

ДОДАТОК А

Апробація результатів наукових досліджень

УДК 004.86:621.396.9 DOI: <https://doi.org/10.30837/IYE.ASCTREDB.2024.011>

ПОХИБКИ РУХУ МОБІЛЬНОЇ ПЛАТФОРМИ ПЕРВИННОЇ ТА ВТОРИННОЇ ОБРОБКИ

Гонтаренко І.О.

Науковий керівник – к. т. н., проф. Новоселов С.П.

Харківський національний університет радіоелектроніки, каф. КІТАР,
м. Харків, Україна

e-mail: igor.hontarenko@nure.ua

This work is devoted to the analysis and determination of errors and their nature when moving a mobile platform in the working areas of production.

Навігаційно-часовими визначеннями (НЧВ) формування оцінок координат, складових швидкості і прискорення мобільного робота та поточного часу в результаті обробки радіосигналів, прийнятих від навігаційних супутників. Зазвичай точність НЧВ визначається двома типами похибок: похибками, що виникають на етапі первинної обробки, і похибками, що відповідають етапу вторинної обробки [1].

При первинній обробці формуються оцінки дальностей (псевдодальностей), швидкостей (псевдошвидкостей) та прискорень (псевдоприскорень), то похибками для даного етапу є похибки визначення псевдодальності, псевдошвидкості та псевдоприскорення. Складові похибки, що виникають на етапі первинної обробки, при визначенні псевдодальності дальномірним методом розглянуті в [2], а складові похибки етапу первинної обробки визначення псевдо швидкості.

На етапі вторинної обробки оцінки псевдодальності, псевдошвидкості та псевдоприскорення перераховуються в оцінки координат мобільного робота, тому похибки цього етапу визначаються факторами, які впливають на ефективність перерахунку.

Похибку етапу вторинної обробки можна розділити на такі складові [2]:

- похибки, що виникають внаслідок неповного врахування умов поширення радіохвиль (вплив тропосфери та іоносфери на запізнення сигналу, релятивістських та гравітаційних ефектів, а також багатопроменевого поширення сигналу);
- похибки бортової апаратури та апаратури мобільного робота;
- похибки, що вносяться на етапі розв'язання навігаційної задачі;
- похибки ефемеридного забезпечення.

Тому аналіз та оцінка складових похибок, що виникає на етапі вторинної обробки НЧВ є досить актуальною та значущою проблемою, що виникає при використанні технологій слідування траєкторії руху мобільної платформи на основі даних від акселерометрів та гіроскопів.

Похибки, що виникають на трасі поширення сигналу, є найменш передбачуваними і тому можуть суттєво впливати на точність визначення координат. Рефракція радіохвиль, яка проявляється у викривленні шляху поширення, також вносить додаткові затримки сигналу. На поширення радіохвиль впливають тропосфера, розташована біля поверхні Землі

Для зменшення впливу атмосферних похибок рекомендується виключати з розрахунків результати вимірювань за сигналами НС, кут місця. Однак значення тропосферної похибки залежить від факторів, які можна досить точно визначити: кліматичних факторів та кута місця навігаційного супутника (кута підняття над горизонтом).

Похибки апаратури мобільного робота це похибки прийомоіндикатора е схеми стеження за затримкою оригнальної та носійної сигналу, при цьому розрізняють шумові й динамічні похибки.

Висновки. Можна зробити висновки, що наразі системи позиціонування мобільних платформ в виробничих приміщення в переважній більшості, знаходяться на стадії доопрацювання і робота в цьому напрямку є затребуваною. Проте вартість їх впровадження та витрати на експлуатацію в поєднанні з недостатньою точністю часто переважають отримані вигоди, що вказує на необхідність їх доопрацювання.

Похибки, що вносяться на етапі розