nabla^3\Theta_1 - \frac{1}{12}\nabla^4\Theta_1). \quad (5.45)$$

5.1.6. Побудова алгоритмів

Поєднаємо одержані результати у єдиний алгоритм окремо для другого, третього та четвертого порядків. Для цього достатньо підставити одержані значення коефіцієнтів у вирази (5.17), (5.18) та (5.19) відповідно до порядків алгоритмів і залишити в одержаних виразах члени порядку малізми, не вище за порядок методу. При цьому зважимо, що скінченна різниця першого порядку між

вимірними послідовними значеннями є пропорційною кроку вимірювання, а тому є малою першого порядку мализни по відношенню до малого кроку інтегрування. Неважко показати, що скінченна різниця наступного порядку є малою більш високого порядку мализни, аніж різниця попереднього порядку. Тобто слід вважати, що порядок мализни скінченної різниці збігається з порядком самої різниці. Тому при підстановці слід це враховувати, вилучаючи члени порядку мализни, вищого за потрібний порядок алгоритму.

Алгоритми другого порядку. У такий спосіб з попереднього виходить наступний алгоритм методу другого порядку:

$$P_{n+1} = E + (\boldsymbol{\omega}_o \times) \cdot h + [(\boldsymbol{\varepsilon} \times) + \frac{1}{2} (\boldsymbol{\omega}_o \times)^2] \cdot h^2.$$

Отже для випадку вимірювання кутової швидкості алгоритм матиме вид:

$$P_{n+1} = E + [(\boldsymbol{\omega}_o \times) + \frac{1}{2} (\nabla \boldsymbol{\omega}_1 \times)] \cdot h + \frac{1}{2} (\boldsymbol{\omega}_o \times)^2 \cdot h^2. \quad (5.46)$$

Для випадку вимірювання приростів псевдокоординат вираз алгоритму визначиться у такий спосіб:

$$P_{n+1} = E + (\nabla \times) + \frac{1}{2} [(\nabla \times) - \frac{1}{2} (\nabla^2 \times)] [(\nabla \times) - \frac{1}{2} (\nabla^2 \times)] \approx E + (\nabla \times) + \frac{1}{2} (\nabla \times)^2. \quad (5.47)$$

Тут і у подальшому для спрощення виразів використовуються позначення:

$$\nabla \Theta_1 = \nabla, \quad \nabla^2 \Theta_1 = \nabla^2, \quad \nabla^3 \Theta_1 = \nabla^3, \quad \nabla^4 \Theta_1 = \nabla^4.$$

Алгоритми третього порядку. Перейдемо до формулювання алгоритмів третього порядку на основі формули (5.19):

$$P_{n+1} = E + [(\boldsymbol{\omega}_o h + \boldsymbol{\varepsilon} h^2 + \mathbf{a} h^3) \times] + \frac{1}{2} (\boldsymbol{\omega}_o \times)^2 \cdot h^2 + \frac{1}{3} [2(\boldsymbol{\omega}_o \times)(\boldsymbol{\varepsilon} \times) + (\boldsymbol{\varepsilon} \times)(\boldsymbol{\omega}_o \times)] h^3 + \frac{1}{6} (\boldsymbol{\omega}_o \times)^3 \cdot h^3.$$

Якщо врахувати (5.26) і (5.27), з цього виразу одержимо наступну формулу для алгоритму у випадку вимірювання проекцій кутової швидкості:

$$P_{n+1} = E + [\boldsymbol{\omega}_o + \frac{1}{2} \nabla \boldsymbol{\omega}_1 - \frac{1}{12} \nabla^2 \boldsymbol{\omega}_1] \times h + \frac{1}{2} (\boldsymbol{\omega}_o \times)^2 \cdot h^2 + \frac{1}{3} [(\boldsymbol{\omega}_o \times)(\nabla \boldsymbol{\omega}_1 \times) + \frac{1}{2} (\nabla \boldsymbol{\omega}_1 \times)(\boldsymbol{\omega}_o \times)] h^2 + \frac{1}{6} (\boldsymbol{\omega}_o \times)^3 \cdot h^3. \quad (5.48)$$

Якщо ж скористатися значеннями (5.40) і (5.41), одержимо алгоритм для випадку вимірювання приростів псевдокоординат:

$$P_{n+1} = E + (\nabla \times) + \frac{1}{2} ((\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3) \times)^2 + \frac{1}{3} [((\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3) \times)(\nabla^2 \times) + (\frac{1}{2} \nabla^2 \times)((\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3) \times)] + \frac{1}{6} [(\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3) \times]^3$$

Залишаючи члени не вище відповідно до порядку алгоритму, остаточно одержуємо :

$$P_{n+1} \approx E + (\nabla \times) + \frac{1}{2} (\nabla \times)^2 + \frac{1}{12} [(\nabla \times)(\nabla^2 \times) - (\nabla^2 \times)(\nabla \times)] + \frac{1}{6} (\nabla \times)^3. \quad (5.49)$$

Алгоритми четвертого порядку. Формули алгоритмів четвертого порядку виводяться аналогічно на основі формули (5.17) шляхом підстановки коефіцієнтів (5.28), (5.29), (5.30) для випадку вимірювання проекцій кутової швидкості та (5.44), (5.45) для випадку вимірювання псевдокоординат.

5.2. Засоби моделювання алгоритмів

У цілях моделювання алгоритмів утворимо процедури, що здійснюють обчислення на кожному кроці інтегрування згідно розроблених алгоритмів.

При моделюванні для випадку вимірювання проекцій кутової швидкості алгоритми реалізуються за допомогою процедур Picar2mm, Picar3mm, Picar4mm, тексти яких приведені нижче.

```
function y=Picar2mm (h,C,omo,omo_1)
% Вычисление матрицы направляющих косинусов через шаг интегрирования
% для случая измерений проекций угловых скоростей
E=eye(3,3);
OMM_1=Vect2KSM(omo_1);
OMM=Vect2KSM(omo);
dt1=OMM_1-OMM;
eps=dt1/h;
dC=E+OMM*h+eps*h^2/2+OMM^2*h^2/2;
y=C*dC;
```

```
function y = Picar3mm (h,C1,C,omo1,omo,omo_1)
% Вычисление матрицы направляющих косинусов через шаг интегрирования
% для случая измерений проекций угловых скоростей
E=eye(3,3);
OMM_1=Vect2KSM(omo_1);
OMM=Vect2KSM(omo);
OMM1=Vect2KSM(omo1);
dt1_1=OMM_1-OMM; dt1=OMM-OMM1;
dt2_1=dt1_1-dt1;
eps=(dt1_1+dt1)/(2*h);
a=dt2_1/(2*h^2);
dC=E+OMM*h+eps*h^2/2+a*h^3/3+...
    OMM^2*h^2/2+(2*OMM*eps+eps*OMM)*h^3/6;
y=C*dC;
```

```
function y=Picar4mm (h,C2,C1,C,omo2,omo1,omo,omo_1)
% Вычисление матрицы направляющих косинусов через шаг интегрирования
% для случая измерений проекций угловых скоростей
E=eye(3,3);
OMM_1=Vect2KSM(omo_1);
OMM=Vect2KSM(omo);
OMM1=Vect2KSM(omo1);
OMM2=Vect2KSM(omo2);
dt1_1=OMM_1-OMM; dt1=OMM-OMM1; dt11=OMM1-OMM2;
dt2_1=dt1_1-dt1; dt2=dt1-dt11;
```

```

dt3_1=dt2_1-dt2;
eps=(dt1_1-dt2_1/2-dt3_1/6)/h;
a=dt2_1/(2*h^2);
b=dt3_1/(6*h^3);
dC=E+OMM*h+eps*h^2/2+a*h^3/3+b*h^4/4+...
    OMM^2*h^2/2+(2*OMM*eps+eps*OMM)*h^3/6+...
    (3*OMM*a+a*OMM)*h^4/12+eps^2*h^4/8;
y=C*dC;

```

Вхідними величинами цих процедур є:

- 1) крок інтегрування h ;
- 2) значення матриці C напрямних косинусів у кількох попередніх моментах часу;
- 3) значення вектора $o_{\text{мо}}$ кутової швидкості основи у відповідні моменти часу i у той момент часу, для якого визначаються параметри орієнтації ($o_{\text{мо}_1}$).

Вихідна величина y - матриця значень матриці напрямних косинусів у наступний момент часу.

Наведені програми процедур використовують процедуру Vect2KSM, яка реалізує перетворення вектора в кососиметричну матрицю. Текст цієї процедури поданий у додатку.

При моделюванні алгоритмів у випадку вимірювання псевдокоординат застосовуються процедури PicPsev2.m, PicPsev3.m, PicPsev4.m.

```

function y=PicPsev2(h,C,k,k_1)
% Вычисление матрицы направляющих косинусов через шаг интегрирования
% для случая измерений проекций угловых скоростей

```

```

E=eye(3,3);
dpk_1=Vect2KSM(k_1);
dpk=Vect2KSM(k);
dpk2_1=dpk_1-dpk;
om=(dpk_1-dpk2_1/2)/h;
eps=dpk2_1/h^2;
dC=E+om*h+eps*h^2/2+om^2*h^2/2;
y=C*dC;

```

```

function y=PicPsev3(h,C1,C,k1,k,k_1)
% Вычисление матрицы направляющих косинусов через шаг интегрирования
% для случая измерений проекций угловых скоростей

```

```

E=eye(3,3);
dpk_1=Vect2KSM(k_1);
dpk=Vect2KSM(k);
dpk1=Vect2KSM(k1);
dpk2_1=dpk_1-dpk; dpk2_2=dpk-dpk1;
dpk3_1=dpk2_1-dpk2_2;
om=(dpk_1-dpk2_1/2-dpk3_1/6)/h;
eps=dpk2_1/h^2;
a=dpk3_1/(2*h^3);
dC=E+om*h+eps*h^2/2+a*h^3/3+...
    om^2*h^2/2+(2*om*eps+eps*om)*h^3/6;
y=C*dC;

```

```

function y=PicPsev4(h,C2,C1,C,k2,k1,k,k_1)
% Вычисление матрицы направляющих косинусов через шаг интегрирования

```

% для случая измерений проекций угловых скоростей

```

E=eye(3,3);
dpk_1=Vect2KSM(k_1);
dpk=Vect2KSM(k);
dpk1=Vect2KSM(k1);
dpk2=Vect2KSM(k2);
dpk2_1=dpk_1-dpk; dpk2_2=dpk-dpk1; dpk2_3=dpk1-dpk2;
dpk3_1=dpk2_1-dpk2_2; dpk3_2=dpk2_2-dpk2_3;
dpk4_1=dpk3_1-dpk3_2;
om=(dpk_1-dpk2_1/2-dpk3_1/6-dpk4_1/12)/h;
eps=(dpk2_1-dpk4_1/12)/h^2;
a=(dpk3_1+dpk4_1/2)/(2*h^3);
b=dpk4_1/(6*h^4);
dC=E+om*h+eps*h^2/2+a*h^3/3+b*h^4/4+...
    om^2*h^2/2+(2*om*eps+eps*om)*h^3/6+...
    (3*om*a+a*om)*h^4/12+eps^2*h^4/8;
y=C*dC;

```

Вхідними величинами цих процедур є:

- 4) крок інтегрування h ;
- 5) значення матриці C напрямних косинусів у кількох попередніх моментах часу;
- 6) значення k приростів псевдокоординат у відповідні моменти часу і у той момент часу, для якого визначаються параметри орієнтації (k_1).

Вихідна величина y - матриця значень матриці напрямних косинусів у наступний момент часу.

5.3. Опис програмних засобів

Програма моделювання процесу накопичення похибок розроблених алгоритмів має містити в себе імітацію кутового руху основи з розрахунками його параметрів орієнтації у дискретні моменти часу; здійснення розрахунків на кожному кроці вимірювання за розробленими алгоритмами і обчислення похибок алгоритмів шляхом віднімання "точних" значень відповідних параметрів орієнтації від тих, що одержані шляхом застосування алгоритмів.

Керуюча програма *Picar_Poisson_om_upr.m* ініціює роботу сукупності програмних одиниць, яка забезпечує обчислення похибок чисельного інтегрування матричного рівняння Пуассона трьома алгоритмами методу Пікара— другого, третього і четвертого порядків.

Використовуються наступні процедури й функції:

- 1) *idbso2UgSk.m* – процедура змінювання значень вхідних даних для програми *Picar_Poisson_om_upr.m*;
- 2) *ugdvoim.m* – функція обчислення параметрів поточного кутового руху основи (кутів, кутових швидкостей, кутових прискорень);
- 3) *Picar2mm.m* – процедура чисельного інтегрування методом Пікара другого порядку (по двох попередніх точках);
- 4) *Picar3mm.m* – процедура чисельного інтегрування методом Пікара третього порядку (по трьох попередніх точках);

5) *Picar4mm.m* – процедура чисельного інтегрування методом Пікара четвертого порядку (по чотирьох попередніх точках)

7) *grafik91.m* – процедура побудови графіків 9 величин як функцій від однієї величини на трьох графічних підвікнах в одному графічному вікні один під одним;

8) *dngrbso3.m* – процедура виведення на поле графіків розрахункових даних у текстовій формі.

Текст головної (керуючої) програми *Picar_Poisson_om_upr.m* наведений нижче.

```
% Picar_Poisson_om_upr
% Програма дослідження похибок алгоритма чисельного інтегрування
% 1) кінематичних рівнянь орієнтації Пуассона
% 2) метод Пікара
% 3) шляхом використання вимірювальної інформації про проекції кутової
% швидкості основи
% Використовуються наступні процедури й функції:
% idbso2UgSk.m - процедура змінювання значень вхідних даних для
% програми Picar_Poisson_om_upr
% ugdvobm.m - функція обчислення параметрів поточного кутового руху
% основи (кутів, кутових швидкостей, кутових прискорень)
% Picar2mm.m - процедура чисельного інтегрування методом Пікара
% 2-го порядку (по двох попередніх точках)
% Picar3mm.m - процедура чисельного інтегрування методом Пікара
% 3-го порядку (по трьох попередніх точках)
% Picar4mm.m - процедура чисельного інтегрування методом Пікара
% 4-го порядку (по чотирьох попередніх точках)
% grafik91.m - процедура побудови графіків 9 величин як функцій
% від однієї величини на трьох підграфіках один під одним
% dngrbso3.m - процедура виведення на поле графіків розрахункових
% даних у текстовій формі

% Лазарев Ю. Ф., Бобровицька Я. 12-03-2006 р.
echo off
clc
clear all
global sprogram sname
global D2
global D5
sprogram='Picar-Poisson-om-upr';
clc
disp(' Ви увійшли в програму Picar_Poisson_om_upr,')
disp('яка інтегрує кінематичні рівняння Ейлера Безплатформової Системи Орієнтації')
disp(' і обчислює похибки цього чисельного інтегрування')
disp(' ')
k0=menu('Що робити?','Продовжити роботу','Закінчити роботу');
if k0==1 % Вхід у програму
    sname='Бобровицька Я.';
    % Введення "вшитих" значень первісних параметрів руху основи
    ps0=0; te0=0; fi0=0;
    psm=0.1; tem=psm; fim=psm;
    omps=2*pi/10; omte=omps; omfi=omps;
    eps=0; ete=0; efi=0;
    % параметрів інтегрування;
    h=0.2; 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 для ПОВТОРУ ЕКСПЕРИМЕНТІВ
while k0==1
  idbso2UgSk; %БЛОК ЗМІНЮВАННЯ ДАНИХ;
  % Визначення мінімального Тi и максимального Та періодів
  qwe=[D2(7) D2(8) D2(9)]; % Формування масиву частот кутових коливань
  Ti = 2*pi/max(qwe); Та=Ti;
  for i=1:3,
    if D2(6+i)>0
      T1=2*pi/D2(6+i);
      if T1>Та Та=T1; end
      if T1<Ti Ti=T1; end
    end
  end
  % Визначення величини кроку інтегрування h
  h = D4(1);
  % Визначення моменту Тn0 часу початку формування
  % вектора масивів даних вимірювань
  Тn0 = Та*D4(2);
  % Визначення кінцевого моменту часу tfinal інтегрування
  tfinal= Та*(D4(2)+D4(3));
  D5=[h Тn0 tfinal];
  % БЛОК ІНТЕГРУВАННЯ
  %%%%%%%%%%%
  t=0; % Завдання початкового моменту часу
  % Завдання вектора початкових умов у попередніх точках
  %x=ugdvobm(t-3*h); % Параметри руху ОСНОВИ у момент часу t-3*h
  %ys3=[-x(1) x(2) x(3)]; om3=[x(7) x(8) x(9)];
  x=ugdvobm(t-2*h); % Параметри руху ОСНОВИ у момент часу t-2*h
  %ys2=[-x(1) x(2) x(3)]; om2=[x(7) x(8) x(9)];
  x=ugdvobm(t-h); % Параметри руху ОСНОВИ у момент часу t-h
  %ys1=[-x(1) x(2) x(3)]; om1=[x(7) x(8) x(9)];
  x=ugdvobm(t); % Параметри руху ОСНОВИ у поточний момент часу t
  %ys=[-x(1) x(2) x(3)]; om=[x(7) x(8) x(9)];
  x=ugdvobm(t+h); % Параметри руху ОСНОВИ у наступний момент часу t+h
  %ys_1=[-x(1) x(2) x(3)]; om_1=[x(7) x(8) x(9)];
  kc=[2,3,1];
  % Обчислення матриць напрямних косинусів
  C2s=ug2mnk(ys,kc); % для метода 2-го порядку
  C3s1=ug2mnk(ys1,kc); C3s=C2s; % для метода 3-ого порядку
  C4s2=ug2mnk(ys2,kc); C4s1=C3s1; C4s=C3s; % для 4-ого порядку
  imas=1; % Лічильник елементів вихідних масивів
  % Цикл інтегрування
  %%%%%%%%%%%
  disp('Почекайте (йде обчислення)')
  while (imas-1)*h<tfinal
    t=(imas-1)*h;
    %%%%%%%%%%% Метод Пікара другого порядку
    C2 = Picar2mm(h,C2s,om,om_1);
    C2s = C2;
    %%%%%%%%%%% Метод Пікара третього порядку
    C3 = Picar3mm(h,C3s1,C3s,om1,om,om_1);
    C3s1=C3s; C3s = C3;
    %%%%%%%%%%% Метод Пікара четвертого порядку
    C4 = Picar4mm(h,C4s2,C4s1,C4s,om2,om1,om,om_1);
    C4s2 = C4s1; C4s1 = C4s; C4s = C4;
    om2=om1; om1=om; om=om_1;
    x=ugdvobm(t+h);
    % для метода другого порядку
    y2=MNK2Ug(C2,kc);
    dps2(imas)=y2(1)+x(1); % Масиви похибок
    dteta2(imas)=y2(2)-x(2); % оцінювання
    dfi2(imas)=y2(3)-x(3); % кутів повороту основи
  end
end

```

```

% для метода третього порядку
y3=MNK2Ug(C3,кc);
dpsi3(imas)=y3(1)+x(1);           % Масиви похибок
dteta3(imas)=y3(2)-x(2);         % оцінювання
dfi3(imas)=y3(3)-x(3);          % кутів повороту основи
% для Адамса-Башфорта
y4=MNK2Ug(C4,кc);
dpsi4(imas)=y4(1)+x(1);           % Масиви похибок
dteta4(imas)=y4(2)-x(2);         % оцінювання
dfi4(imas)=y4(3)-x(3);          % кутів повороту основи
x=ugdвobm(t+2*h); % Параметри руху ОСНОВИ у наступний момент часу t+2h
ys_1=[x(1) x(2) x(3)];  om_1=[x(7) x(8) x(9)];
tt(imas)=t;               % Массив моментів времени
imas=imas+1;
end
% Побудова Графіків
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
stit1='ПОХИБКИ БСО (кутові мінути) - МЕТОД ПІКАРА';
sy1ab1='Метод 2-го порядку';  sx1ab1=' ';
stit2=' ';  sy1ab2='Метод 3-го порядку';  sx1ab2=' ';
stit3=' ';  sy1ab3='Метод 4-го порядку';  sx1ab3='Час, с';
grafik91(tt,dpsi2*180*60/pi,dteta2*180*60/pi,dfi2*180*60/pi,stit1,sy1ab1,sx1ab1,...
dpsi3*180*60/pi,dteta3*180*60/pi,dfi3*180*60/pi,stit2,sy1ab2,sx1ab2,...
dpsi4*180*60/pi,dteta4*180*60/pi,dfi4*180*60/pi,stit3,sy1ab3,sx1ab3);
dngrbso3; % Виведення на поле графіків розрахункових даних
% Кінець процедури виведення графіків
k0=menu('Що робити?','Продовжити роботу','Закінчити роботу');
end
end % Вихід з циклу програми
clear global
disp('Кінець програми')

```

Допоміжні програми *idbso2UgSk.m*, *dngrbso3.m*, *ugdвobm.m*, *grafik91.m* є аналогічними до тих самих програм, наведених у попередньому розділі 4.

Головна програма *Picar_Poisson_Pseudo_upr* ініціює роботу сукупності програмних одиниць, що забезпечує обчислення похибок чисельного інтегрування матричного рівняння Пуассона у випадку вимірювання приростів псевдокоординат трьома алгоритмами методу Пікара. Використовуються такі процедури й функції:

- 1) *idbso2Pseudo.m* – процедура змінювання значень вхідних даних для програми *Picar_Poisson_Pseudo_upr*;
- 2) *UgDvObPseudo.m* – функція обчислення параметрів поточного кутового руху основи (кутів, кутових швидкостей, кутових прискорень, приростів псевдокоординат);
- 3) *PicPsev2.m* – процедура чисельного інтегрування методом Пікара другого порядку (по двох попередніх точках);
- 4) *PicPsev3.m* – процедура чисельного інтегрування методом Пікара третього порядку (по трьох попередніх точках);
- 5) *PicPsev4.m* – процедура чисельного інтегрування методом Пікара четвертого порядку (по чотирьох попередніх точках);
- 7) *grafik91.m* – процедура побудови графіків 9 величин як функцій від однієї величини на трьох графічних підвікнах в одному графічному вікні один під одним;

8) *dngrbso2.m* – процедура виведення на поле графіків розрахункових даних у текстовій формі.

Текст головної (керуючої) програми *Picar_Poisson_om_upr.m* наведений нижче.

```
% Picar_Poisson_Pseudo_upr
% Програма Picar_Poisson_Pseudo_upr
% дослідження похибок алгоритма чисельного інтегрування
% 1) кінематичних рівнянь орієнтації Пуассона
% 2) методами другого, третього і четвертого порядку
% 3) шляхом використання вимірювальної інформації про прирости
% псевдокоординат (інтегралів від проекцій кутової швидкості основи)
% Використовуються наступні процедури й функції:
% idbso2Pseudo.m - процедура змінювання значень вхідних даних для
% програми BSOPseudo_uprNm
% UgDvObPseudo.m - функція обчислення параметрів поточного кутового руху
% основи (кутів, кутових швидкостей, кутових прискорень, псевдокоординат)
% PicPsev2.m - процедура чисельного інтегрування методом Пікара
% 2-го порядку (по двох попередніх точках)
% PicPsev3.m - процедура чисельного інтегрування методом Пікара
% 3-го порядку (по трьох попередніх точках)
% PicPsev4.m - процедура чисельного інтегрування методом Пікара
% 4-го порядку (по чотирьох попередніх точках)
% grafik91.m - процедура побудови графіків 9 величин як функцій
% від однієї величини на трьох підграфіках один під одним
% dngrbso2.m - процедура виведення на поле графіків розрахункових
% даних у текстовій формі

% Лазарєв Ю. Ф., Бобровицька Я. 27-03-2006 р.
echo off, clc, clear all
global sprogram sname
global D2
global D5
sprogram='Picar-Poisson-Pseudo-upr.m'; clc
disp(' Ви увійшли в програму Picar_Poisson_Pseudo_upr,')
disp('яка інтегрує кінематичні рівняння у кватерніонах')
disp('по вимірних псевдокоординатах і обчислює похибки цього чисельного інтегру-
вання'), disp(' ')
k0=menu('Що робити?','Продовжити роботу','Закінчити роботу');
if k0==1 % Вхід у програму
    sname='Бобровицька Я.';
    % Введення "вшитих" значень первісних параметрів руху основи
    ps0=0; te0=0; fi0=0;
    psm=0.1; tem=psm; fim=psm;
    omps=2*pi/10; omte=omps; omfi=omps;
    eps=0; ete=pi/2; efi=0;
    % параметрів інтегрування;
    h=0.2; 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 для ПОВТОРУ ЕКСПЕРИМЕНТІВ
    while k0==1
        idbso2Pseudo; % БЛОК ЗМІНЮВАННЯ ДАНИХ;
        % Визначення мінімального Tі и максимального Tа періодів
        qwe=[D2(7) D2(8) D2(9)]; % Формування масиву частот кутових коливань
        Tі = 2*pi/max(qwe); Tа=Tі;
        for i=1:3,
            if D2(6+i)>0
                T1=2*pi/D2(6+i);
```

```

    if T1>Ta Ta=T1; end
    if T1<Ti Ti=T1; end
end
end
% Визначення величини кроку інтегрування h
h = D4(1);
% Визначення моменту Tn0 часу початку формування
% вектора масивів даних вимірювань
Tn0 = Ta*D4(2);
% Визначення кінцевого моменту часу tfinal інтегрування
tfinal= Ta*(D4(2)+D4(3));
D5=[h Tn0 tfinal];
%
% БЛОК ІНТЕГРУВАННЯ
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t=0; % Завдання початкового моменту часу
% Завдання вектора початкових умов у попередніх точках
z=UgDvObPseudo(t-2*h,h); % Пар-ри руху ОСНОВИ у момент часу два кроки назад
pk2=[z(4) z(5) z(6)]; % Прирости псевдокоординат у момент часу на два кроки назад
ys2=[-z(1) z(2) z(3)];
z=UgDvObPseudo(t-h,h); % Параметри руху ОСНОВИ у попередній точці
pk1=[z(4) z(5) z(6)]; % Прирости псевдокоординат у попередній момент часу
ys1=[-z(1) z(2) z(3)];
z=UgDvObPseudo(t,h); % Параметри руху ОСНОВИ у поточній точці
pk=[z(4) z(5) z(6)]; % Прирости псевдокоординат у поточний момент часу
ys=[-z(1) z(2) z(3)];
z=UgDvObPseudo(t+h,h); % Параметри руху ОСНОВИ у наступній точці
pk_1=[z(4) z(5) z(6)]; % Прирости псевдокоординат у наступний момент часу
kc=[2 3 1];
C2s=ug2mnk(ys,kc); % для метода 2-го порядку
C3s1=ug2mnk(ys1,kc); C3s=C2s; % для метода 3-ого порядку
C4s2=ug2mnk(ys2,kc); C4s1=C3s1; C4s=C3s; % для 4-ого порядку
imas=1; % Лічильник елементів вихідних масивів
% Цикл інтегрування
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('Почекайте (іде обчислення)')
while (imas-1)*h<tfinal
    t=(imas-1)*h;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Метод Пікара другого порядку
    C2 = PicPsev2(h,C2s,pk,pk_1);
    C2s = C2;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Метод Пікара третього порядку
    C3 = PicPsev3(h,C3s1,C3s,pk1,pk,pk_1);
    C3s1=C3s; C3s = C3;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Метод Пікара четвертого порядку
    C4 =PicPsev4(h,C4s2,C4s1,C4s,pk2,pk1,pk,pk_1);
    C4s2 = C4s1; C4s1 = C4s; C4s = C4;
    pk2=pk1; pk1=pk; pk=pk_1;
    z=UgDvObPseudo(t+h,h);
    % для метода другого порядку
    y2=MNK2Ug(C2,kc);
    dps2(imas)=y2(1)+z(1); % Масиви похибок
    dteta2(imas)=y2(2)-z(2); % оцінювання
    dfi2(imas)=y2(3)-z(3); % кутів повороту основи
    % для метода третього порядку
    y3=MNK2Ug(C3,kc);
    dps3(imas)=y3(1)+z(1); % Масиви похибок
    dteta3(imas)=y3(2)-z(2); % оцінювання
    dfi3(imas)=y3(3)-z(3); % кутів повороту основи
    % для Адамса-Башфорта
    y4=MNK2Ug(C4,kc);
    dps4(imas)=y4(1)+z(1); % Масиви похибок
    dteta4(imas)=y4(2)-z(2); % оцінювання

```

```

dfi4(imas)=y4(3)-z(3); % кутів повороту основи
%x=ugdovobm(t+2*h); % Параметри руху ОСНОВИ у наступний момент часу t+2h
z=UgDvObPseudo(t+2*h,h);
ys_1=[-z(1) z(2) z(3)]; pk_1=[z(4) z(5) z(6)];
tt(imas)=t; % Масив моментів часу
imas=imas+1;
end
% Побудова Графіків
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
stit1='ПОХИБКИ БСО (кутові мінути) - МЕТОД ПІКАРА';
syllab1='Метод 2-го порядку';
sxlab1=' ';
stit2=' ';
syllab2='Метод 3-го порядку';
sxlab2=' ';
stit3=' ';
syllab3='Метод 4-го порядку';
sxlab3='Час, с';
grafik91(tt,dpsi2*180*60/pi,dteta2*180*60/pi,dfi2*180*60/pi,stit1,syllab1,sxlab1,...
dpsi3*180*60/pi,dteta3*180*60/pi,dfi3*180*60/pi,stit2,syllab2,sxlab2,...
dpsi4*180*60/pi,dteta4*180*60/pi,dfi4*180*60/pi,stit3,syllab3,sxlab3);
dngrbso2; % 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')

```

Процедури *idbso2Pseudo.m*, *dngrbso2.m*, *grafik91.m* також є аналогічними до тих самих програм, наведених у попередньому розділі 4.

Текст процедури *UgDvObPseudo.m* наведений нижче.

```

function z=UgDvObPseudo(t,h)
% Процедура обчислення поточних кутів повороту ОСНОВИ
% і приростів псевдокоординат на попередньому кроці
% Вхідні параметри
% t - поточний час;
% h = крок інтегрування
% Kt, Km, omK, eK - параметри руху з кута курсу
% ( K(t) = Kt*t + Km*sin(omK*t+eK) );
% Tt, Tm, omT, eT - параметри руху з кута тангажа teta
% ( teta(t) = Tt*t + Tm*sin(omT*t+eT) );
% Gt, Gm, omG, eG - параметри руху з кута крена gamma
% ( gamma(t) = Gt*t + Gm*sin(omG*t+eG) );
% Вихідні параметри:
% z(1) - кут курсу; z(2) - кут тангажа; z(3) - кут крена у момент часу t
% z(4), z(5), z(6) - псевдокоординати відносно осей X, Y і Z відповідно;
% Псевдокоордината = це інтеграл з часу від проекції кутової швидкості
% тіла на відповідну координатну вісь у діапазоні від t-h до t

% Лазарев Ю. Ф. 14=03-2006

global D2
tt=[t,t-h/2,t-h];
for k=1:3
ttd=tt(k);
for i=1:3
f=D2(i+6)*ttd+D2(i+9);
x(i)=D2(i)*ttd+D2(i+3)*sin(f); % Обчислення кутів
if k==1
z(i)=x(i);

```

```

    end
    x(i+3)=D2(i)+D2(i+6)*D2(i+3)*cos(f); % Обчислення кутових швидкостей
end
% Обчислення проекцій кутової швидкості
sg=sin(x(3));cg=cos(x(3)); st=sin(x(2)); ct=cos(x(2));
x1(k) = x(6) - x(4)*st;
x2(k) = - x(4)*ct*cg + x(5)*sg;
x3(k) = x(4)*ct*sg + x(5)*cg;
end
% Далі йдуть формули Сімпсона чисельного інтегрування проекцій кутової швидкості
% на осі системи координат
z(4)=(x1(1)+4*x1(2)+x1(3))*h/6;
z(5)=(x2(1)+4*x2(2)+x2(3))*h/6;
z(6)=(x3(1)+4*x3(2)+x3(3))*h/6;

```

На відміну від процедури *ugdvo**m***, яка використовувалася раніше, процедура *UgDvObPseudo.m* обчислює не вектор проекцій кутової швидкості основи, а вектор приростів псевдокоординат на відповідному кроці інтегрування (вимірювання). Вхідними величинами тут є значення t моменту часу, для якого обчислюється значення кутів повороту об'єкта і приростів псевдокоординат, і крок інтегрування h . Вихідною величиною є вектор із шести компонентів. Перші три визначають значення кутів повороту об'єкта, а решта три - прирости псевдокоординат. Прирости псевдокоординат чисельно обчислюються за формулою Сімпсона як інтеграли від проекцій кутової швидкості на поточному інтервалі інтегрування.

5.4. Результати моделювання

На рис. 5.1...5.8 наведені результати роботи програм моделювання, описаних вище. Рис. 5.1...5.4 подають графіки залежності похибок чисельного інтегрування рівняння Пуассона при вимірюванні проекцій кутової швидкості основи, а рис. 5.5... 5.8 - при вимірюванні приростів псевдокоординат.

Перші два рисунки з кожної четвірки відповідають випадку синхронної і синфазної хитавиці основи з усіх трьох кутів, решта два – випадку синхронних коливань із зсувом фаз 90° . Моделювання з кожного з вказаних випадків здійснено для двох значень кроку інтегрування $h = 0,2$ с і $h = 0,02$ с.

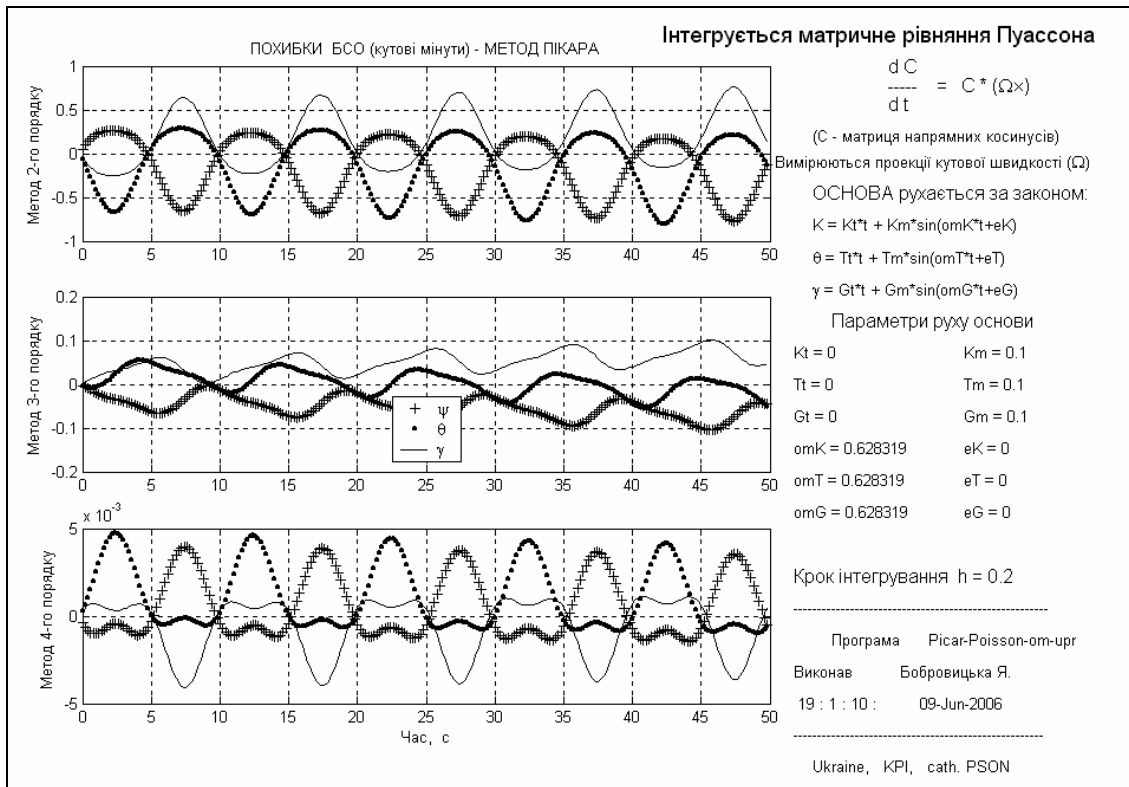


Рис. 5.1. Похибки алгоритмів методу Пікара для рівняння Пуассона ($h=0.2c$) при синхронних і синфазних коливаннях основи з трьох кутів (вимірюються проєкції кутової швидкості)

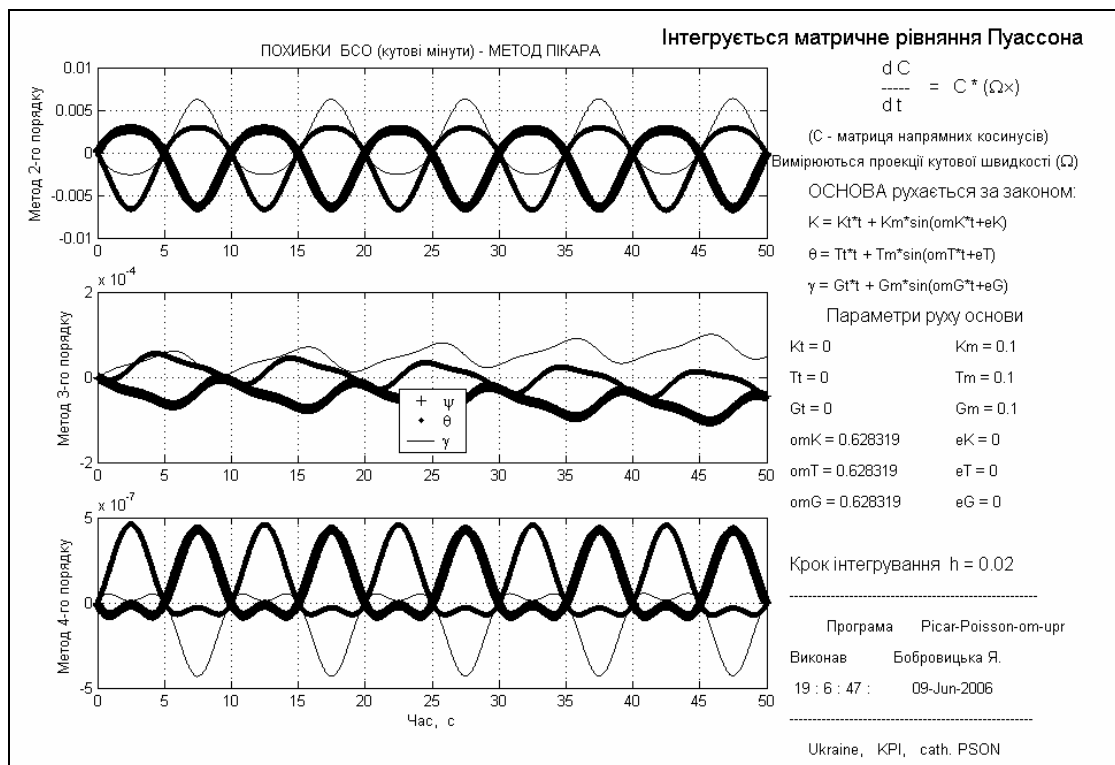


Рис. 5.2. Похибки алгоритмів методу Пікара для рівняння Пуассона ($h=0.02c$) при синхронних і синфазних коливаннях основи з трьох кутів (вимірюються проєкції кутової швидкості)

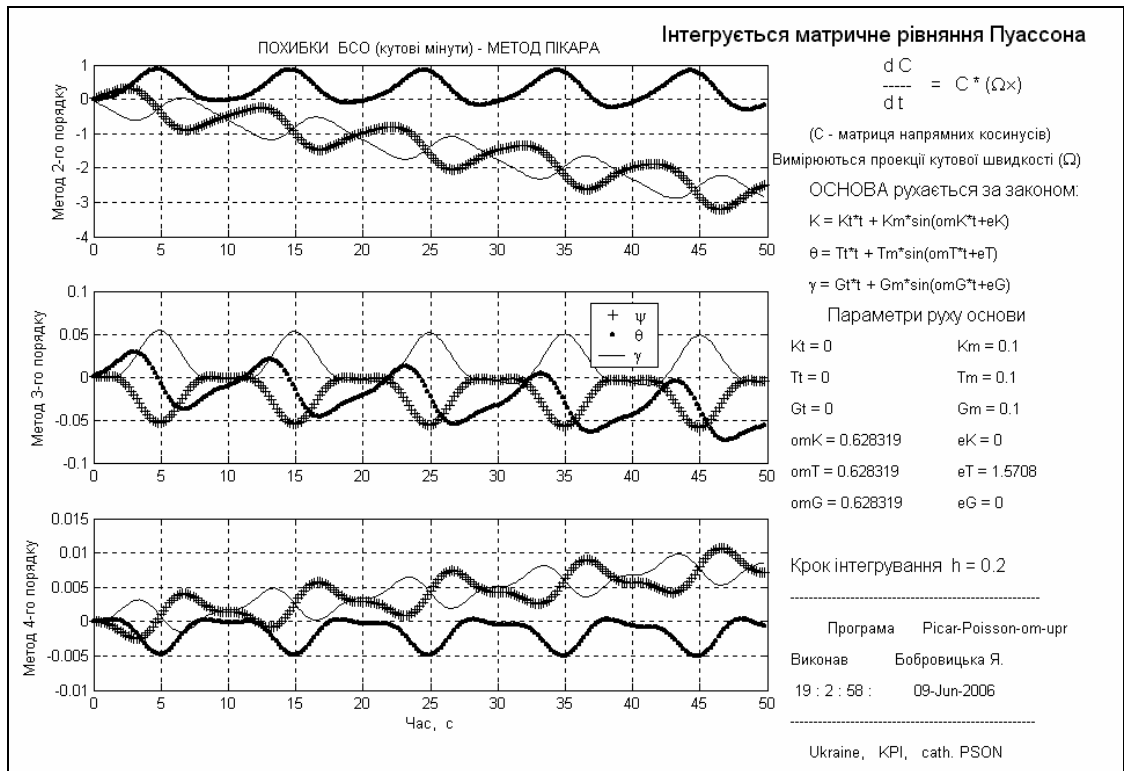


Рис. 5.3. Похибки алгоритмів методу Пікара для рівняння Пуассона ($h=0.2c$) при синхронних коливаннях основи з зсувом фаз 90° (вимірюються проєкції кутової швидкості)

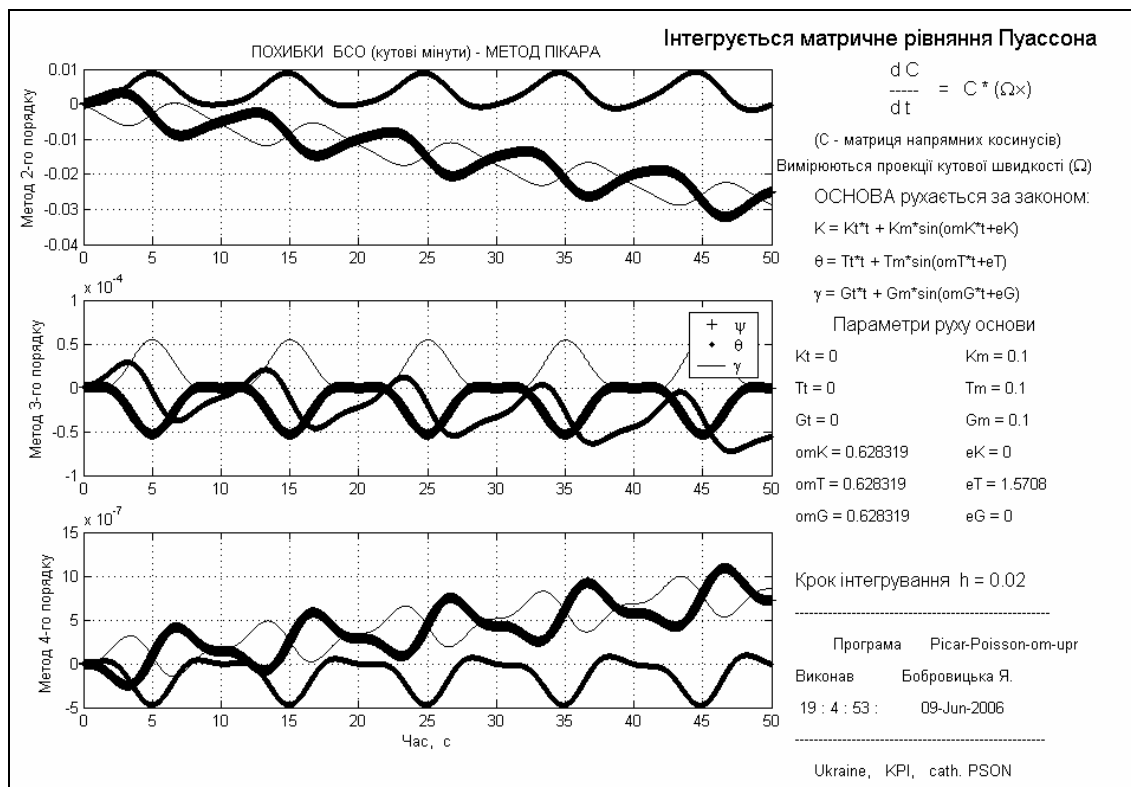


Рис. 5.4. Похибки алгоритмів методу Пікара для рівняння Пуассона ($h=0.02c$) при синхронних коливаннях основи з зсувом фаз 90° (вимірюються проєкції кутової швидкості)

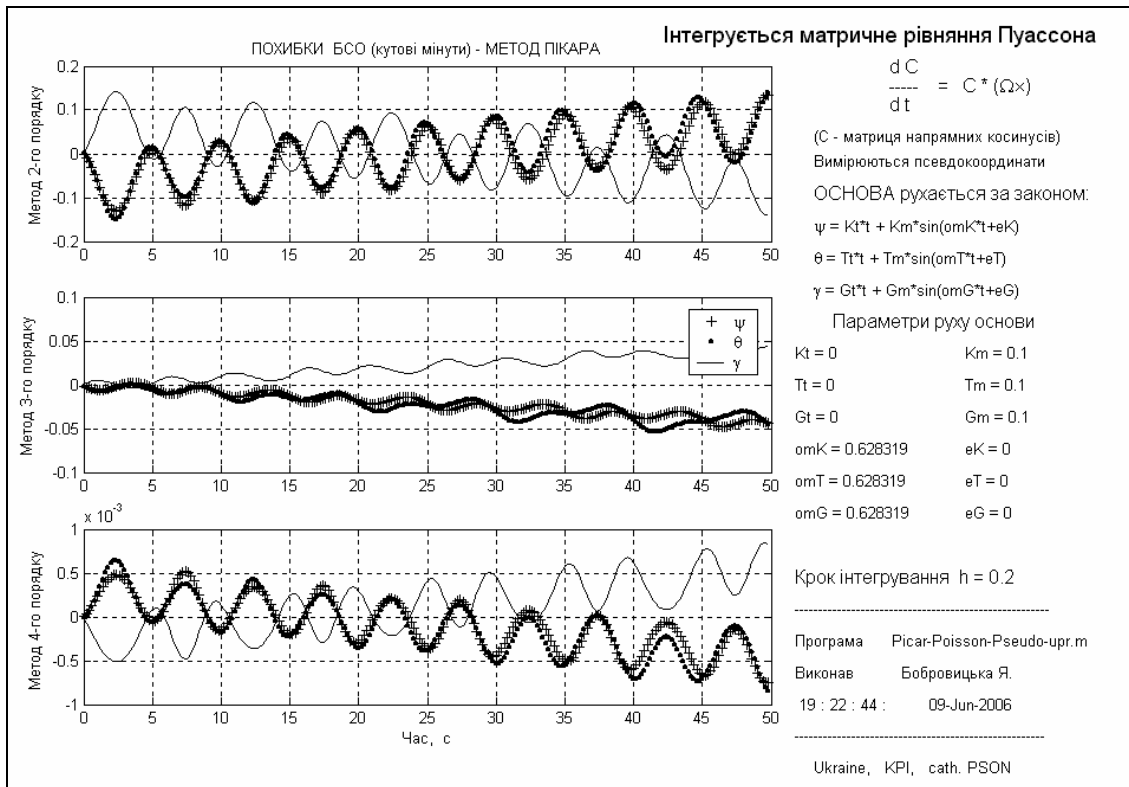


Рис. 5.5. Похибки алгоритмів методу Пікара для рівняння Пуассона (h=0.2c) при синхронних і синфазних коливаннях основи з трьох кутів (вимірюються прирости псевдокоординат)

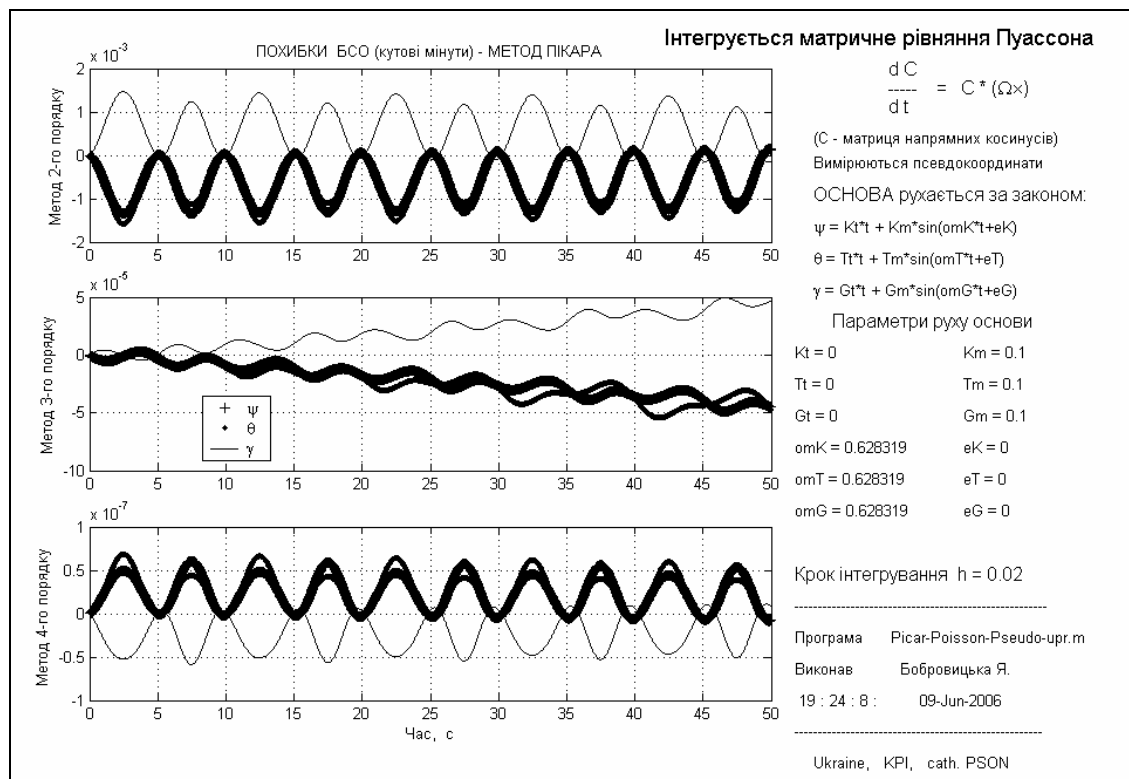


Рис. 5.6. Похибки алгоритмів методу Пікара для рівняння Пуассона (h=0.02c) при синхронних і синфазних коливаннях основи з трьох кутів (вимірюються прирости псевдокоординат)

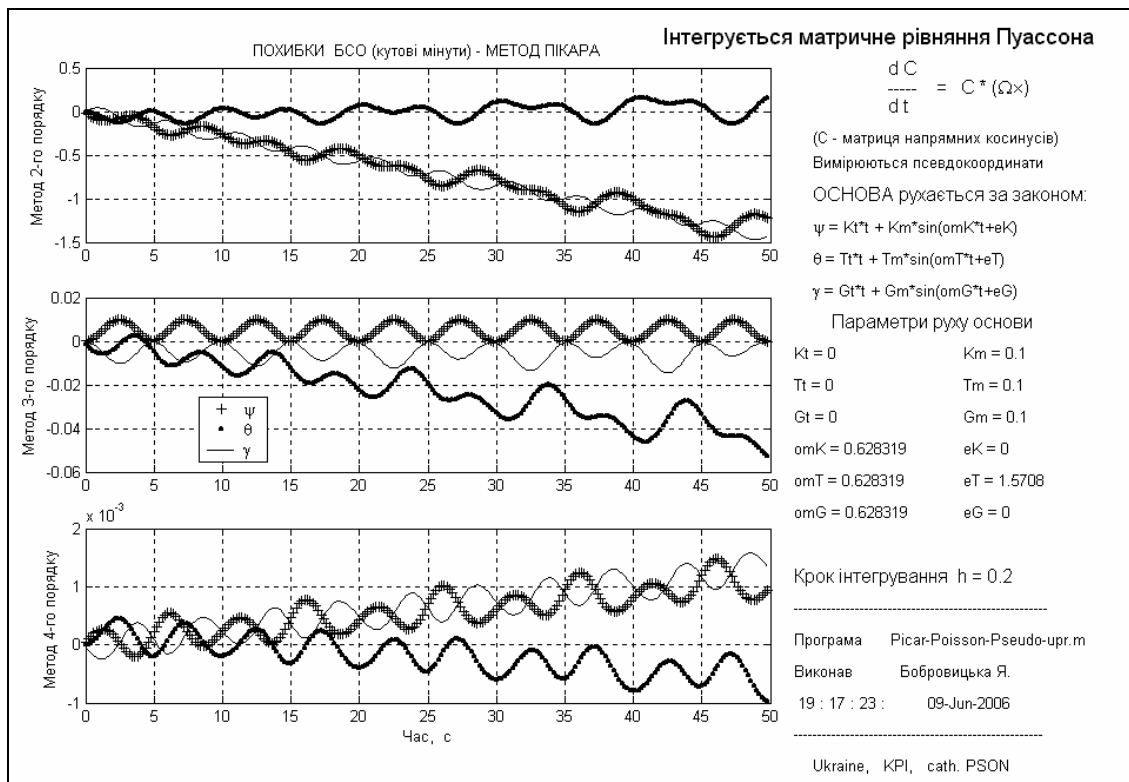


Рис. 5.7. Похибки алгоритмів методу Пікара для рівняння Пуассона (h=0.2с) при синхронних коливаннях основи з зсувом фаз 90° (вимірюються прирости псевдокоординат)

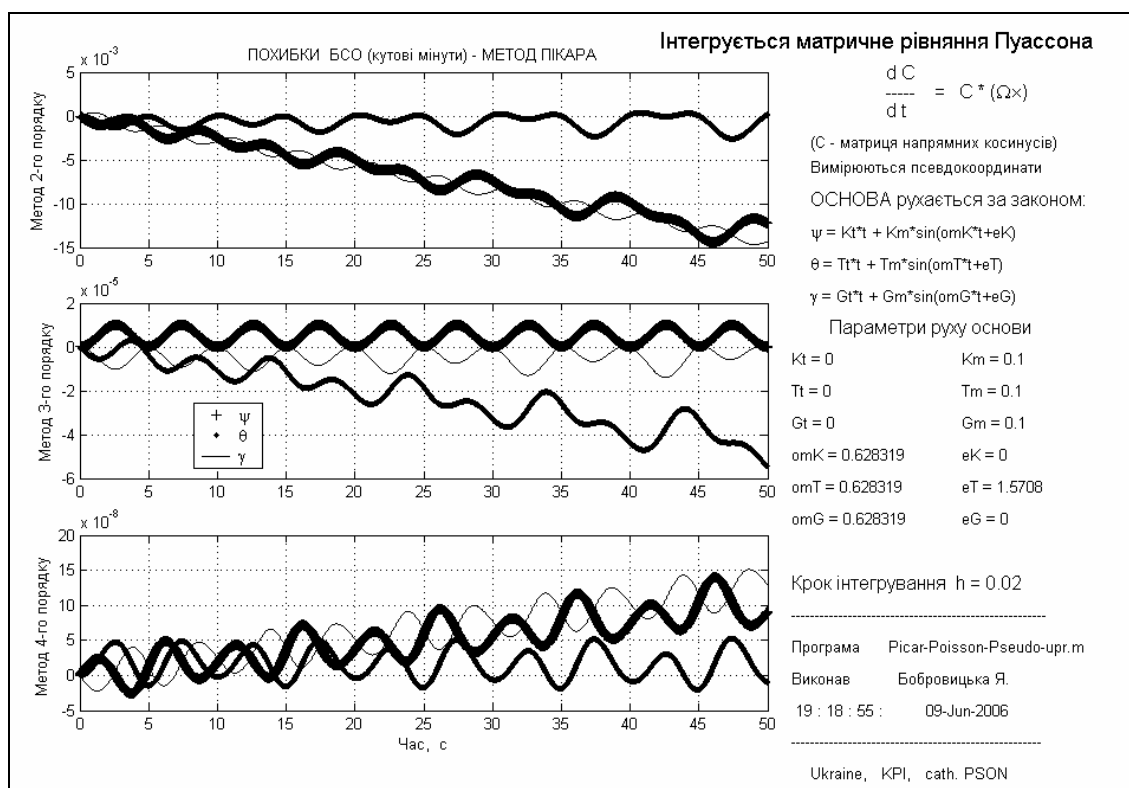


Рис. 5.8. Похибки алгоритмів методу Пікара для рівняння Пуассона (h=0.02с) при синхронних коливаннях основи з зсувом фаз 90° (вимірюються прирости псевдокоординат)

Деякі характеристики одержаних внаслідок моделювання похибок алгоритмів наведені у таблицях 5. 1 і 5.2.

Таблиця 5.1. Параметри похибок алгоритмів методу Пікара для рівняння Пуассона при синфазній хитавиці основи

Вид руху основи	Крок	Порядок методу	Вид параметра	Вимірювання проекцій кутової швидкості	Вимірювання приростів псевдокоординат
Синфазні і синхронні коливання основи з трьох кутів з періодом 10 с і амплітудою 0,1 радіан	0,2с	2	Амплітуда	$(5 \cdot 10^{-1})'$	$(5 \cdot 10^{-2})'$
			Дрейф з ψ	0	$(2,4 \cdot 10^{-2})'/c$
			Дрейф з ϑ	0	$(4 \cdot 10^{-3})'/c$
			Дрейф з γ	0	$(2,8 \cdot 10^{-2})'/c$
		3	Амплітуда	$(2,5 \cdot 10^{-2})'$	$(5 \cdot 10^{-3})'$
			Дрейф з ψ	$-(1 \cdot 10^{-3})'/c$	$(1,2 \cdot 10^{-3})'/c$
			Дрейф з ϑ	$-(1 \cdot 10^{-3})'/c$	$(1,2 \cdot 10^{-3})'/c$
			Дрейф з γ	$(1 \cdot 10^{-3})'/c$	$(1,2 \cdot 10^{-3})'/c$
		4	Амплітуда	$(2,5 \cdot 10^{-3})'$	$(2,5 \cdot 10^{-4})'$
			Дрейф з ψ	0	$(2 \cdot 10^{-5})'/c$
			Дрейф з ϑ	0	$-(2 \cdot 10^{-5})'/c$
			Дрейф з γ	0	$(2 \cdot 10^{-5})'/c$
	0,02с	2	Амплітуда	$(5 \cdot 10^{-3})'$	$(5 \cdot 10^{-4})'$
			Дрейф з ψ	0	$-(2,4 \cdot 10^{-4})'/c$
			Дрейф з ϑ	0	0
			Дрейф з γ	0	$-(2,8 \cdot 10^{-4})'/c$
		3	Амплітуда	$(2,5 \cdot 10^{-5})'$	$(5 \cdot 10^{-6})'$
			Дрейф з ψ	$-(1 \cdot 10^{-6})'/c$	$(1 \cdot 10^{-6})'/c$
			Дрейф з ϑ	$-(1 \cdot 10^{-6})'/c$	$(1 \cdot 10^{-6})'/c$
			Дрейф з γ	$(1 \cdot 10^{-6})'/c$	$(1 \cdot 10^{-6})'/c$
		4	Амплітуда	$(2,5 \cdot 10^{-7})'$	$(2,5 \cdot 10^{-8})'$
			Дрейф з ψ	0	$(2 \cdot 10^{-9})'/c$
			Дрейф з ϑ	0	0
			Дрейф з γ	0	$-(2,8 \cdot 10^{-4})'/c$

Таблиця 5.2. Параметри похибок алгоритмів методу Пікара для рівняння Пуассона при хитавиці основи з зсувом фаз 90°

Вид руху основи	Крок	Порядок методу	Вид параметра	Вимірювання проєкцій кутової швидкості	Вимірювання приростів псевдокоординат
Синфазні коливання основи з трьох кутів з періодом 10 с і амплітудою 0,1 радіан і з зсувом фаз 90°	0,2с	2	Амплітуда	$(5 \cdot 10^{-1})'$	$(5 \cdot 10^{-2})'$
			Дрейф з ψ	$-(5 \cdot 10^{-2})'/c$	$-(2,4 \cdot 10^{-2})'/c$
			Дрейф з ϑ	0	$(4 \cdot 10^{-3})'/c$
			Дрейф з γ	$-(6 \cdot 10^{-2})'/c$	$-(2,8 \cdot 10^{-2})'/c$
		3	Амплітуда	$(2,5 \cdot 10^{-2})'$	$(5 \cdot 10^{-3})'$
			Дрейф з ψ	0	0
			Дрейф з ϑ	$-(1 \cdot 10^{-3})'/c$	$-(1 \cdot 10^{-3})'/c$
			Дрейф з γ	0	0
		4	Амплітуда	$(2,5 \cdot 10^{-3})'$	$(2,5 \cdot 10^{-4})'$
			Дрейф з ψ	$(1,6 \cdot 10^{-4})'/c$	$(2 \cdot 10^{-5})'/c$
			Дрейф з ϑ	0	$-(2 \cdot 10^{-5})'/c$
			Дрейф з γ	$(1,6 \cdot 10^{-4})'/c$	$(2,6 \cdot 10^{-5})'/c$
	0,02с	2	Амплітуда	$(5 \cdot 10^{-3})'$	$(5 \cdot 10^{-3})'$
			Дрейф з ψ	$-(5 \cdot 10^{-4})'/c$	$-(2,5 \cdot 10^{-4})'$
			Дрейф з ϑ	0	0
			Дрейф з γ	$-(5 \cdot 10^{-4})'/c$	$-(2,5 \cdot 10^{-4})'$
		3	Амплітуда	$(2,5 \cdot 10^{-5})'$	$(2,5 \cdot 10^{-6})'$
			Дрейф з ψ	0	0
			Дрейф з ϑ	$(1 \cdot 10^{-6})'/c$	$-(1 \cdot 10^{-6})'/c$
			Дрейф з γ	0	0
		4	Амплітуда	$(2,5 \cdot 10^{-7})'$	$(2,5 \cdot 10^{-8})'$
			Дрейф з ψ	$(1,6 \cdot 10^{-8})'/c$	$(2 \cdot 10^{-9})'/c$
			Дрейф з ϑ	0	0
			Дрейф з γ	$(1,6 \cdot 10^{-8})'/c$	$(2,6 \cdot 10^{-9})'/c$

5.5. Висновки

1. Розроблені алгоритми на основі методу Пікара для рівняння Пуассона є стійкими і забезпечують достатню точність визначення параметрів орієнтації.

2. Амплітуди похибок і сталі складові дрейфу похибок методів відповідного порядку при вимірюванні проєкцій кутової швидкості практично не відрізняються від таких для методів прогноз-корекції.

3. Амплітуди похибок у випадку вимірювання приростів псевдокоординат менші за такі для випадку вимірювання кутової швидкості приблизно на порядок.

4. Застосування вимірювачів приростів псевдокоординат дещо змінює характер прояву дрейфів похибок.

6. Алгоритми метода Пікара для рівняння у кватерніонах

Розглянемо більш докладно алгоритми інтегрування рівнянь орієнтації у кватерніонах, побудовані на основі застосування метода послідовних наближень Пікара..

6.1. Побудова алгоритмів

Інтегруючи почленно у межах від t_0 до t кінематичне диференціальне рівняння у кватерніонах

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

де \circ позначає кватерніонне множення, а

$$\omega(t) = 0 + \boldsymbol{\omega} = 0 + \mathbf{i}\omega_x + \mathbf{j}\omega_y + \mathbf{k}\omega_z$$

є вектор-кватерніоном вимірюваних проекцій кутової швидкості, одержимо

$$\Lambda(t) = \Lambda(t_0) + \frac{1}{2} \int_{t_0}^t \Lambda(\xi) \circ \omega(\xi) d\xi. \quad (6.1)$$

Покладемо

$$\Lambda(t) = \Lambda(t_0) \circ N(t). \quad (6.2)$$

Підставимо (2) у (1):

$$\Lambda(t_0) \circ N(t) = \Lambda(t_0) + \frac{1}{2} \int_{t_0}^t \Lambda(t_0) \circ N(\xi) \circ \omega(\xi) d\xi,$$

звідки випливає

$$N(t) = 1 + \frac{1}{2} \int_{t_0}^t N(\xi) \circ \omega(\xi) d\xi, \quad (6.3)$$

а рівняння (1) набуває вигляду

$$\Lambda(t) = \Lambda(t_0) \circ N(t). \quad (6.4)$$

Розглянемо алгоритм відшукування значення кватерніону повороту $\Lambda_{n+1} = \Lambda(t_{n+1})$ у момент часу $t = t_{n+1} = t_n + h$ за заданим його значенням $\Lambda_n = \Lambda(t_n)$ у попередній момент часу $t = t_n$, де h - крок інтегрування.

Застосовуючи (4), дійдемо

$$\Lambda_{n+1} = \Lambda_n \circ N_{n+1}, \quad (6.5)$$

причому, у відповідності до (3), N_{n+1} є розв'язком рівняння

$$N(t) = 1 + \frac{1}{2} \int_{t_n}^t N(\xi) \circ \omega(\xi) d\xi \quad (6.6)$$

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

Рівняння (6), за методом Пікара, розв'язується послідовними наближеннями.

6.1.1. Побудова наближених розв'язків допоміжного рівняння

Перше наближення.

У нульовому наближенні приймемо $N_{(0)}(t) = 1$. Підставивши це у праву частину (6), одержимо розв'язок рівняння (6) у першому наближенні:

$$N_{(1)}(t) = 1 + \frac{1}{2} \int_{t_n}^t N_{(0)}(\xi) \circ \omega(\xi) d\xi = 1 + \frac{1}{2} \int_{t_n}^t \omega(\xi) d\xi \quad (6.7)$$

Друге наближення.

$$\begin{aligned} N_{(2)}(t) &= 1 + \frac{1}{2} \int_{t_n}^t N_{(1)}(\xi) \circ \omega(\xi) d\xi = 1 + \frac{1}{2} \int_{t_n}^t \left(1 + \frac{1}{2} \int_{t_n}^{\xi} \omega(\eta) d\eta \right) \circ \omega(\xi) d\xi = \\ &= 1 + \frac{1}{2} \int_{t_n}^t \omega(\xi) d\xi + \frac{1}{4} \int_{t_n}^t \left(\int_{t_n}^{\xi} \omega(\eta) d\eta \right) \circ \omega(\xi) d\xi \quad . \end{aligned} \quad (6.8)$$

Третє наближення.

$$\begin{aligned} N_{(3)}(t) &= 1 + \frac{1}{2} \int_{t_n}^t N_{(2)}(\xi) \circ \omega(\xi) d\xi = 1 + \frac{1}{2} \int_{t_n}^t \left(1 + \frac{1}{2} \int_{t_n}^{\xi} \omega(\eta) d\eta \right) \circ \omega(\xi) d\xi = \\ &= 1 + \frac{1}{2} \int_{t_n}^t \left[1 + \frac{1}{2} \int_{t_n}^{\xi} \omega(\eta) d\eta + \frac{1}{4} \int_{t_n}^{\xi} \left(\int_{t_n}^{\eta} \omega(\zeta) d\zeta \right) \circ \omega(\eta) d\eta \right] \circ \omega(\xi) d\xi = \\ &= 1 + \frac{1}{2} \int_{t_n}^t \omega(\xi) d\xi + \frac{1}{4} \int_{t_n}^t \left(\int_{t_n}^{\xi} \omega(\eta) d\eta \right) \circ \omega(\xi) d\xi + \\ &+ \frac{1}{8} \int_{t_n}^t \left[\int_{t_n}^{\xi} \left(\int_{t_n}^{\eta} \omega(\zeta) d\zeta \right) \circ \omega(\eta) d\eta \right] \circ \omega(\xi) d\xi \end{aligned} \quad (6.9)$$

Четверте наближення.

$$\begin{aligned} N_{(4)}(t) &= 1 + \frac{1}{2} \int_{t_n}^t N_{(3)}(\xi) \circ \omega(\xi) d\xi = \\ &= 1 + \frac{1}{2} \int_{t_n}^t \omega(\xi) d\xi + \frac{1}{4} \int_{t_n}^t \left(\int_{t_n}^{\xi} \omega(\eta) d\eta \right) \circ \omega(\xi) d\xi + \\ &+ \frac{1}{8} \int_{t_n}^t \left[\int_{t_n}^{\xi} \left(\int_{t_n}^{\eta} \omega(\zeta) d\zeta \right) \circ \omega(\eta) d\eta \right] \circ \omega(\xi) d\xi + \\ &+ \frac{1}{16} \int_{t_n}^t \left\{ \int_{t_n}^{\xi} \left[\int_{t_n}^{\eta} \left(\int_{t_n}^{\zeta} \omega(v) dv \right) \circ \omega(\zeta) d\zeta \right] \circ \omega(\eta) d\eta \right\} \circ \omega(\xi) d\xi \quad . \end{aligned} \quad (6.10)$$

6.1.2. Уточнення розв'язків для поліноміальної апроксимації кутової швидкості

Як і раніше, припустимо, що в діапазоні від t_n до t_{n+1} (від $\tau = 0$ до $\tau = h$) вектор $\boldsymbol{\omega} = \mathbf{i}\omega_x + \mathbf{j}\omega_y + \mathbf{k}\omega_z$ вимірюваних проекцій кутової швидкості об'єкту залежить від часу за законом

$$\boldsymbol{\omega} = \boldsymbol{\omega}_o + 2\boldsymbol{\varepsilon} \cdot \tau + 3\mathbf{a} \cdot \tau^2 + 4\mathbf{b} \cdot \tau^3, \quad (6.11)$$

де $\boldsymbol{\omega}_o$, $\boldsymbol{\varepsilon}$, \mathbf{a} і \mathbf{b} - постійні вектори, значення яких визначається шляхом поліноміального інтерполювання по вимірних значеннях кутової швидкості (чи інтеграла від неї) на кількох попередніх кроках інтегрування.

З врахуванням цього відшукаємо інтеграли, що входять у (10), залишаючи члени не вище за четвертий порядок мализни відносно малого кроку інтегрування.

$$\int_{t_n}^t \boldsymbol{\omega}(\xi) d\xi = \int_0^\tau (\boldsymbol{\omega}_o + 2\boldsymbol{\varepsilon} \cdot \xi + 3\mathbf{a} \cdot \xi^2 + 4\mathbf{b} \cdot \xi^3) d\xi = \boldsymbol{\omega}_o \cdot \tau + \boldsymbol{\varepsilon} \cdot \tau^2 + \mathbf{a} \cdot \tau^3 + \mathbf{b} \cdot \tau^4. \quad (6.12)$$

Далі, використовуючи знайдений вираз (12), відшукаємо наступні інтеграли:

$$\begin{aligned} & \int_{t_n}^t \left(\int_{t_n}^{\xi} \boldsymbol{\omega}(\eta) d\eta \right) \circ \boldsymbol{\omega}(\xi) d\xi = \\ & = \int_0^\tau (\boldsymbol{\omega}_o \cdot \xi + \boldsymbol{\varepsilon} \cdot \xi^2 + \mathbf{a} \cdot \xi^3 + \mathbf{b} \cdot \xi^4) \circ (\boldsymbol{\omega}_o + 2\boldsymbol{\varepsilon} \cdot \xi + 3\mathbf{a} \cdot \xi^2 + 4\mathbf{b} \cdot \xi^3) d\xi \approx \\ & \approx \frac{1}{2} (\boldsymbol{\omega}_o \circ \boldsymbol{\omega}_o) \cdot \tau^2 + \frac{1}{3} (2\boldsymbol{\omega}_o \circ \boldsymbol{\varepsilon} + \boldsymbol{\varepsilon} \circ \boldsymbol{\omega}_o) \cdot \tau^3 + \frac{1}{4} (3\boldsymbol{\omega}_o \circ \mathbf{a} + 2\boldsymbol{\varepsilon} \circ \boldsymbol{\varepsilon} + \mathbf{a} \circ \boldsymbol{\omega}_o) \tau^4 = \\ & = \frac{1}{2} (-\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_o) \cdot \tau^2 + \frac{1}{3} (-2\boldsymbol{\omega}_o \cdot \boldsymbol{\varepsilon} + 2\boldsymbol{\omega}_o \times \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon} \cdot \boldsymbol{\omega}_o + \boldsymbol{\varepsilon} \times \boldsymbol{\omega}_o) \cdot \tau^3 + \\ & + \frac{1}{4} (-3\boldsymbol{\omega}_o \cdot \mathbf{a} + 3\boldsymbol{\omega}_o \times \mathbf{a} - 2\boldsymbol{\varepsilon} \cdot \boldsymbol{\varepsilon} - \boldsymbol{\omega}_o \cdot \mathbf{a} + \mathbf{a} \times \boldsymbol{\omega}_o) \tau^4 = \\ & = -\frac{1}{2} |\boldsymbol{\omega}_o|^2 \tau^2 + \frac{1}{3} (-3\boldsymbol{\omega}_o \cdot \boldsymbol{\varepsilon} + \boldsymbol{\omega}_o \times \boldsymbol{\varepsilon}) \tau^3 + \frac{1}{2} (-2\boldsymbol{\omega}_o \cdot \mathbf{a} + \boldsymbol{\omega}_o \times \mathbf{a} - |\boldsymbol{\varepsilon}|^2) \tau^4. \quad (6.13) \end{aligned}$$

Тут враховано, що кватерніонний добуток двох векторів-кватерніонів дорівнює

$$\mathbf{a} \circ \mathbf{b} = (0 + \mathbf{a}) \circ (0 + \mathbf{b}) = -\mathbf{a} \cdot \mathbf{b} + \mathbf{a} \times \mathbf{b},$$

тобто різниці між векторним добутком векторів і їхнім скалярним добутком.

Наступний (потрійний) інтеграл матиме вигляд:

$$\begin{aligned} & \int_0^\tau \left[\int_0^\xi \left(\int_0^\eta \boldsymbol{\omega}(\zeta) d\zeta \right) \circ \boldsymbol{\omega}(\eta) d\eta \right] \circ \boldsymbol{\omega}(\xi) d\xi = \\ & = \int_0^\tau \left[-\frac{1}{2} |\boldsymbol{\omega}_o|^2 \xi^2 + \frac{1}{3} (-3\boldsymbol{\omega}_o \cdot \boldsymbol{\varepsilon} + \boldsymbol{\omega}_o \times \boldsymbol{\varepsilon}) \xi^3 + \frac{1}{2} (-2\boldsymbol{\omega}_o \cdot \mathbf{a} + \boldsymbol{\omega}_o \times \mathbf{a} - |\boldsymbol{\varepsilon}|^2) \xi^4 \right] \circ \\ & \circ (\boldsymbol{\omega}_o + 2\boldsymbol{\varepsilon} \cdot \xi + 3\mathbf{a} \cdot \xi^2 + 4\mathbf{b} \cdot \xi^3) d\xi \approx \end{aligned}$$

$$\begin{aligned}
&\approx -\frac{1}{6}|\omega_o|^2 \omega_o \tau^3 + \frac{1}{4} \left[-|\omega_o|^2 \boldsymbol{\varepsilon} + \frac{1}{3}(-3\omega_o \cdot \boldsymbol{\varepsilon} + \omega_o \times \boldsymbol{\varepsilon}) \circ \omega_o \right] \tau^4 \approx \\
&\approx -\frac{1}{6}|\omega_o|^2 \omega_o \tau^3 + \frac{1}{4} \left[-|\omega_o|^2 \boldsymbol{\varepsilon} + \frac{1}{3}(-3(\omega_o \cdot \boldsymbol{\varepsilon})\omega_o - (\omega_o \times \boldsymbol{\varepsilon})\omega_o + (\omega_o \times \boldsymbol{\varepsilon}) \times \omega_o) \right] \tau^4 \approx \\
&\approx -\frac{1}{6}|\omega_o|^2 \omega_o \tau^3 + \frac{1}{4} \left[-|\omega_o|^2 \boldsymbol{\varepsilon} + \frac{1}{3}(-4(\omega_o \cdot \boldsymbol{\varepsilon})\omega_o + |\omega_o|^2 \boldsymbol{\varepsilon}) \right] \tau^4 \approx \\
&\approx -\frac{1}{6}|\omega_o|^2 \omega_o \tau^3 - \frac{1}{6} \left[|\omega_o|^2 \boldsymbol{\varepsilon} + 2(\omega_o \cdot \boldsymbol{\varepsilon})\omega_o \right] \tau^4. \tag{6.14}
\end{aligned}$$

Нарешті четвертий інтеграл набуде такого виду:

$$\begin{aligned}
&\int_0^\tau \left\{ \int_0^\xi \left[\int_0^\eta \left(\int_0^\zeta \omega(v) dv \right) \circ \omega(\zeta) d\zeta \right] \circ \omega(\eta) d\eta \right\} \circ \omega(\xi) d\xi = \\
&= \int_{t_n}^t \left\{ -\frac{1}{6}|\omega_o|^2 \omega_o \xi^3 - \frac{1}{6} \left[|\omega_o|^2 \boldsymbol{\varepsilon} + 2(\omega_o \cdot \boldsymbol{\varepsilon})\omega_o \right] \xi^4 \right\} \circ \\
&\quad \circ (\omega_o + 2\boldsymbol{\varepsilon} \cdot \xi + 3\mathbf{a} \cdot \xi^2 + 4\mathbf{b} \cdot \xi^3) d\xi \approx -\frac{1}{24}|\omega_o|^2 (\omega_o \circ \omega_o) \tau^4 = \frac{1}{24}|\omega_o|^4 \tau^4. \tag{6.15}
\end{aligned}$$

У підсумку, з точністю до малих четвертого порядку величина N_{n+1} визначається формулою

$$\begin{aligned}
N_{n+1} &= 1 + \frac{1}{2}(\omega_o \cdot h + \boldsymbol{\varepsilon} \cdot h^2 + \mathbf{a} \cdot h^3 + \mathbf{b} \cdot h^4) + \\
&+ \frac{1}{4} \left[-\frac{1}{2}|\omega_o|^2 h^2 + \frac{1}{3}(-3\omega_o \cdot \boldsymbol{\varepsilon} + \omega_o \times \boldsymbol{\varepsilon})h^3 + \frac{1}{2}(-2\omega_o \cdot \mathbf{a} + \omega_o \times \mathbf{a} - |\boldsymbol{\varepsilon}|^2)h^4 \right] + \\
&+ \frac{1}{8} \left\{ -\frac{1}{6}|\omega_o|^2 \omega_o h^3 - \frac{1}{6} \left[|\omega_o|^2 \boldsymbol{\varepsilon} + 2(\omega_o \cdot \boldsymbol{\varepsilon})\omega_o \right] h^4 \right\} + \frac{1}{16} \left(\frac{1}{24}|\omega_o|^4 h^4 \right). \tag{6.16}
\end{aligned}$$

Залишаючи члени до другого порядку мализни включно, одержуємо формулу для методу Пікара другого порядку:

$$N_{n+1} = 1 + \frac{1}{2}\omega_o \cdot h + \frac{1}{2} \left(\boldsymbol{\varepsilon} - \frac{1}{4}|\omega_o|^2 \right) h^2. \tag{6.17}$$

Аналогічно одержимо формулу для методу Пікара третього порядку:

$$N_{n+1} = 1 + \frac{1}{2}\omega_o \cdot h + \frac{1}{2} \left(\boldsymbol{\varepsilon} - \frac{1}{4}|\omega_o|^2 \right) h^2 + \frac{1}{2} \left[\mathbf{a} + \frac{1}{6} \left(-3\omega_o \cdot \boldsymbol{\varepsilon} + \omega_o \times \boldsymbol{\varepsilon} - \frac{1}{4}|\omega_o|^2 \omega_o \right) \right] h^3. \tag{6.18}$$

Метод Пікара четвертого порядку визначається формулою (16)

$$\begin{aligned}
N_{n+1} &= 1 + \frac{1}{2}\omega_o \cdot h + \frac{1}{2} \left(\boldsymbol{\varepsilon} - \frac{1}{4}|\omega_o|^2 \right) h^2 + \frac{1}{2} \left[\mathbf{a} + \frac{1}{6} \left(-3\omega_o \cdot \boldsymbol{\varepsilon} + \omega_o \times \boldsymbol{\varepsilon} - \frac{1}{4}|\omega_o|^2 \omega_o \right) \right] h^3 + \\
&+ \frac{1}{2} \left[\mathbf{b} + \frac{1}{4}(-2\omega_o \cdot \mathbf{a} + \omega_o \times \mathbf{a} - |\boldsymbol{\varepsilon}|^2) - \frac{1}{24}(|\omega_o|^2 \boldsymbol{\varepsilon} + 2(\omega_o \cdot \boldsymbol{\varepsilon})\omega_o) + \frac{1}{192}|\omega_o|^4 \right] \cdot h^4. \tag{6.19}
\end{aligned}$$

6.1.3. Побудова алгоритмів

Формули алгоритмів виходять шляхом використання відшуканих у п. 5.1.4 і п. 5.1.5 виразів (5.21)...(5.45) коефіцієнтів інтерполяції у виразах (17)...(19) розв'язків проміжного рівняння.

Алгоритми другого порядку. У такий спосіб з попереднього виходить наступний алгоритм методу другого порядку:

$$\begin{aligned} N_{n+1} &= 1 + \frac{1}{2} \boldsymbol{\omega}_o \cdot h + \frac{1}{2} \left(\boldsymbol{\varepsilon} - \frac{1}{4} |\boldsymbol{\omega}_o|^2 \right) h^2 = 1 + \frac{1}{2} \boldsymbol{\omega}_o \cdot h + \frac{1}{2} \left(\frac{1}{2h} \Delta \boldsymbol{\omega}_1 - \frac{1}{4} |\boldsymbol{\omega}_o|^2 \right) h^2 = \\ &= 1 + \frac{1}{2} \boldsymbol{\omega}_o \cdot h + \frac{1}{4} (\boldsymbol{\omega}_1 - \boldsymbol{\omega}_o) h - \frac{1}{8} |\boldsymbol{\omega}_o|^2 h^2. \end{aligned} \quad (6.20)$$

Отже для випадку вимірювання кутової швидкості алгоритм матиме вид:

$$N_{n+1} = 1 + \frac{1}{4} (\boldsymbol{\omega}_o + \boldsymbol{\omega}_1) \cdot h - \frac{1}{8} |\boldsymbol{\omega}_o|^2 h^2. \quad (6.21)$$

Для випадку вимірювання приростів псевдокоординат вираз алгоритму визначиться у такий спосіб:

$$\begin{aligned} N_{n+1} &= 1 + \frac{1}{2} \boldsymbol{\omega}_o \cdot h + \frac{1}{2} \left(\boldsymbol{\varepsilon} - \frac{1}{4} |\boldsymbol{\omega}_o|^2 \right) h^2 = \\ &= 1 + \frac{1}{2} \cdot \frac{1}{h} (\nabla \boldsymbol{\Theta}_1 - \frac{1}{2} \nabla^2 \boldsymbol{\Theta}_1) \cdot h + \frac{1}{2} \left(\frac{1}{2h^2} \nabla^2 \boldsymbol{\Theta}_1 - \frac{1}{4} \left| \frac{1}{h} (\nabla \boldsymbol{\Theta}_1 - \frac{1}{2} \nabla^2 \boldsymbol{\Theta}_1) \right|^2 \right) h^2. \end{aligned}$$

Для спрощення виразів будемо використовувати позначення:

$$\nabla \boldsymbol{\Theta}_1 = \nabla, \quad \nabla^2 \boldsymbol{\Theta}_1 = \nabla^2, \quad \nabla^3 \boldsymbol{\Theta}_1 = \nabla^3, \quad \nabla^4 \boldsymbol{\Theta}_1 = \nabla^4.$$

Остаточно одержимо:

$$N_{n+1} = 1 + \frac{1}{2} \Delta \boldsymbol{\Theta}_1 - \frac{1}{8} (\Delta \boldsymbol{\Theta}_1 - \frac{1}{2} \Delta^2 \boldsymbol{\Theta}_1)^2 = 1 + \frac{1}{2} \nabla - \frac{1}{8} (\nabla - \frac{1}{2} \nabla^2)^2 \quad (6.21a)$$

Алгоритми третього порядку. Перейдемо до формулювання алгоритмів третього порядку на основі формули (18).

$$N_{n+1} = 1 + \frac{1}{2} \boldsymbol{\omega}_o \cdot h + \frac{1}{2} \left(\boldsymbol{\varepsilon} - \frac{1}{4} |\boldsymbol{\omega}_o|^2 \right) h^2 + \frac{1}{2} \left[\mathbf{a} + \frac{1}{6} \left(-3 \boldsymbol{\omega}_o \cdot \boldsymbol{\varepsilon} + \boldsymbol{\omega}_o \times \boldsymbol{\varepsilon} - \frac{1}{4} |\boldsymbol{\omega}_o|^2 \boldsymbol{\omega}_o \right) \right] h^3$$

Використовуючи вирази (5.26) і (5.27), одержимо алгоритм для випадку вимірювання кутової швидкості:

$$\begin{aligned} N_{n+1} &= 1 + \frac{1}{2} \left[\boldsymbol{\omega}_o + \frac{1}{4} (\nabla \boldsymbol{\omega}_1 + \nabla \boldsymbol{\omega}_o) + \frac{1}{6} \nabla^2 \boldsymbol{\omega}_1 \right] h - \frac{1}{8} |\boldsymbol{\omega}_o|^2 h^2 + \\ &+ \frac{1}{48} \left(-3 \boldsymbol{\omega}_o \cdot (\nabla \boldsymbol{\omega}_1 + \nabla \boldsymbol{\omega}_o) + \boldsymbol{\omega}_o \times (\nabla \boldsymbol{\omega}_1 + \nabla \boldsymbol{\omega}_o) - |\boldsymbol{\omega}_o|^2 \boldsymbol{\omega}_o \right) h^2. \end{aligned} \quad (6.22)$$

Якщо ж скористатися значеннями (5.40) і (5.41), одержимо алгоритм для випадку вимірювання приростів псевдокоординат:

$$N_{n+1} = 1 + \frac{1}{2} \nabla - \frac{1}{8} \left| \nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3 \right|^2 + \frac{1}{24} \left[-3 \left(\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3 \right) \cdot \nabla^2 + \right.$$

$$\begin{aligned}
& + \left(\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3 \right) \times \nabla^2 - \frac{1}{2} \left(\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3 \right)^2 \cdot \left(\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3 \right)] \\
& + \frac{1}{24} \left(\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3 - \frac{1}{12} \nabla^4 \right) \times \left(\nabla^2 - \frac{1}{12} \nabla^4 \right) - \\
& - \frac{1}{48} \left(\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3 - \frac{1}{12} \nabla^4 \right)^2 \cdot \left(\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3 - \frac{1}{12} \nabla^4 \right) + \\
& + \frac{1}{48} \left(\nabla^4 - \frac{1}{24} \left(\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3 - \frac{1}{12} \nabla^4 \right) \cdot \left(\nabla^3 + \frac{1}{2} \nabla^4 \right) \right) + \\
& + \frac{1}{48} \nabla^4 - \frac{1}{24} \left(\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3 - \frac{1}{12} \nabla^4 \right) \times \left(\nabla^3 + \frac{1}{2} \nabla^4 \right) - \\
& - \frac{1}{32} \left(\nabla^2 - \frac{1}{12} \nabla^4 \right)^2 - \frac{1}{96} \left(\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3 - \frac{1}{12} \nabla^4 \right)^2 \cdot \left(\nabla^2 - \frac{1}{12} \nabla^4 \right) - \\
& - \frac{1}{48} \left(\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3 - \frac{1}{12} \nabla^4 \right) \cdot \left(\nabla^2 - \frac{1}{12} \nabla^4 \right) \cdot \left(\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3 - \frac{1}{12} \nabla^4 \right) + \\
& + \frac{1}{384} \left(\nabla - \frac{1}{2} \nabla^2 - \frac{1}{6} \nabla^3 - \frac{1}{12} \nabla^4 \right)^4.
\end{aligned}$$

Залишаючи члени не вище відповідно до порядку алгоритму, остаточно одержуємо :

$$\mathbf{N}_{n+1} = \left(1 - \frac{1}{8} |\nabla|^2 \right) + \left(\frac{1}{2} \nabla - \frac{1}{48} |\nabla|^2 \cdot \nabla + \frac{1}{24} \nabla \times \nabla^2 \right). \quad (6.22a)$$

Алгоритми четвертого порядку. Використовуючи значення коефіцієнтів (5.28)...(5.30), (5.44) і (5.45) та підставляючи їх у вираз (19), можемо одержати алгоритми для випадків вимірювання кутової швидкості і приростів псевдокоординат.

Залишаючи члени не вище відповідно до порядку алгоритму, для випадку вимірювання приростів псевдокоординат остаточно одержуємо :

$$\mathbf{N}_{n+1} = \left(1 - \frac{1}{8} |\nabla|^2 + \frac{1}{384} |\nabla|^4 \right) + \left(\frac{1}{2} \nabla_1 - \frac{1}{48} |\nabla|^2 \cdot \nabla + \frac{1}{24} \nabla \times \nabla^2 + \frac{1}{48} \nabla \times \nabla^3 \right). \quad (6.49a)$$

6.2. Засоби моделювання алгоритмів

З метою моделювання похибок розроблених алгоритмів потрібно створити процедури, які б виконували ті обчислювальні дії, що закладені у ці алгоритми. Приклад таких програм, написаних мовою системи Matlab, наведені нижче.

Програми *Picar2Kv.m*, *Picar3Kv.m*, *Picar4Kv.m* реалізують алгоритми, розраховані на вимірювання проєкцій кутової швидкості основи, відповідно другого, третього і четвертого порядку.

```

function Nout = Picar2Kv(h,t,omo,omo_1)
% Обчислення кватерніона-множника для кватерніона повороту основи
% через крок інтегрування
% для вимірювання вектора кутової швидкості

```

```

domo=omo_1-omo; eps=domo/(2*h); omo2=omo*omo';
n0=1-omo2/8*h^2; n=omo/2*h+eps/2*h^2; N=[n0,n];
Nout=quatern(N);

```

```

function Nout = Picar3Kv(h,t,omo1,omo,omo_1)
% Обчислення кватерніона-множника для кватерніона повороту основи
% через крок інтегрування
% для вимірювання вектора кутової швидкості
domo_1=omo_1-omo; domo=omo-omo1; domo2=domo_1-domo;
a=domo2/(6*h^2); eps=(domo_1+domo)/(4*h); omo2=omo*omo';
omeps=cross(omo,eps); n0=1-omo2/8*h^2-omo*eps'/4*h^3;
n=omo*h/2+eps*h^2/2+(a+omeps/6-omo2*omo/24)/2*h^3; N=[n0,n];
Nout=quatern(N);

```

```

function Nout = Picar4Kv(h,t,omo2,omo1,omo,omo_1)
% Обчислення кватерніона-множника для кватерніона повороту основи
% через крок інтегрування
% для вимірювання вектора кутової швидкості
domo_1=omo_1-omo; domo=omo-omo1; domo1=omo1-omo2;
domo2_1=domo_1-domo; domo2=domo-domo1; domo3_1=domo2_1-domo2;
b=domo3_1/(24*h^3); a=domo2_1/(6*h^2);
eps=(domo_1-domo2_1/2-domo3_1/6)/(2*h); omo2=omo*omo';
eps2=eps*eps'; omeps=cross(omo,eps);
oma=cross(omo,a);
n0=1-omo2*h^2/8-omo*eps'/4*h^3+(-omo*a'/4-eps2/8+omo2^2/384)*h^4;
n=omo*h/2+eps*h^2/2+(a+omeps/6-omo2*omo/24)/2*h^3+...
(b+oma/4-omo2*eps/24-(omo*eps')*omo/12)/2*h^4; N=[n0,n];
Nout=quatern(N);

```

Вхідними величинами цих процедур є:

- 1) величина h кроку інтегрування (вимірювання);
- 2) значення t поточного моменту часу;
- 3) значення omo вектора проєкцій кутової швидкості у попередні моменти часу і у момент часу, для якого розраховується кватерніон повороту.

Вихідною величиною є значення $Nout$ кватерніона N множника, який дозволяє у подальшому визначити значення кватерніона повороту шляхом його кватерніонного множення на значення кватерніона повороту на попередньому кроці інтегрування.

Робота наведених процедур спирається на операції з кватерніонами, які реалізовані у виді введення спеціального класу кватерніонів і програм методів оперування з представниками цього класу. Усі операції з об'єктами класу *qatern* зосереджені у каталозі *@qatern*. Процедури цього класу наведені у Додатку.

Наступні три процедури *PSK2m.m*, *PSK3.m* і *PSK4m.m* реалізують алгоритми, які використовують значення pk приростів псевдокоординат для обчислення кватерніона повороту.

```

function Nout= PSK2m(h,t,pk,pk_1)
% Обчислення кватерніона-множника кватерніону повороту на наступному кроці
% для випадку вимірювання приростів псевдокоординат
dT_1=pk_1; dT=pk;
n1=dT_1/2; n2=dT_1+dT; n3=n2*n2'/32; n0=1-n3; Nk=[n0,n1];
Nout=quatern(Nk);

```

```

function Nout= PSK3(h,t,pk,pk_1)
% Обчислення кватерніона-множника кватерніону повороту на наступному кроці
% для випадку вимірювання приростів псевдокоординат
dT1_1=pk_1; dT1=pk; dT2_1=dT1_1-dT1;
n1=dT1_1/2; kvdT1_1=(dT1_1*dT1_1); n2=kvdT1_1/24; n3=kvdT1_1/8;
n4=cross(dT1_1,dT2_1)/24; n0=1-n3; n=n1*(1-n2)+n4; Nk=[n0,n];
Nout=quatern(Nk);

function Nout= PSK4m(h,t,pk2,pk1,pk,pk_1)
% Обчислення кватерніона-множника кватерніону повороту на наступному кроці
% для випадку вимірювання приростів псевдокоординат
dT1_1=pk_1; dT1=pk; dT11=pk1; dT12=pk2;
dT2_1=dT1_1-dT1; dT2=dT1-dT11; dT21=dT11-dT12;
dT3_1=dT2_1-dT2; dT3=dT2-dT21;
n1=dT1_1/2; n2=dT1_1*dT1_1; n3=n2*n2;
n4=cross(dT1_1,dT2_1)/24; n5=cross(dT1_1,dT3_1)/48; n6=n2*dT1_1/48;
n0=1-n2/8+n3/384; n=n1-n6+n4+n5; Nk=[n0,n];
Nout=quatern(Nk);

```

6.3. Опис програмних засобів моделювання алгоритмів

Щоб вивчати похибки розроблених алгоритмів, потрібно розробити комплекс процедур, аналогічних тим, що були докладно подані у п. 4 і п. 5.

Програма *Picar_Quatern_om_upr.m* забезпечує запуск до виконання й взаємодію комплексу програмних засобів, які здійснюють чисельне інтегрування кінематичного рівняння орієнтації у кватерніонах методами Пікара різних порядків за умови одержання вимірювальної інформації у вигляді проекцій кутової швидкості основи і обчислюють похибки алгоритму такого чисельного інтегрування.

Використовуються наступні процедури й функції:

- 1) *idbso2UgSk.m* – процедура змінювання значень вхідних даних для програми *Picar_Quatern_om_upr*;
- 2) *UgDvObm.m* – функція обчислення параметрів поточного кутового руху основи (кутів, кутових швидкостей, кутових прискорень);
- 3) *Picar2Kv.m* – процедура чисельного інтегрування методом Пікара 2-го порядку (по двох попередніх точках) кінематичного рівняння у кватерніонах;
- 4) *Picar3Kv.m* – процедура чисельного інтегрування методом Пікара 3-го порядку (по трьох попередніх точках) кінематичного рівняння у кватерніонах;
- 5) *Picar4Kv.m* – процедура чисельного інтегрування методом Пікара 4-го порядку (по чотирьох попередніх точках) кінематичного рівняння у кватерніонах;
- 6) *grafik91.m* – процедура побудови графіків 9 величин як функцій від однієї величини;
- 7) *DnGrBSOUgSk.m* – процедура виведення на поле графіків розрахункових даних у текстовій формі.

Текст головної програми наведений нижче.

```

% Picar_Quatern_om_upr
% Програма Picar_Quatern_om_upr визначення похибок алгоритмів чисельн. інтегр.

```

```

% 1) кінематичних рівнянь орієнтації у кватерніонах
% 2) методами Пікара другого, третього і четвертого порядку
% 3) шляхом використання вимірювальної інформації про
%   проекції кутової швидкості основи
% Використовуються наступні процедури й функції:
%   idbso2UgSk.m - процедура змінювання значень вхідних даних для
%                 програми BSOintUgSk_upr
%   UgDvOb.m - функція обчислення параметрів поточного кутового руху
%                 основи (кутів, кутових швидкостей, кутових прискорень)
%   Picar2Kv.m - процедура чисельного інтегрування методом Пікара
%                 2-го порядку (по двох попередніх точках)
%   Picar3Kv.m - процедура чисельного інтегрування методом Пікара
%                 3-го порядку (по трьох попередніх точках)
%   Picar4Kv.m - процедура чисельного інтегрування методом Пікара
%                 4-го порядку (по чотирьох попередніх точках)
%   grafik91.m - процедура побудови графіків 9 величин як функцій від однієї величини
%                 на трьох підграфіках один під одним
%   DnGrBSOUgSk.m - процедура виведення на поле графіків розрахункових
%                 даних у текстовій формі

```

```

% Лазарев Ю. Ф., Бобровицька Я. 27-03-2006 р.
echo off, clc, clear all
global sprogram sname
global D2
global D5
sprogram='Picar-Quatern-om-upr.m'; clc
disp(' Ви увійшли в програму Picar_Quatern_om_upr,')
disp(' яка інтегрує кінематичні рівняння у кватерніонах')
disp(' по виміряних проекціях кутової швидкості')
disp(' і обчислює похибки цього чисельного інтегрування ')
disp(' ')
k0=menu('Що робити?','Продовжити роботу','Закінчити роботу');
if k0==1 % Вхід у програму
    sname='Бобровицька Я.';
    % Введення "вшитих" значень первісних параметрів руху основи
    ps0=0; te0=0; fi0=0;
    psm=0.1; tem=psm; fim=psm;
    omps=2*pi/10; omte=omps; omfi=omps;
    eps=0; ete=0; efi=0;
    % параметрів інтегрування;
    h=0.2; 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 для ПОВТОРУ ЕКСПЕРИМЕНТІВ
    while k0==1
        idbso2UgSk; % БЛОК ЗМІНЮВАННЯ ДАНИХ;
        % Визначення мінімального Ti и максимального Ta періодів
        qwe=[D2(7) D2(8) D2(9)]; % Формування масиву частот кутових коливань
        Ti = 2*pi/max(qwe); Ta=Ti;
        for i=1:3,
            if D2(6+i)>0
                T1=2*pi/D2(6+i);
                if T1>Ta Ta=T1; end
                if T1<Ti Ti=T1; end
            end
        end
        % Визначення величини кроку інтегрування h
        h = D4(1);
        % Визначення моменту Tn0 часу початку формування
        % вектора масивів даних вимірювань
        Tn0 = Ta*D4(2);
    end
end

```

```

% Визначення кінцевого моменту часу tfinal інтегрування
tfinal= Ta*(D4(2)+D4(3));
D5=[h Tn0 tfinal];
% БЛОК ІНТЕГРУВАННЯ
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t=0; % Завдання початкового моменту часу
% Завдання вектора початкових умов у попередніх точках
x=Ugdvobm(t-2*h); % Параметри руху ОСНОВИ 2 точки потому
ys2=[-x(1) x(2) x(3)]; om2=[x(7) x(8) x(9)];
x=Ugdvobm(t-h); % Параметри руху ОСНОВИ 1 точки потому
ys1=[-x(1) x(2) x(3)]; om1=[x(7) x(8) x(9)];
x=Ugdvobm(t); % Параметри руху ОСНОВИ у поточній точці
ys=[-x(1) x(2) x(3)]; om=[x(7) x(8) x(9)];
Ls=Ug2quat(-x(1),2,x(2),3,x(3),1);
x=Ugdvobm(t+h); % Параметри руху ОСНОВИ у наступній точці
om_1=[x(7) x(8) x(9)];
kv=[2 3 1];
L2s=Ls; % для 2-ого порядку
L3s=Ls; % для 3-ого порядку
L4s=Ls; % для 4-ого порядку
imas=1; % Лічильник елементів вихідних масивів
% Цикл інтегрування
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('Почекайте (йде обчислення)')
while (imas-1)*h<tfinal
t=(imas-1)*h;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Метод другого порядку
N2= Picar2Kv(h,t,om,om_1);
L2=L2s*N2; L2s=L2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Метод третього порядку
N3= Picar3Kv(h,t,om1,om,om_1);
L3=L3s*N3; L3s=L3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Метод четвертого порядку
N4= Picar4Kv(h,t,om2,om1,om,om_1);
L4=L4s*N4; L4s=L4;
om2=om1; om1=om; om=om_1;
x=UgDvObm(t+2*h); % Поточне положення основи
om_1=[x(7) x(8) x(9)]; x=UgDvObm(t+h);
% для метода другого порядку
Yg2=Quat2Ug(L2,kv);
dpsi2(imas)=Yg2(1)+x(1); % Масиви похибок
dteta2(imas)=Yg2(2)-x(2); % оцінювання
dfi2(imas)=Yg2(3)-x(3); % кутів повороту основи
% для метода третього порядку
Yg3=Quat2Ug(L3,kv);
dpsi3(imas)=Yg3(1)+x(1); % Масиви похибок
dteta3(imas)=Yg3(2)-x(2); % оцінювання
dfi3(imas)=Yg3(3)-x(3); % кутів повороту основи
% для Адамса-Башфорта
Yg4=Quat2Ug(L4,kv);
dpsi4(imas)=Yg4(1)+x(1); % Масиви похибок
dteta4(imas)=Yg4(2)-x(2); % оцінювання
dfi4(imas)=Yg4(3)-x(3); % кутів повороту основи
tt(imas)=t; % Массив моментів времени
imas=imas+1
end
% Побудова Графіків
stit1='ПОХИБКИ БСО (кутові мінути) - метод ПІКАРА';
sylab1='Метод 2-го порядку'; sxlab1=' '; stit2=' ';
sylab2='Метод 3-го порядку'; sxlab2=' '; stit3=' ';
sylab3='Метод 4-го порядку'; sxlab3='Час, с';
grafik91(tt,dpsi2*180*60/pi,dteta2*180*60/pi,dfi2*180*60/pi,stit1,sylab1,sxlab1,...

```



```

dpsi3*180*60/pi,dteta3*180*60/pi,dfi3*180*60/pi,stit2,sylab2,sxlab2,...
dpsi4*180*60/pi,dteta4*180*60/pi,dfi4*180*60/pi,stit3,sylab3,sxlab3);
DnGrBSOUgSk; % 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')

```

Наступна сукупність програм ініціюється і керується головною програмою *Picar_Quatern_Pseudo_upr*. Вона забезпечує обчислення похибок чисельного інтегрування кватерніонного рівняння орієнтації за допомогою алгоритмів методу Пікара. При цьому припускається, що вимірювальна інформація надається у вигляді значень приростів псевдокоординат (інтегралів від відповідних проекцій кутової швидкості основи) за крок інтегрування.

Використовуються наступні процедури й функції:

1) *idbso2Pseudo.m* – процедура змінювання значень вхідних даних для програми *Picar_Quatern_Pseudo_upr*;

2) *UgDvObPseudo.m* – функція обчислення параметрів поточного кутового руху основи (кутів, кутових швидкостей, псевдокоординат);

3) *Kvaz2.m* – процедура визначення значень псевдокоординат (чисельного інтегрування) методом Пікара 2-го порядку (по двох попередніх точках) кватерніонного рівняння орієнтації;

4) *Kvaz3.m* – процедура визначення значень псевдокоординат (чисельного інтегрування) методом Пікара 3-го порядку (по трьох попередніх точках) кватерніонного рівняння орієнтації;

5) *Kvaz4.m* – процедура визначення значень псевдокоординат (чисельного інтегрування) методом Пікара 4-го порядку (по чотирьох попередніх точках) кватерніонного рівняння орієнтації;

6) *grafik91.m* - процедура побудови графіків 9 величин як функцій від однієї величини;

7) *dngrbsoPseudo.m* - процедура виведення на поле графіків розрахункових даних у текстовій формі.

Далі наводиться текст головної програми.

```

% Picar_Quatern_Pseudo_upr
% Програма Picar_Quatern_Pseudo_upr визначення похибок алгоритму чисельн. інтегр.
% 1) кінематичних рівнянь орієнтації у кватерніонах
% 2) багатокроковими методами другого, третього і четвертого порядку
% 3) шляхом використання вимірювальної інформації про прирости
% псевдокоординат (інтегралів від проекцій кутової швидкості основи)
% Використовуються наступні процедури й функції:
% idbso2Pseudo.m - процедура змінювання значень вхідних даних для
% програми Pseudo_upr2
% UgDvObPseudo.m - функція обчислення параметрів поточного кутового руху
% основи (кутів, кутових швидкостей, псевдокоординат )
% Kvaz2.m - процедура чисельного інтегрування методом
% 2-го порядку (по двох попередніх точках)
% Kvaz3.m - процедура чисельного інтегрування методом
% 3-го порядку (по трьох попередніх точках)
% Kvaz4.m - процедура чисельного інтегрування методом

```

```

%          4-го порядку (по чотирьох попередніх точках)
% grafik91.m - процедура побудови графіків 9 величин як функцій від однієї величини
%          на трьох підграфіках один під одним
% dngrbsoPseudo.m - процедура виведення на поле графіків розрахункових
%          даних у текстовій формі

% Лазарев Ю. Ф., Бобровицька Я. 27-03-2006 р.
echo off, clc, clear all
global sprogram sname
global D2
global D5
sprogram='Picar-Quatern-Pseudo-upr.m'; clc
disp(' Ви увійшли в програму Picar_Quatern_Pseudo_upr,')
disp('яка інтегрує кінематичні рівняння у кватерніонах')
disp('по вимірних псевдокоординатах і обчислює похибки цього чисельного інтегру-
вання'), disp(' ')
k0=menu('Що робити?','Продовжити роботу','Закінчити роботу');
if k0==1 % Вхід у програму
    sname='Бобровицька Я.';
    % Введення "вшитих" значень первісних параметрів руху основи
    ps0=0;      te0=0;      fi0=0;
    psm=0.1;    tem=psm;    fim=psm;
    omps=2*pi/10; omte=omps; omfi=omps;
    eps=0;      ete=0;      efi=0;
    % параметрів інтегрування;
    h=0.2;      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 для ПОВТОРУ ЕКСПЕРИМЕНТІВ
    while k0==1
        idbso2Pseudo; % БЛОК ЗМІНЮВАННЯ ДАНИХ;
        % Визначення мінімального Ti і максимального Ta періодів
        qwe=[D2(7) D2(8) D2(9)]; % Формування масиву частот кутових коливань
        Ti = 2*pi/max(qwe); Ta=Ti;
        for i=1:3,
            if D2(6+i)>0
                T1=2*pi/D2(6+i);
                if T1>Ta Ta=T1; end
                if T1<Ti Ti=T1; end
            end
        end
        % Визначення величини кроку інтегрування h
        h = D4(1);
        % Визначення моменту Tn0 часу початку формування
        % вектора масивів даних вимірювань
        Tn0 = Ta*D4(2);
        % Визначення кінцевого моменту часу tfinal інтегрування
        tfinal= Ta*(D4(2)+D4(3));
        D5=[h Tn0 tfinal];
        % БЛОК ІНТЕГРУВАННЯ
        %%%%%%%%%%%%%%%
        t=0; % Завдання початкового моменту часу
        % Завдання вектора початкових умов у попередніх точках
        z=UgDvObPseudo(t-2*h,h); % Пар-ри руху ОСНОВИ у момент часу на два кроки назад
        rk2=[z(4) z(5) z(6)]; % Прирости псевдокоординат у момент часу на два кроки назад
        z=UgDvObPseudo(t-h,h); % Параметри руху ОСНОВИ у попередній точці
        rk1=[z(4) z(5) z(6)]; % Прирости псевдокоординат у попередній момент часу
        z=UgDvObPseudo(t,h); % Параметри руху ОСНОВИ у поточній точці
        rk=[z(4) z(5) z(6)]; % Прирости псевдокоординат у поточний момент часу
        Ls=Ug2quat(-z(1),2,z(2),3,z(3),1); % Кути у поточний момент часу
        z=UgDvObPseudo(t+h,h); % Параметри руху ОСНОВИ у наступній точці
    end
end

```

```

pk_1=[z(4) z(5) z(6)]; % Прирости псевдокоординат у наступний момент часу
Ls2=Ls; Ls3=Ls; Ls4=Ls;
kv=[2 3 1];
imas=1; % Лічильник елементів вихідних масивів
% Цикл інтегрування
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('Почекайте (йде обчислення)')
while (imas-1)*h<tfinal
    t=(imas-1)*h;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Метод першого порядку
    N2= PSK2m(h,t,pk,pk_1);
    L2 =Ls2*N2; Ls2 = L2;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Метод третього порядку
    N3= PSK3(h,t,pk,pk_1);
    L3 =Ls3*N3; Ls3 = L3;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Метод четвертого порядку
    N4= PSK4m(h,t,pk2,pk1,pk,pk_1);
    L4 =Ls4*N4; Ls4 = L4;
    pk2=pk1; pk1=pk; pk=pk_1;
    z=UgDvObPseudo(t+2*h,h); % Поточне положення основи
    pk_1=[z(4) z(5) z(6)]; z=UgDvObPseudo(t+h,h);
    % Ug=[z(1) z(2) z(3)]
    % для метода другого порядку
    Yg2=Quat2Ug(L2,kv);
    dps2(imas)=Yg2(1)+z(1); % Масиви похибок
    dteta2(imas)=Yg2(2)-z(2); % оцінювання
    dfi2(imas)=Yg2(3)-z(3); % кутів повороту основи
    % для метода третього порядку
    Yg3=Quat2Ug(L3,kv);
    dps3(imas)=Yg3(1)+z(1); % Масиви похибок
    dteta3(imas)=Yg3(2)-z(2); % оцінювання
    dfi3(imas)=Yg3(3)-z(3); % кутів повороту основи
    % для Адамса-Башфорта
    Yg4=Quat2Ug(L4,kv);
    dps4(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; % Массив моментів времени
    imas=imas+1;
end
% Побудова Графіків
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
stit1='ПОХИБКИ БСО (кутові мінути) - метод ПІКАРА';
sy1ab1='Метод 2-го порядку'; sxl1ab1=''; stit2='';
sy1ab2='Метод 3-го порядку'; sxl1ab2=''; stit3='';
sy1ab3='Метод 4-го порядку'; sxl1ab3='Час, с';
grafik91(tt,dps2*180*60/pi,dteta2*180*60/pi,dfi2*180*60/pi,stit1,sy1ab1,sxl1ab1,...
    dps3*180*60/pi,dteta3*180*60/pi,dfi3*180*60/pi,stit2,sy1ab2,sxl1ab2,...
    dps4*180*60/pi,dteta4*180*60/pi,dfi4*180*60/pi,stit3,sy1ab3,sxl1ab3);
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')

```

Нагадаємо, що комплекси програм працюють лише за умови використання бібліотеки програм роботи з кватерніонами і матрицями напрямних косинусів, поданих у Додатку.

6.4. Результати моделювання

Результати роботи вищезазначених програм моделювання подаються у виді графіків залежностей похибок алгоритмів від часу (див. рис. 6.1...6.8).

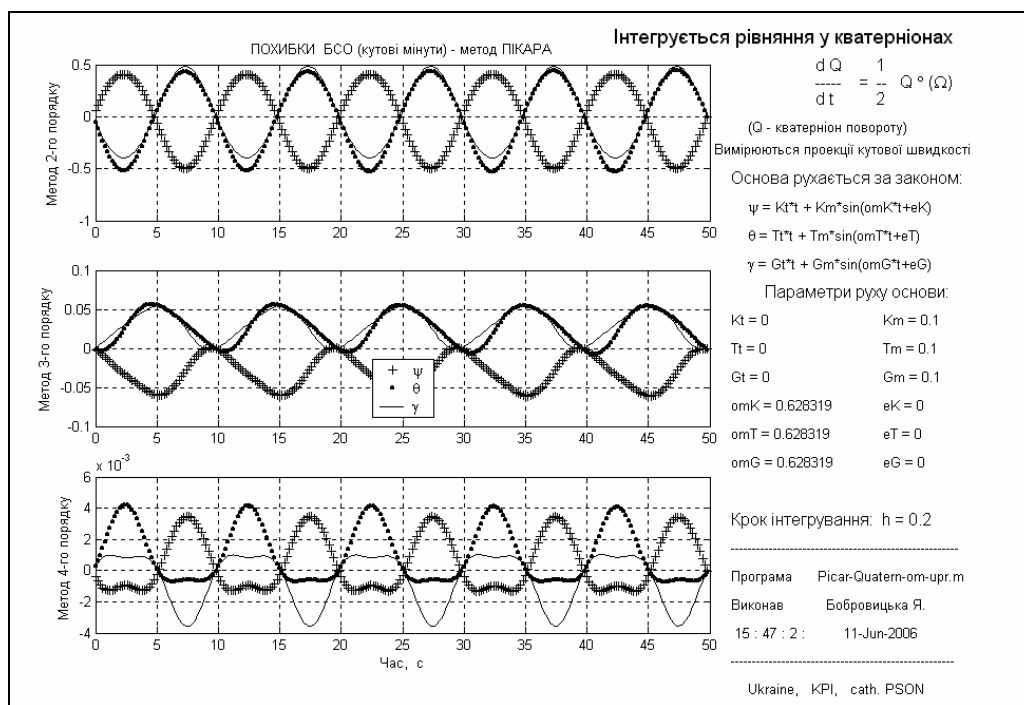


Рис. 6.1. Похибки алгоритмів методу Пікара для кватерніонного рівняння орієнтації, вимірюванні кутової швидкості при синхронній і синфазній хитавиці основи ($h=0,2$ с)

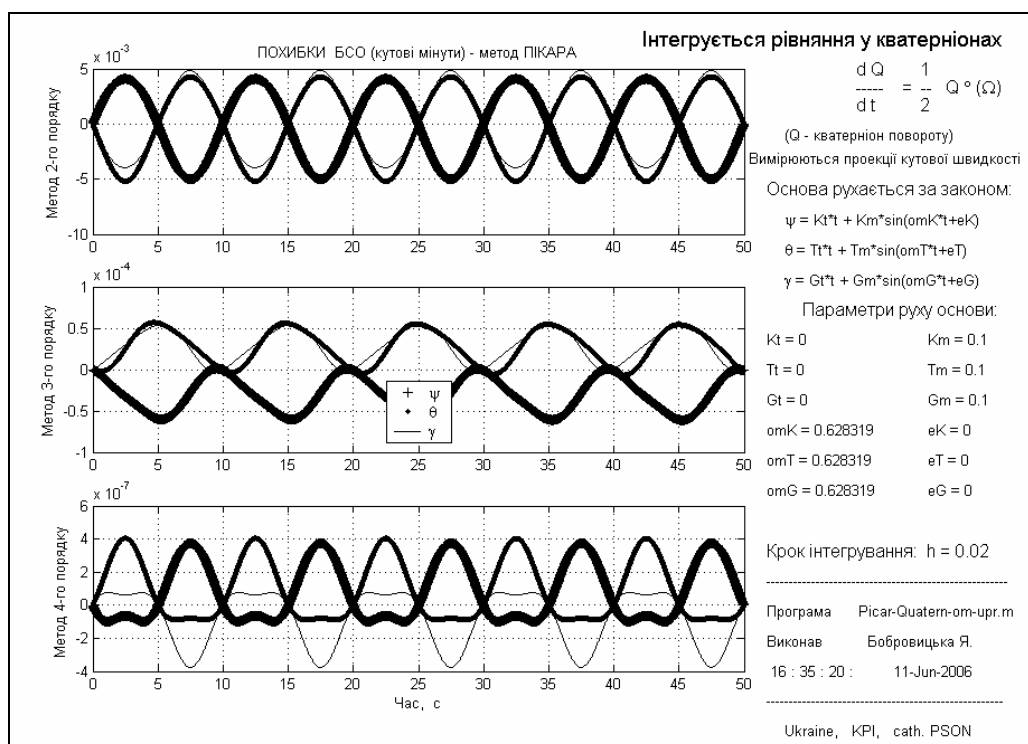


Рис. 6.2. Похибки алгоритмів методу Пікара для кватерніонного рівняння орієнтації, вимірюванні кутової швидкості при синхронній і синфазній хитавиці основи ($h=0,02$ с)

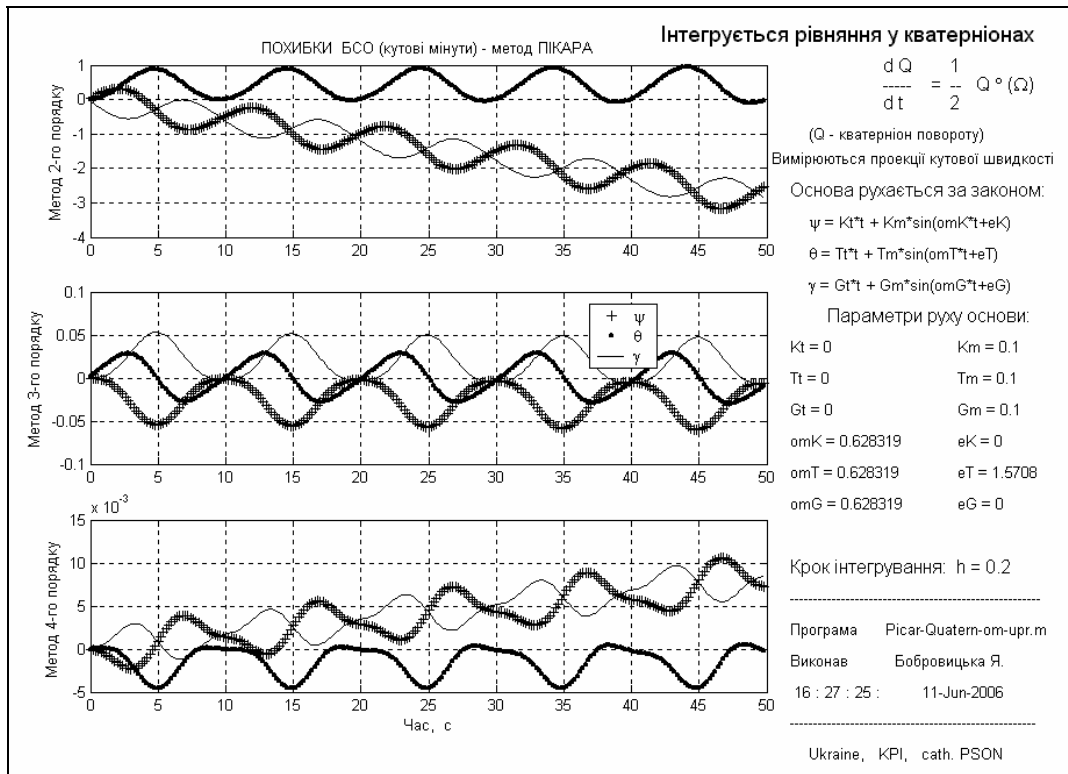


Рис. 6.3. Похибки алгоритмів методу Пікара для кватерніонного рівняння орієнтації, вимірюванні кутової швидкості при синхронній хитавиці основи з зсувом фаз 90° (h=0,2 c)

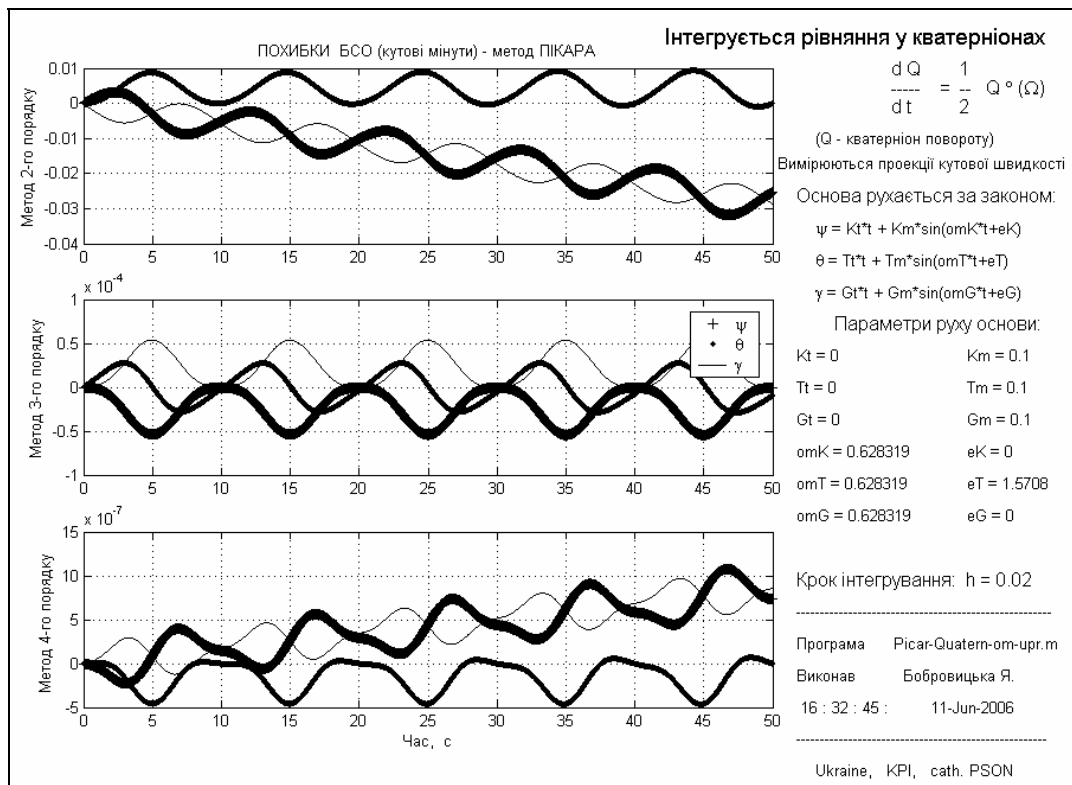


Рис. 6.4. Похибки алгоритмів методу Пікара для кватерніонного рівняння орієнтації, вимірюванні кутової швидкості при синхронній хитавиці основи з зсувом фаз 90° (h=0,02 c)

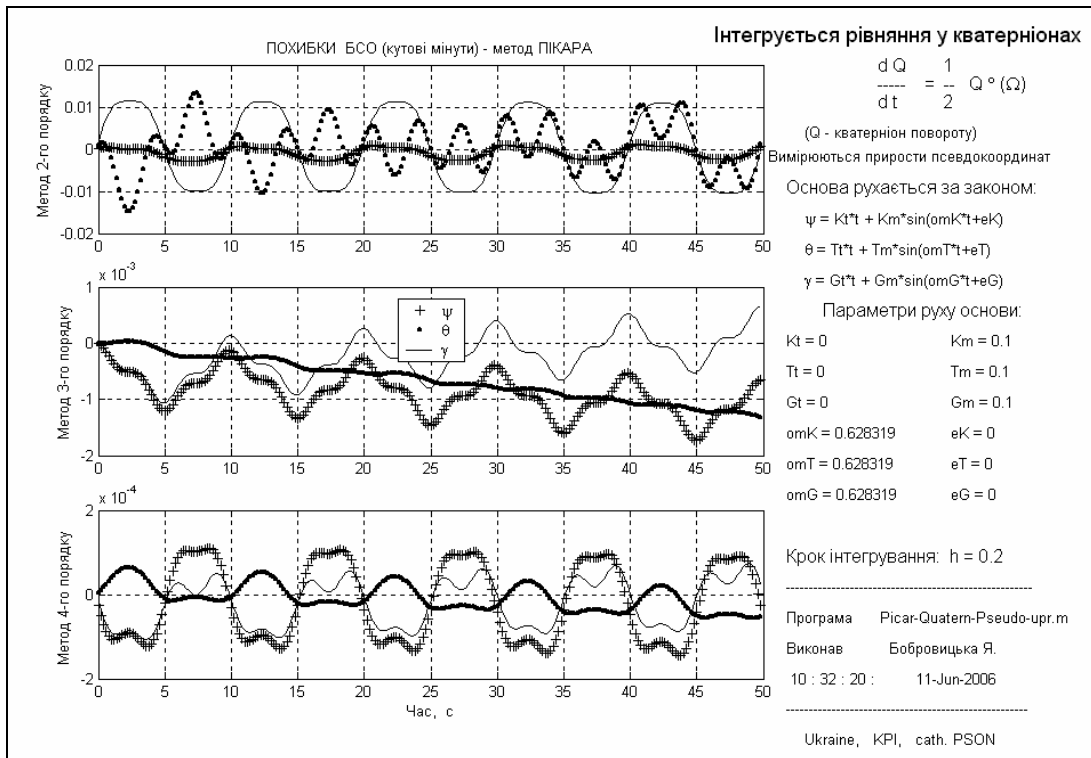


Рис. 6.5. Похибки алгоритмів методу Пікара для кватерніонного рівняння орієнтації, вимірюванні приростів псевдокоординат при синхронній і синфазній хитавиці основи (h=0,2 c)

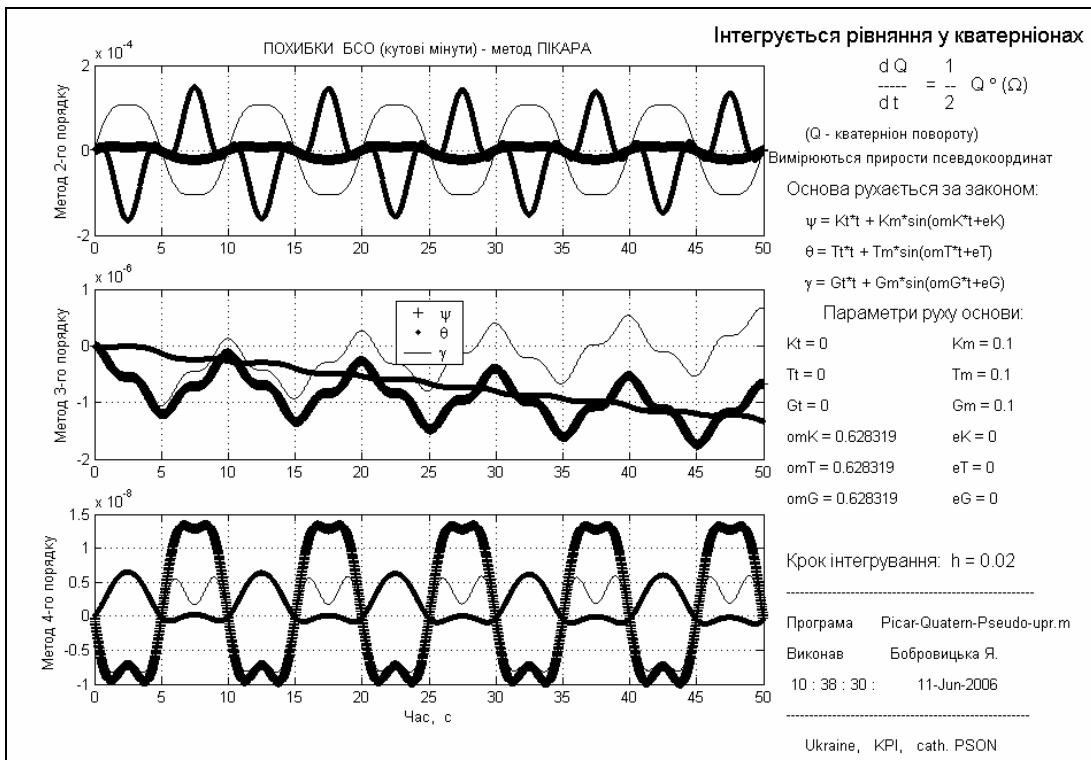


Рис. 6.6. Похибки алгоритмів методу Пікара для кватерніонного рівняння орієнтації, вимірюванні приростів псевдокоординат при синхронній і синфазній хитавиці основи (h=0,02 c)

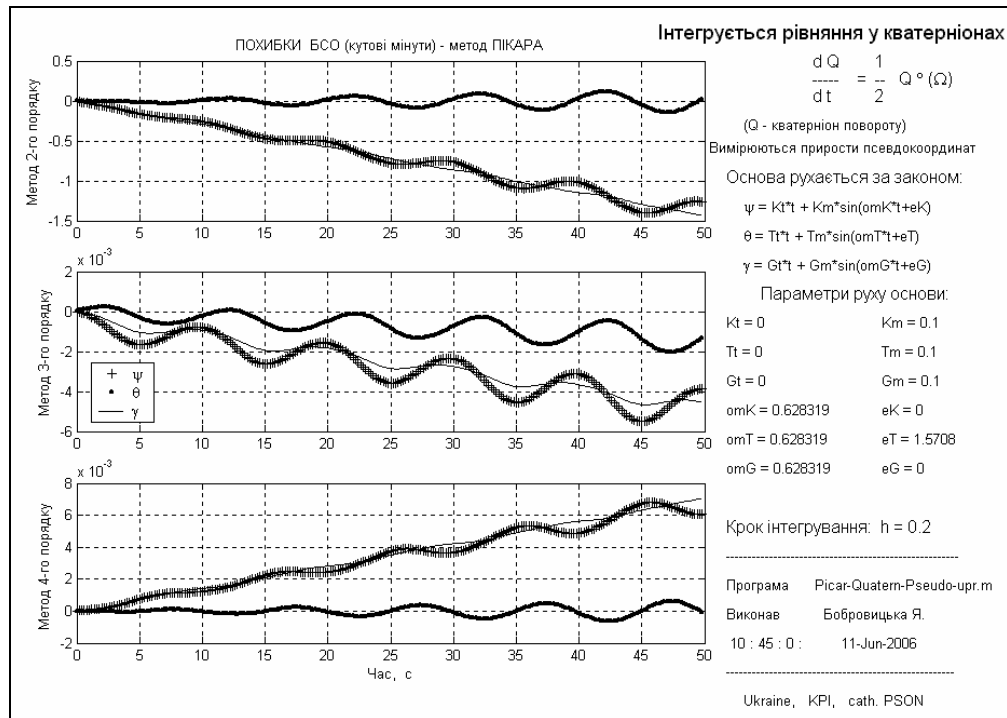


Рис. 6.7. Похибки алгоритмів методу Пікара для кватерніонного рівняння орієнтації, вимірюванні приростів псевдокоординат при синхронній хитавиці основи з зсувом фаз 90° ($h=0,2$ с)

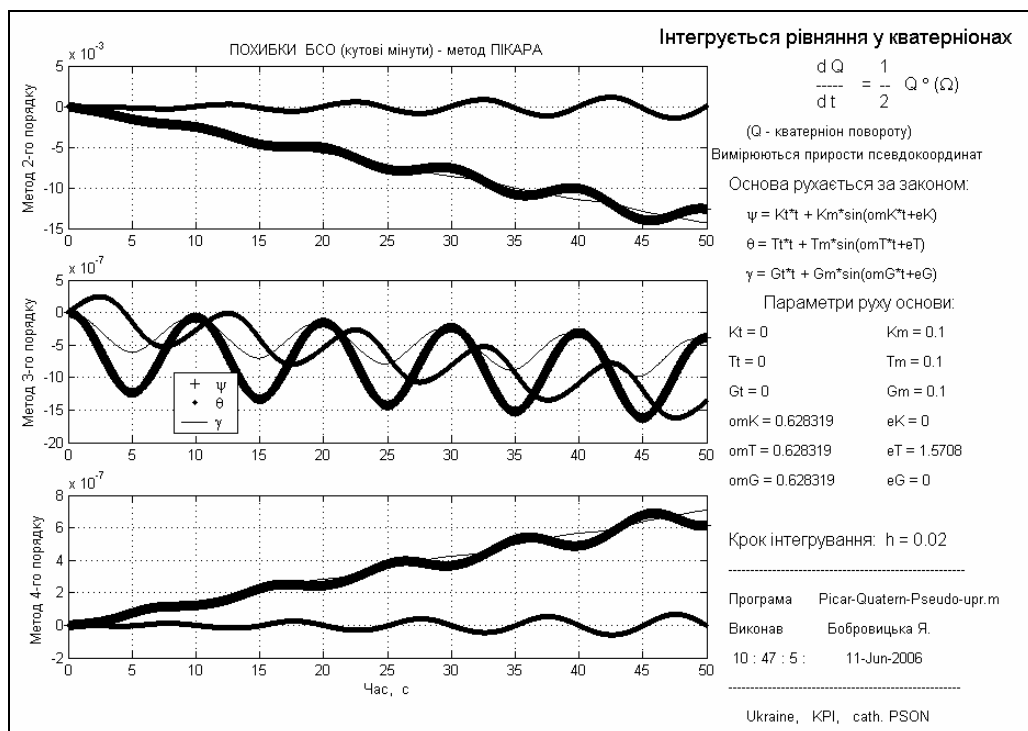


Рис. 6.8. Похибки алгоритмів методу Пікара для кватерніонного рівняння орієнтації, вимірюванні приростів псевдокоординат при синхронній хитавиці основи з зсувом фаз 90° ($h=0,02$ с)

Значення узагальнених параметрів похибок визначення кутів за наведеними результатами експериментів на програмних моделях показані у таблицях 6.1 і 6.2.

Таблиця 6.1. Параметри похибок алгоритмів методу Пікара при синфазній хитавиці основи

Вид руху основи	Крок	Порядок методу	Вид параметра	Вимірювання проєкцій кутової швидкості	Вимірювання приростів псевдокоординат
Синфазні і синхронні коливання основи з трьох кутів з періодом 10 с і амплітудою 0,1 радіан	0,2с	2	Амплітуда	$(5 \cdot 10^{-1})'$	$(1 \cdot 10^{-2})'$
			Дрейф з ψ	0	0
			Дрейф з ϑ	0	0
			Дрейф з γ	0	0
		3	Амплітуда	$(2,5 \cdot 10^{-2})'$	$(5 \cdot 10^{-5})'$
			Дрейф з ψ	0	$-(1 \cdot 10^{-5})'/c$
			Дрейф з ϑ	0	$-(3 \cdot 10^{-5})'/c$
			Дрейф з γ	0	$(1 \cdot 10^{-5})'/c$
		4	Амплітуда	$(2,5 \cdot 10^{-3})'$	$(1 \cdot 10^{-4})'$
			Дрейф з ψ	0	0
			Дрейф з ϑ	0	0
			Дрейф з γ	0	0
	0,02с	2	Амплітуда	$(5 \cdot 10^{-3})'$	$(1 \cdot 10^{-4})'$
			Дрейф з ψ	0	0
			Дрейф з ϑ	0	0
			Дрейф з γ	0	0
		3	Амплітуда	$(2,5 \cdot 10^{-5})'$	$(5 \cdot 10^{-7})'$
			Дрейф з ψ	0	$-(1 \cdot 10^{-8})'/c$
			Дрейф з ϑ	0	$-(3 \cdot 10^{-8})'/c$
			Дрейф з γ	0	$(1 \cdot 10^{-8})'/c$
4		Амплітуда	$(2,5 \cdot 10^{-7})'$	$(1 \cdot 10^{-8})'$	
		Дрейф з ψ	0	0	
		Дрейф з ϑ	0	0	
		Дрейф з γ	0	0	

Таблиця 6.2. Параметри похибок алгоритмів методу Пікара при хитавиці основи з зсувом фаз 90°

Вид руху основи	Крок	Порядок методу	Вид параметра	Вимірювання проєкцій кутової швидкості	Вимірювання приростів псевдокоординат
Синфазні коливання основи з трьох кутів з періодом 10 с і амплітудою 0,1 радіан і з зсувом фаз 90°	0,2с	2	Амплітуда	$(5 \cdot 10^{-1})'$	$(1 \cdot 10^{-2})'$
			Дрейф з ψ	$-(5 \cdot 10^{-2})'/c$	$-(3 \cdot 10^{-2})'/c$
			Дрейф з ϑ	0	0
			Дрейф з γ	$-(6 \cdot 10^{-2})'/c$	$-(3 \cdot 10^{-2})'/c$
		3	Амплітуда	$(2,5 \cdot 10^{-2})'$	$(5 \cdot 10^{-4})'$
			Дрейф з ψ	0	$-(8 \cdot 10^{-5})'/c$
			Дрейф з ϑ	0	$-(2 \cdot 10^{-5})'/c$
			Дрейф з γ	0	$-(8 \cdot 10^{-5})'/c$
		4	Амплітуда	$(2,5 \cdot 10^{-3})'$	$(1 \cdot 10^{-4})'$
			Дрейф з ψ	$(1,6 \cdot 10^{-4})'/c$	$(1,2 \cdot 10^{-4})'/c$
			Дрейф з ϑ	0	0
			Дрейф з γ	$(1,6 \cdot 10^{-4})'/c$	$(1,4 \cdot 10^{-4})'/c$
	0,02с	2	Амплітуда	$(5 \cdot 10^{-3})'$	$(1 \cdot 10^{-4})'$
			Дрейф з ψ	$-(5 \cdot 10^{-4})'/c$	$-(2,6 \cdot 10^{-4})'$
			Дрейф з ϑ	0	0
			Дрейф з γ	$-(5 \cdot 10^{-4})'/c$	$-(2,6 \cdot 10^{-4})'$
		3	Амплітуда	$(2,5 \cdot 10^{-5})'$	$(5 \cdot 10^{-7})'$
			Дрейф з ψ	0	$-(1 \cdot 10^{-8})'/c$
			Дрейф з ϑ	0	$-(3 \cdot 10^{-8})'/c$
			Дрейф з γ	0	$-(1 \cdot 10^{-8})'/c$
		4	Амплітуда	$(2,5 \cdot 10^{-7})'$	$(1 \cdot 10^{-8})'$
			Дрейф з ψ	$(1,6 \cdot 10^{-8})'/c$	$(1,2 \cdot 10^{-8})'/c$
			Дрейф з ϑ	0	0
			Дрейф з γ	$(1,6 \cdot 10^{-8})'/c$	$(1,2 \cdot 10^{-8})'/c$

6.5. Порівняння алгоритмів чисельного інтегрування кінематичних рівнянь орієнтації у кватерніонах за методом Пікара

При дослідженні методів чисельного інтегрування кінематичних рівнянь орієнтації твердого тіла за алгоритмами, поданими в існуючій літературі ([7], [8]), були виявлені деякі невідповідності, пов'язані з неузгодженістю заявлених порядків методу чисельного інтегрування з результатами, що виявилися при моделюванні цих алгоритмів на ЕОМ (під порядком методу інтегрування розуміється показник степеня у степеневій залежності похибки інтегрування від кроку). Тому виникла потреба більш ретельно побудувати алгоритми чисельного інтегрування, які приведені вище.

Для порівняння подамо алгоритми, які наведені в [7]:

$$\begin{aligned}
 & \text{- другого порядку: } \mathbf{N}_{n+1} = 1 + \frac{1}{2} \nabla - \frac{1}{8} |\nabla|^2; \\
 & \text{- третього порядку: } \mathbf{N}_{n+1} = 1 + \frac{1}{2} \nabla - \frac{1}{8} |\nabla|^2 - \frac{1}{48} |\nabla|^2 \nabla + \frac{1}{24} (\nabla \times \nabla^2); \\
 & \text{- четвертого порядку: } \mathbf{N}_{n+1} = 1 + \frac{1}{2} \nabla - \frac{1}{8} |\nabla|^2 + \frac{1}{24} (\nabla \times \nabla^2) - \\
 & \quad - \frac{1}{48} |\nabla|^2 \nabla + \frac{1}{48} (\nabla \times \nabla^3) - \frac{1}{192} |\nabla|^2 \nabla^2 + \frac{1}{384} |\nabla|^4. \tag{6.50}
 \end{aligned}$$

У [8] відрізняється лише алгоритм четвертого порядку. Він має вигляд

$$\begin{aligned}
 \mathbf{N}_{n+1} = & 1 + \frac{1}{2} \nabla - \frac{1}{8} (\nabla)^2 - \frac{1}{48} \nabla \cdot (\nabla)^2 + \frac{1}{24} \nabla \times \nabla^2 + \\
 & + \frac{1}{48} \nabla \times \nabla^3 + \frac{1}{16} (\nabla^2)^2 + \frac{1}{48} \nabla \cdot (\nabla \cdot \nabla^2) - \frac{1}{96} (\nabla)^2 \nabla^2 + \\
 & + \frac{1}{64} \nabla \cdot (\nabla^2)^2 + \frac{1}{144} \nabla \times \nabla^4. \tag{6.51}
 \end{aligned}$$

Порівняння виведених формул із формулами, що подані у літературі [7], [8], приводить до таких висновків. В усіх випадках формули для методу третього порядку повністю збігаються. Деяка відмінність спостерігається в алгоритмах методів другого порядку. Найбільш суттєво відрізняються формули для алгоритмів методів четвертого порядку. Важливо встановити, чим обумовлені виявлені відмінності і чи відповідає залежність похибки інтегрування за цими алгоритмами порядку метода, що заявлений для цього алгоритму.

Задля перевірки точності алгоритмів було проведено моделювання згаданих алгоритмів на ЕОМ у середовищі Matlab. Як тестовий, у програмах моделювання був прийнятий коливальний (синусоїдальний) синхронний і синфазний рух основи з усіх трьох кутових координат з періодом 10 с, амплітудою 0,1 радіан (приблизно 6°). Для виявлення залежності похибки алгоритму від кроку інтегрування моделювання проводилося при двох значеннях кроку $h = 0,2$ с і

$h = 0,02$ с. Рис. 6.9, 6.10 подають похибки за алгоритмами (6.50) з [7], а на рис. 6.11, 6.12 наведені похибки за алгоритмом (6.51) з [8].

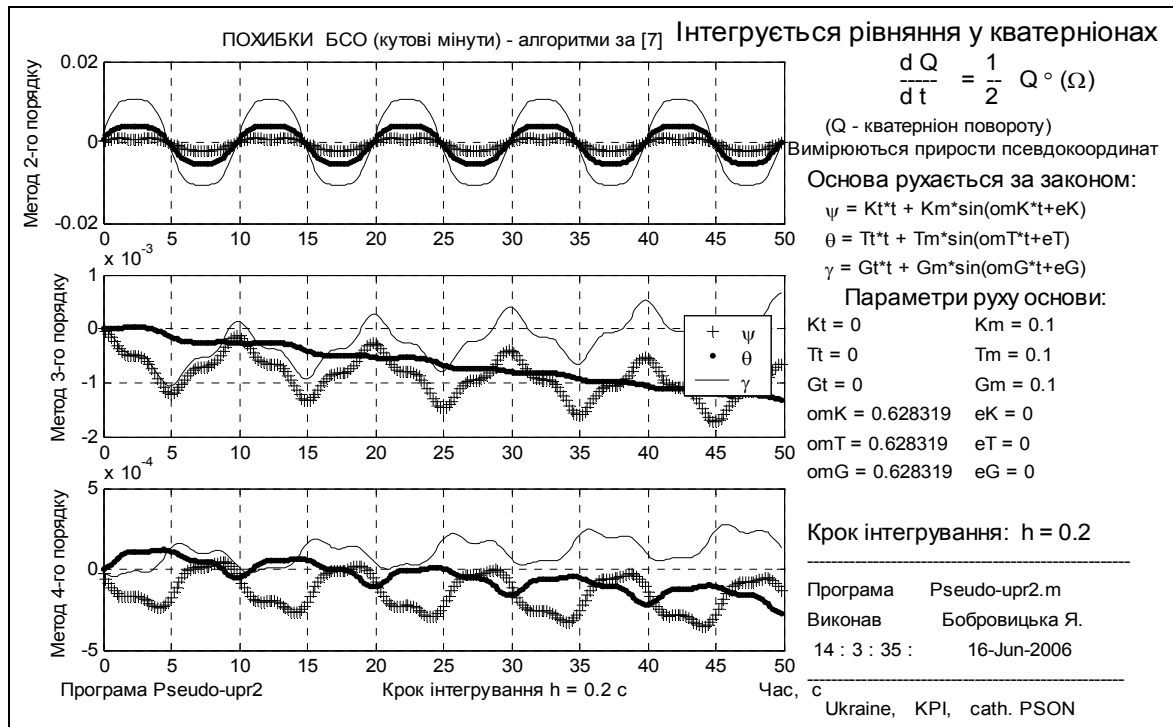


Рис. 6.9. Похибки алгоритмів за [7] при значенні кроку $h=0,2$ с

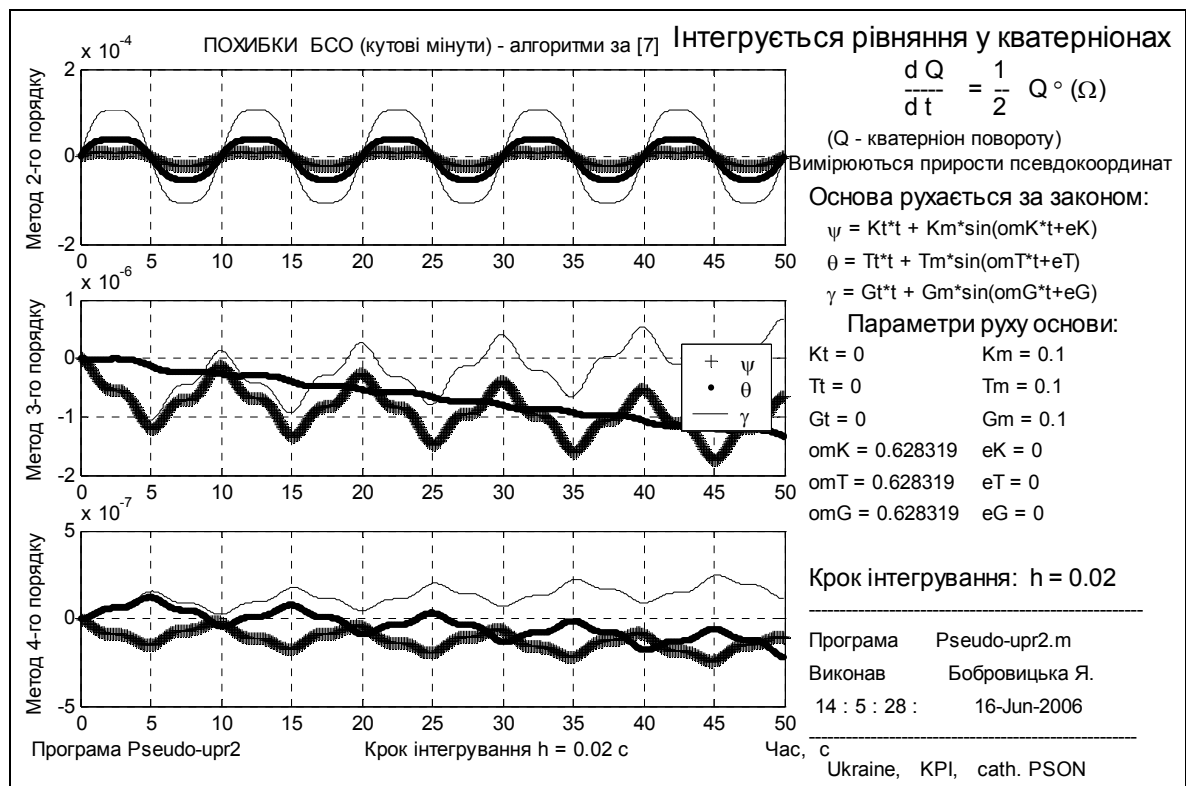
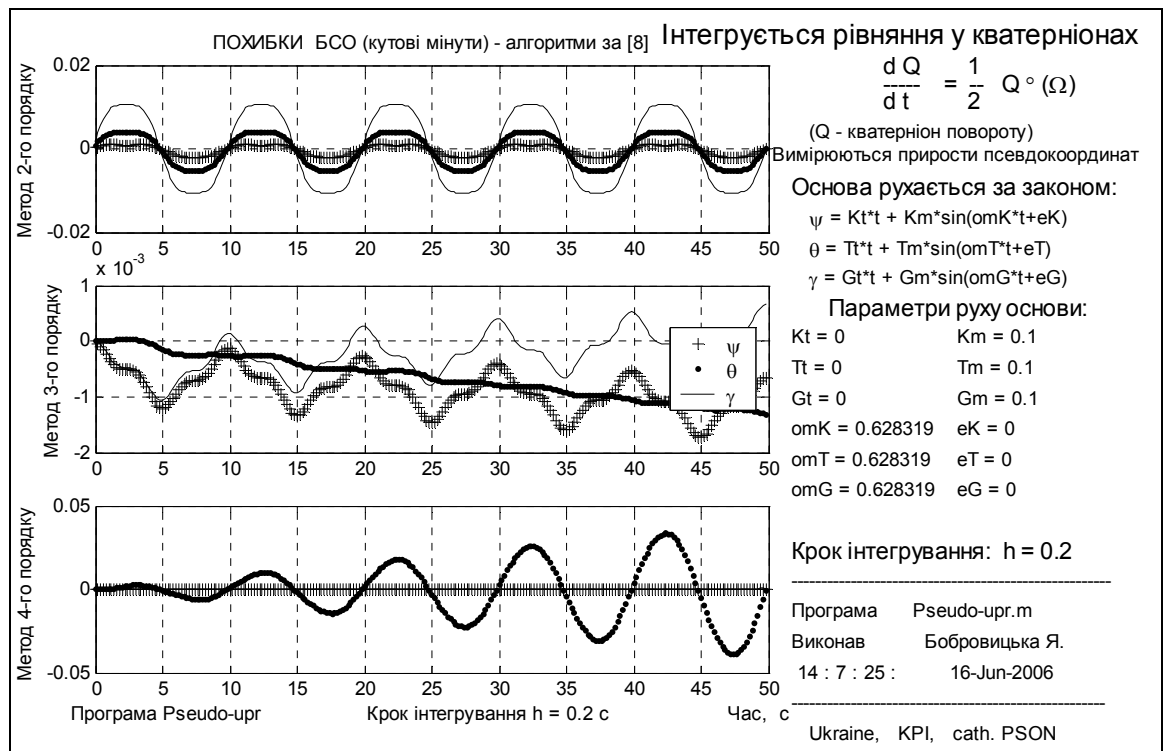
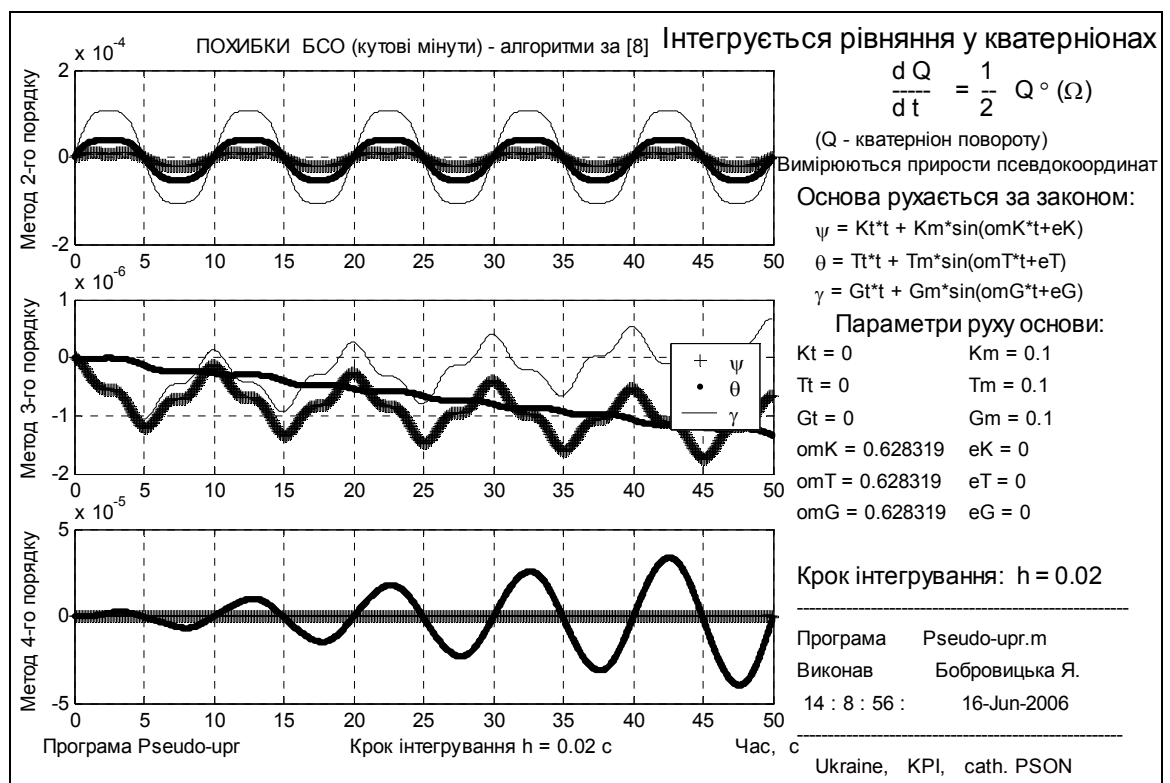


Рис. 6.10. Похибки алгоритмів за [7] при значенні кроку $h=0,02$ с

Рис. 6.11. Похибки алгоритмів за [8] при значенні кроку $h=0,2$ сРис. 6.12. Похибки алгоритмів за [8] при значенні кроку $h=0,02$ с

Порівнюючи амплітуди похибок того самого алгоритму при різних значеннях кроку інтегрування, можна встановити порядок алгоритму. Алгоритм другого порядку при зменшенні кроку у 10 разів, має зменшувати похибку інтегрування у 10^2 разів, алгоритм третього порядку – у 10^3 разів, а четвертого – у 10^4 разів. Легко впевнитися, що усі розглянуті алгоритми

методів другого і третього порядку забезпечують залежність похибки від кроку з показником 2 і 3 відповідно, тобто репрезентують методи саме відповідного порядку. Результати моделювання доводять, що алгоритм метода четвертого порядку, який наведений у [7], залежить від кроку інтегрування як метод третього порядку, тобто не є алгоритмом четвертого порядку. Алгоритм, поданий у [8], взагалі є нестійким. Скоріше за все він містить суттєву помилку.

Ці обставини відзначено у [13].

Єдиний з розглянутих алгоритмів методів четвертого порядку, який забезпечує степеневу залежність похибки від кроку саме з показником степеня 4, – це розроблений алгоритм за формулою (6.49а).

Цікавою особливістю, яку виявляє моделювання, є те, що за прийнятого руху основи у вигляді синхронних і синфазних коливань, методи парного порядку практично не мають дрейфу похибки, а методи непарного порядку утворюють складову похибки, яка збільшується пропорційно часу - дрейф. Як показує модельний експеримент, кутова швидкість дрейфу методу третього порядку складає величину порядку $1 \cdot 10^{-5}$ кутові мінути у хвилину для кроку $h = 0,2$ с і $1 \cdot 10^{-8}$ кутові мін./хв. для кроку $h = 0,02$ с.

6.6. Висновки

1. Розроблені алгоритми на основі методу Пікара для кватерніонного рівняння орієнтації є стійкими і забезпечують достатню точність визначення параметрів орієнтації.

2. У випадку вимірювання кутової швидкості чисельне інтегрування рівняння орієнтації не призводить до появи дрейфів у жодному з методів.

3. Алгоритми методу Пікара для кватерніонного рівняння при вимірюванні псевдокоординат підвищують точність інтегрування від 2 до 10 разів за метод Пікара для рівняння Пуассона.

4. Амплітуди похибок у випадку вимірювання приростів псевдокоординат менші за такі для випадку вимірювання кутової швидкості приблизно на порядок.

5. Застосування вимірювачів приростів псевдокоординат дещо змінює характер прояву дрейфів похибок.

6. Побудовані уточнені, порівняно з відомими з оприлюдненої літератури, алгоритми чисельного інтегрування кінематичних рівнянь у кватерніонах за методом Пікара. Зокрема, вперше розроблено алгоритм четвертого порядку (відомі алгоритми не є алгоритмами четвертого порядку). Запропонована методика побудови алгоритмів є надійною і дозволяє будувати алгоритми й більш високого порядку.

Загальні висновки

1. Застосування відомих методів прогноз-корекції і метода послідовних наближень Пікара забезпечує утворення алгоритмів різних порядків для будь-яких видів рівнянь орієнтації.
2. Моделювання на ЕОМ показує дієвість алгоритмів побудованих на основі згаданих методів.
3. Встановлено хибність деяких алгоритмів, що наведені у відомій літературі. Розроблені методи, які дозволяють уникнути цих хиб.
4. Найкращим за точністю згідно результатам моделювання є метод Пікара застосований до кватерніонного рівняння орієнтації при вимірюванні приростів псевдокоординат.
5. Наведені програми моделювання похибок алгоритмів орієнтації БСО є достатньо ефективними і можуть бути застосовані для моделювання інших алгоритмів БСО.

Література

1. Божко А. П., Бранец В. Н., Захаров Ю. М., Шмыглевский И. П. Применение кватернионов в теории конечного поворота твердого тела // Изв. АН СССР. Механика твердого тела. – 1971. – № 1.
2. Богданский Е. Д., Фурман В. Д. О погрешностях численного интегрирования кинематических уравнений Пуассона // Космические исследования. – 1970. – Вып. 6.
3. Бранец В. Н. О точности решения кинематических уравнений. Уравнение ошибок // Космические исследования. – 1982. – Т. 20, вып. 2.
4. Бранец В. Н. О точности решения кинематических уравнений. Исследование квазикординат // Космические исследования. – 1982. – Т. 20, вып. 3.
5. Бранец В. Н., Шмыглевский И. П. Кинематические задачи ориентации во вращающейся системе координат // Изв. АН СССР. Механика твердого тела. – 1972. – № 6.
6. Бранец В. Н., Шмыглевский И. П. Применение кватернионов в управлении угловым положением твердого тела // Изв. АН СССР. Механика твердого тела. – 1972. – №4.
7. Бранец В. Н., Шмыглевский И. П. Применение кватернионов в задачах ориентации твердого тела. – М.: Наука, 1973. – 303 с.
8. Бранец В. Н., Шмыглевский И. П. Введение в теорию бесплатформенных инерциальных систем. – М.: Наука, 1992. – 280 с.
9. Диментберг Ф. М., Севостьянова Е. Г. К теории конечных перемещений твердого тела // Изв. АН СССР. Механика твердого тела. – 1969. - № 2.
10. Дмитроченко Л. А., Лопатин В. И. Особенности алгоритмов БИНС. // Вопросы управления космическими аппаратами. – М.: Мир, 1975. – С. 95-117.
11. Инерциальные системы без гиросtabilизированной платформы: Обзор. Пер. с англ. // Вопросы ракетной техники. – 1967. - № 1(145). – С. 61-77.
12. Климов Д. М., Рабинович Ю. И. О кинематических ошибках инерциальной системы навигации // Изв. АН СССР. Механика. – 1965. – № 6
13. Лазарев Ю. Ф., Бобровицька Я. Г. Зіставлення алгоритмів чисельного інтегрування рівнянь орієнтації твердого тіла у кватерніонах за методом Пікара // Наукові вісті НТУУ "КПІ". – 2006. – № 6. – С. 113 – 118.
14. Лебедев Д. В. Управление ориентацией твердого тела с использованием параметров Родрига-Гамильтона // Автоматика. – 1974. - № 4. – с. 29-32.
15. Лурье А. И. Аналитическая механика. – М.: Физматгиз. 1961.

16. Онищенко С. М. Применение гиперкомплексных чисел в теории БИНС \ \ Сб. научных трудов Ин-та кибернетики АН УССР. – Киев, 1982. – С. 44-62.
17. Оттен. Инерциальные навигационные системы без гиросtabilизированной платформы. Перев. с англ. \ \ Вопросы ракетной техники. – 1967. – № 12.
18. Панов А. П. Синтез методов вычисления координат вектора ориентации \ \ Кибернетика и выч. техника. – 1979. – Вып. 43. – С. 122-130.
19. Панов А. П. Кинематические дифференциальные уравнения для собственных векторов операторов вращения твердого тела \ \ Изв. АН ССР, ММТ. – 1985. – №4. – С. 26-33.
20. Панов А. П. Асимптотические оценки погрешностей методов вычисления параметров ориентации твердого тела \ \ Кибернетика в выч. техника. – 1980. – Вып. 47. – С. 59-71.
21. Панов А. П. О выборе кинематических параметров и уравнений вращения для численного интегрирования в ЦВМ \ \ Кибернетика в выч. техника. – 1984. – Вып. 62. – С. 104-111.
22. Ткаченко А. И. Повышение точности вычисления кинематических параметров \ \ Кибернетика в выч. техника. – 1973. – Вып. 19. – С. 117–121.
23. Ткаченко А. И. Определение ориентации и калибровка пространственного измерителя угловой скорости с использованием угловой информации \ \ Изв. АН СССР, ММТ. – 1983. - №3. – С. 19–23.
24. Челноков Ю. И. Кватернионные алгоритмы систем пространственной инерциальной навигации \ \ Изв. АН СССР, ММТ. – 1983. - № 6. – С. 14–21.
25. Эдвардс А. Бесплатформенные инерциальные навигационные системы \ \ Вопросы ракетной техники. – 1973. - № 5. – С. 47–70.

Додаток. Бібліотека допоміжних програм

```
% ContentsQwat.m Зміст блоку процедур класу 'quatern'

% Автор Лазарев Ю.Ф. 5-07-2003
% Процедури-методи класу 'quatern'
% 1. Конструктори кватерніонів
% quatern(x) - основний конструктор кватерніонів
% за заданим вектором 'x' довжиною 4
% pov2qw(ug,NKO) - формує кватерніон повороту за кутом Ейлера
% и напрямними косинусами осі повороту
% ug2quat(x1,k1,x2,k2,x3,k3) - формує кватерніон
% за заданими значеннями трьох кутів (x1, x2, x3)
% послідовних поворотів та номерами осей (k1, k2, k3),
% навколо яких ці повороти здійснюються
% quate(x,k) - формування ЕЛЕМЕНТАРНОГО Кватерніона ПОВОРОТА
% (який відповідає одному повороту навколо
% однієї з координатних декартових осей)
% за заданим значення куту повороту (x) та номера (k)
% осі, навколо якої здійснюється поворот
% norm2qw(Norm,Arg,Ncos) - формує кватерніон повороту за його нормалізованим поданням
% нормою, аргументом и напрямними косинусами осі повороту
% vect2quat(x) - перетворює трикомпонентний вектор 'x' у вектор-кватерніон
% mnk2quat(y) - перетворює матрицю напрямних косинусів 'y'
% у кватерніон
% 2. Процедури формування символічного зображення кватерніона
% char(qw) -формує символічний рядок, зображуючий кватерніон
% display(qw) - здійснює неявне виведення на екран
% символічного подання кватерніона
% double(q) - виділення з кватерніону q скалярної частини
% і вектора векторної частини
% 3. Вбудовані операції з кватерніонами
% uplus(q) - (+) унарний плюс перед кватерніоном
% uminus(q) - (-) унарний плюс перед кватерніоном
% plus(q1,q2) - (+) обчислення СУМИ двох кватерніонів
% minus(q1,q2) - (-) обчислення РІЗНИЦІ двох кватерніонів
% mtimes(q1,q2) - (*) обчислення ДОБУТКУ двох кватерніонів
% mrdivide(q1,q2) - (/) операція ПРАВОГО ДІЛЕННЯ двох кватерніонів
% mldivide(q1,q2) - (\) операція ЛІВОГО ДІЛЕННЯ двох кватерніонів
% ctranspose(q1) - (') обчислення кватерніону, СПРЯЖЕНОГО до заданого
% transpose(q1) - (.) операція перетворення кватерніону у матрицю-стовпець
% 4. Інші операції
% inv(q) - ОБЕРНЕННЯ кватерніону
% qw2norm(qw) - формує нормалізовану форму подання кватерніону
%
% quat2ug(qw,kv) - перетворює кватерніон у кути Ейлера
% quat2mnk(x1) - перетворює кватерніон у матрицю напрямних косинусів
%
% Процедури, які не є методами класу 'quatern'
```

```
function y=quatern(x)
% Головний конструктор кватерніонів
% Лазарев Ю.Ф. 18-07-1998р.
k=length(x);
if nargin==0
    y.c=[];        y.v=[];        p=class(y,'quatern');
elseif isa(x,'quatern');
    y=x;
elseif k==4
```



```

        y.c=x(1);      y.v=[x(2),x(3),x(4)];    y=class(y,'quatern');
elseif k==1
        y.c=x;        y.v=zeros(1,3);      y=class(y,'quatern');
else
        disp('Довжина вхідного вектора має дорівнювати 4 або 1 !');
end

```

```

function qw=rov2qw(ug,NKO)
% Процедура POV2QW формує кватерніон повороту за кутом
% и напрямними косинусами осі повороту
%Лазарев Ю.Ф., каф. ПСОИ, К П і Початок 31-01-2000 р.
u=ug/2;      k=length(NKO);
if k==3
    norm=NKO*NKO';
    if norm<1.1
        c=NKO/sqrt(norm);      q=zeros(1,4);
        q(1)=cos(u);  q(2:4)=sin(u)*c;      qw=quatern(q);
    else
        disp('Завеликий модуль! Перевірте Напрямні Косинуси!');
    end
else
    disp('Замала довжина вектора Напрямних Косинусів!');
end

```

```

function y = ug2quat(x1,k1,x2,k2,x3,k3)
% Процедура розрахунку Кватерніону (Y)
% за заданими значеннями трьох кутів (x1, x2, x3)
% послідовних поворотів та номерами осей (k1, k2, k3),
% навколо яких ці повороти здійснюються
% Лазарев Ю.Ф. 18-07-1998р.
% Останні змінювання 4-12-2001
y = quate(x1,k1)*quate(x2,k2)*quate(x3,k3)

```

```

function y=quate(x,k)
% Формування ЕЛЕМЕНТАРНОГО Кватерніона ПОВОРОТА
% (який відповідає одному повороту навколо
% однієї з координатних декартових осей)
% за заданим значення куту повороту (x) та номера (k)
% осі, навколо якої здійснюється поворот
% Лазарев Ю.Ф. 4-12-2001
y(1) = cos(x/2);% Скалярна частина кватерніона
% Далі визначається векторна частина кватерніону
if k==1
    y(2) = sin(x/2);      y(3) = 0;      y(4) = 0;
end
if k==2
    y(2) = 0;      y(3) = sin(x/2);      y(4) = 0;
end
if k==3
    y(2) = 0;      y(3) = 0;      y(4) = sin(x/2);
end
y=quatern(y); % Закінчення процедури 'QUATE'

```

```

function qw=norm2qw(Norm,Arg,Ncos)

```

```

% Процедура NORM2QW формує кватерніон повороту за його нормалізованим поданням
% нормою, аргументом и напрямними косинусами осі повороту
% Лазарев Ю.Ф., 5-07-2003
k=length(Ncos);
if k==3
    norm=Ncos*Ncos.';
    if norm>0.9999 & norm<1.0001
        c=Ncos;    q=zeros(1,4); q(1)=cos(Arg);
        q(2:4)=sin(Arg)*c;    qw=quatern(Norm*q);
    else
        disp('Напрявні Косинуси не знормовані!');
    end
else
    disp('Довжина вектора Напрявних Косинусів має дорівнювати 3!');
end

```

```

function qv= vect2quat(x)
% Перетворює трикомпонентний вектор 'x' у вектор-кватерніон 'qv'
% qv = vect2quat(x) обчислює кватерніон "qv"
% за заданим трикомпонентним вектором-рядком "x"
% Лазарев Ю.Ф. 4-12-2001
y(1) = 0;    y(2) = x(1);    y(3) = x(2);    y(4) = x(3);
qv=quatern(y); % перетворення до кватерніону

```

```

function [qw,d,e]= mnk2quat(y)
% mnk2quat перетворює матрицю напрямних косинусів "y"
% у кватерніон qw
% Додатково видає значення кута 'd' повороту Ейлера
% і вектор напрямних косинусів 'e' Ейлерової осі
% Лазарев Ю.Ф. 5-12-2001
cd=(trace(y)-1)/2; % Знаходження косинуса кута Ейлера
d=acos(cd)/2; % Кут повороту Ейлера
sd2=2*sqrt(1-cd^2); e1=(y(3,2)-y(2,3))/sd2;
e2=(y(1,3)-y(3,1))/sd2; e3=(y(2,1)-y(1,2))/sd2; e=[e1,e2,e3];
cd2=cos(d);    sd=sin(d);
x(1)=cd2;    x(2:4)=sd*[e1,e2,e3];
qw=quatern(x);    % Кінець процедури "mnk2quat"

```

```

function s=char(qw)
% Формує символний рядок, зображуючий кватерніон

% Лазарев Ю.Ф. 18-07-1998р.
% Перевірено. Працює правильно.
c=[qw.c,qw.v];
if all(c==0)
    s='0';
else
    s=[];
    if c(1)~=0
        s=[s num2str(c(1))];
    end
    if c(2)~=0
        if c(2)>0
            s=[s '+'];
        end
        s=[s num2str(c(2)) '*I'];
    end
end

```

```

    if c(3)~=0
        if c(3)>0
            s=[s '+'];
        end
        s=[s num2str(c(3)) '*J'];
    end
    if c(4)~=0
        if c(4)>0
            s=[s '+'];
        end
        s=[s num2str(c(4)) '*K'];
    end
end
end

```

```

function display(qw)
% Здійснює неявне виведення на екран символного подання кватерніону
% Лазарєв Ю.Ф. 18-07-1998р.
disp("");      s=char(qw);   disp(['',inputname(1),'=',s,'];); disp("");

```

```

function [v,c]=double(q)
% Процедура виділення з заданого кватерніону q
% його скалярної частини c і векторної частини v
% послідовних поворотів

```

```

% Лазарєв Ю.Ф. 18-07-1998р.
q=quatern(q);      c=q.c;      v=q.v;

```

```

function q=uplus(q1)
% Арифметична операція (+) УНАРНИЙ ПЛЮС перед кватерніоном
% тобто розрахунку Кватерніону q=+q1

```

```

% Лазарєв Ю.Ф.      18-07-1998р.
% Останні змінювання 4-12-2001
q1=quatern(q1);      q.c=q1.c;      q.v=q1.v;
q=class(q,'quatern');

```

```

function q=uminus(q1)
% Арифметична операція (-) ЗМІНЮВАННЯ ЗНАКА перд кватерніоном
% тобто розрахунку Кватерніону q=-q1

```

```

% Лазарєв Ю.Ф.      18-07-1998р.
% Останні змінювання 4-12-2001
q1=quatern(q1);      q.c=-q1.c;      q.v=-q1.v;
q=class(q,'quatern');

```

```

function q=plus(q1,q2)
% Процедура обчислення СУМИ двох кватерніонів
% тобто розрахунку Кватерніону (q)
% за заданими значеннями двох кватерніонів (q1 і q2)

```

```

% Лазарєв Ю.Ф.      18-07-1998р.
% Останні змінювання 4-12-2001
q1=quatern(q1);      q2=quatern(q2);

```

```
q.c=q1.c+q2.c;      q.v=q1.v+q2.v;
q=class(q,'quatern');
```

```
function q=minus(q1,q2)
% Процедура обчислення РІЗНИЦІ двох кватерніонів
% тобто розрахунку Кватерніону (q)
% за заданими значеннями двох кватерніонів (q1 і q2)
% Лазарев Ю.Ф      4-12-2001
q1=quatern(q1);    q2=quatern(q2);
q.c=q1.c-q2.c;     q.v=q1.v-q2.v;
q=class(q,'quatern');
```

```
function q=mtimes(q1,q2)
% Процедура обчислення ДОБУТКУ двох кватерніонів
% тобто розрахунку Кватерніону (q)
% за заданими значеннями двох кватерніонів (q1 і q2)
% послідовних поворотів

% Лазарев Ю.Ф      4-12-2001
q1=quatern(q1);    q10=q1.c;    q1v=q1.v;
Q1v=[0 -q1v;q1v.' vect2ksm(q1v)];  Q1=q10*eye(4)+Q1v;
q2=quatern(q2);    q2=[q2.c q2.v];    q=(Q1*q2.'.');
q=quatern(q);
```

```
function q=mrdivide(q1,q2)
% Арифметична операція (/) ПРАВОГО ДІЛЕННЯ двох кватерніонів
% тобто розрахунку Кватерніону q=q1/q2
% Лазарев Ю.Ф      4-12-2001
q1=quatern(q1);    q2=quatern(q2);    q=q1*inv(q2);
```

```
function q=mldivide(q1,q2)
% Арифметична операція (\) ЛІВОГО ДІЛЕННЯ двох кватерніонів
% тобто розрахунку Кватерніону q=q1\q2
% Лазарев Ю.Ф      4-12-2001
q1=quatern(q1);    q2=quatern(q2);    q=inv(q1)*q2;
```

```
function q=ctranspose(q1)
% Процедура обчислення кватерніону, СПРЯЖЕНОГО до заданого
% Лазарев Ю.Ф      4-12-2001
q1=quatern(q1);    q.c=q1.c;    q.v=-q1.v;
q=class(q,'quatern');
```

```
function q=transpose(q1)
% Арифметична операція (.' ) перетворення кватерніону
% у матрицю-стовпець
% Лазарев Ю.Ф      4-12-2001
q1=quatern(q1);    % перетворення у кватерніон
q=[q1.c;q1.v.'];
```

```

function q=inv(q1)
% Арифметична операція ОБЕРНЕННЯ кватерніону
% тобто розрахунку Кватерніону q=invers(q1)
% Лазарев Ю.Ф.      5-07-2003
q1=quatern(q1);      % перетворення у кватерніон
mq=(q1.c)^2+q1.v*(q1.v)'; % квадрат модул. кватерніону
q.c=q1.c/mq; % скалярна частина зворотного кватерніону
q.v=-q1.v/mq; % векторна частина зворотного кватерніону
q=class(q,'quatern'); % перетворення у кватерніон

```

```

function [Norm,Arg,Ncos,q]=qw2norm(qw)
% Процедура QW2Norm формує нормалізовану форму подання кватерніону qw
%      1) Norm - норму (модуль, довжину) кватерніону,
%      2) Arg - аргумент (кут повороту Ейлера),
%      3) Ncos - вектор напрямн[ косинусів осі Ейлерового повороту
%      4) q - довжину векторної частини кватерніону
% за заданим кватерніоном повороту 'qw'
% Лазарев Ю.Ф. 5-07-2003 р.
qw=quatern(qw); % перетворення у кватерніон
q=sqrt(qw.v*(qw.v)'); % довжина вектора кватерніона
Norm=sqrt(q^2+(qw.c)^2); % модуль кватерніона
qw2=qw/Norm; % "Нормалізація" кватерніону
Ca=qw2.c; Sa=q/Norm; Arg=atan2(Sa,Ca); % Аргумент кватерніона
Ncos=qw.v/q;

```

```

function y= quat2ug(qw,kv)
% QUAT2UG перетворює кватерніон у кути Ейлера
% Звернення y = quat2ug(qw,kv)
% дозволяє обчислити вектор "y", що складається зі
% значень y(1),y(2),y(3) кутів послідовних поворотів
% навколо координатних осей з номерами,
% що вказані у векторі "kv"
% за заданим кватерніоном "qw" повороту
% Процедура спирається на функції QUAT2MNK і MNK2UG
% Лазарев Ю. Ф.      5-07-2003
z= quat2mnk(qw);
y= mnk2ug(z,kv); % Кінець процедури "quat2ug"

```

```

function y= quat2mnk(x1)
% QUAT2MNK перетворює кватерніон у матрицю напрямних косинусів
% Звернення:
%      y = quat2ug(x)
% дозволяє обчислити матрицю напрямних косинусів "y"
% за заданим кватерніоном "x"
%      Лазарев Ю.Ф.      5-12-2001
x1=quatern(x1); [v,c]=double(x1); x=[c,v];
y(1,1) = x(1)^2 + x(2)^2 - x(3)^2 - x(4)^2;
y(2,1) = 2 * (x(1)*x(4) + x(2)*x(3));
y(3,1) = 2 * (x(2)*x(4) - x(1)*x(3));
y(1,2) = 2 * (x(2)*x(3) - x(1)*x(4));
y(2,2) = x(1)^2 + x(3)^2 - x(2)^2 - x(4)^2;
y(3,2) = 2 * (x(1)*x(2) + x(3)*x(4));
y(1,3) = 2 * (x(1)*x(3) + x(2)*x(4));
y(2,3) = 2 * (x(3)*x(4) - x(1)*x(2));
y(3,3) = x(1)^2 + x(4)^2 - x(2)^2 - x(3)^2; % Кінець процедури "quat2mnk"

```

```

%%%%%%%%%%%%%%
% Contents.m
%
% Автор Лазарєв Ю.Ф. 5-07-2003
%
% vect2ksm(x) - перетворює трикомпонентний вектор у кососиметричну матрицю (3*3)
% ksm2vect(x) - перетворює кососиметричну матрицю (3*3) у трикомпонентний вектор
% ug2mnpk(x,k) - процедура знаходження Матриці Напрямних Косинусів за кутами Єйлера
% mnpk2ug(x,kv) - перетворює матрицю напрямних косинусів у кути Ейлера
% mnpke(x,k) - Функція формування Матриці Напрямних Косинусів
% за заданим значення куту повороту (x) та номера (k)
% осі, навколо якої здійснюється поворот
%%%%%%%%%%%%%%

```

```

function y = vect2ksm(x)
% vect2ksm перетворює трикомпонентний вектор у КС-матрицю (3*3)
% y = vect2ksm(x) обчислює кососиметричну матрицю "y" такого вигляду
%      [0 -x3 x2
%      y = x3 0 -x1
%      -x2 x1 0]
% за заданим трикомпонентним вектором-рядком "x"
%      x =[x1,x2,x3]
% Ю.Ф.Лазарєв, 8 грудня 1999р.
n = length(x);
if n==3
    y = zeros(3,3);
    y(1,3) = x(2); y(3,1) = - y(1,3);
    y(2,1) = x(3); y(1,2) = - y(2,1);
    y(3,2) = x(1); y(2,3) = - y(3,2);
else
    disp('Невірна розмірність вектору');
end

```

```

function y = ksm2vect(x)
% ksm2vect перетворює КС матрицю (3*3) у трикомпонентний вектор
%
% y = ksm2vect(x) обчислює вектор-рядок "y" за заданою
% кососиметричною матрицею "x"
% Ю.Ф.Лазарєв, 22 вересня 1998р.
% Останні змінювання 5-12-2001
y(1) = x(3,2); y(2) = x(1,3); y(3) = x(2,1);

```

```

function y = ug2mnpk(x,k)
% Процедура знаходження Матриці Напрямних Косинусів за кутами Єйлера
%      y = ug2mnpk(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 'mnpke'
% (с) Лазарєв Ю.Ф. 18-07-1998р.
y = mnpke(x(1),k(1))*mnpke(x(2),k(2))*mnpke(x(3),k(3));

```

```

function y= mnpk2ug(x,kv)

```

```

% MNK2UG перетворює матрицю напрямних косинусів у кути Ейлера
% Звернення:      у = mnk2ug(x,kv)
% дозволяє обчислити вектор "у", що складається зі
% значень у(1),у(2),у(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 вісями початкової та кінцевої
% систем координат
%      x(i,j) = cos(lp,jk),
%      де lp - позначення осі з номером "i" початкової,
%      а jk - осі з номером "j" кінцевої систем координат.
% Примітка:
%      kv(1) не повинно дорівнювати kv(2), а kv(2) не повинно
%      дорівнювати kv(3)
%      Ю.Ф.Лазарєв,      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      % Кінець процедури "mnk2ug"

```

```

function y=mnke(x,k)
% Функція формування Матриці Напрямних Косинусів (у)
% за заданим значенням куту повороту (x) та номера (k)
% осі, навколо якої здійснюється поворот
% (с) Лазарєв Ю.Ф.      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

```

Абетковий покажчик

А	
Алгоритм двокроковий	27
Алгоритм інерціальної навігаційної системи	18
Алгоритм однокроковий	27
Алгоритм чотирикроковий	27
Алгоритми другого порядку	107
Алгоритми інтегрування рівнянь орієнтації у кватерніонах методом Пікара	103
Алгоритми Рунге-Кутта	14
Алгоритми третього порядку	107
Алгоритми четвертого порядку	108

Б	
Безплатформова інерціальна навігаційна система . 5, 8, 46	
Безплатформова система орієнтації	5, 39, 46
БІНС	5, 8, 13, 14, 16, 17
БСО	5, 39, 47

В	
Вектор Ейлера	42
Вектор кінцевого повороту	42
Вектор орієнтації	26, 42
Векторне рівняння орієнтації	46
ВОГ	11
Волоконно-оптичний гіроскоп	11

Г	
Гіростабілізована платформа	7
Гіростабілізована платформа ІНС	8
ГСП	7, 8

Д	
Динамічно настроєний гіроскоп	9
ДНГ	9, 10
Дрейф	9, 11
Друге наближення	26, 77, 104

З	
Земна система відліку	16
ЗСВ	16, 18

І	
Інерціальна навігаційна система	7
Інерціальна система відліку	15
ІНС	7, 8, 9, 11
ІСВ	15, 18

К	
Кватерніони	26
Кватерніонне рівняння	46
Кватерніонне рівняння орієнтації	53
Кільцевий лазерний гіроскоп	10
Кінематичне рівняння вектора орієнтації	42

Кінематичне рівняння для параметрів Родріга-Гамільтона	42
Кінематичне рівняння Ейлера	54
Кінематичне рівняння Пуассона	25
Кінематичне рівняння у кватерніонах	63
Кінематичні рівняння для напрямних косинусів	43
Кінематичні рівняння Ейлера	40, 46, 47, 53
Кінематичні рівняння у кутах Ейлера	43
КЛГ	10, 11
Координати вектора кінцевого повороту	41
Координати вектора орієнтації	41

Л	
Лазерна інерціальна навігаційна система	10, 11

М	
Матриця напрямних косинусів 14, 16, 18, 20, 21, 25, 30, 46	
Матричне рівняння орієнтації Пуассона	46
Матричне рівняння Пуассона	53, 60, 76
Метод Пікара другого порядку	80
Метод Пікара третього порядку	80
Метод послідовних наближень	26
Метод послідовних наближень Пікара	76
Методи багатокрокові	45
Методи другого порядку	44
Методи однокрокові	45
Методи першого порядку	44
Методи прогнозу-корекції	45, 54, 63
Методи третього порядку	44
Методи четвертого порядку	44
МНК	16
Моделювальна схема алгоритму	17
Моделювання алгоритмів	49
Модель прискорення	20

Н	
Навігаційна система координат	16
НСВ	16, 19
Нульове наближення	26, 77

О	
Об'єктова система відліку	15
ОСВ	15, 20
Основне рівняння інерціальної навігації	21

П	
Параметри Кейлі-Клейна	41
Параметри Родріга-Гамільтона	26, 41, 42, 43
Період опитування датчиків	17
Перше наближення	26, 77, 104
Побудова наближених розв'язків	77
Поліноміальне інтерполювання кутової швидкості	78
Поплавцевий гіроскоп	9
Похибки алгоритмів методу Пікара для кватерніонного рівняння орієнтації	116
Похибки алгоритмів прогнозу-корекції	74
Похибки методів прогнозу-корекції	66
Похибки орієнтації	30, 35

Похибки чисельного інтегрування матричного рівняння Пуассона алгоритмами методу Пікара	88
Похибки чисельного інтегрування рівняння Пуассона при вимірюванні приростів псевдокоординат	95
Похибки чисельного інтегрування рівняння Пуассона при вимірюванні проєкцій кутової швидкості основи	95
Приріст куту повороту	17
Прирости квазікоординат	76
Програмне моделювання	49

	Р	
Рівняння Пуассона		46

	Т	
Третє наближення		27, 77, 104

	Ч	
Четверте наближення		78, 104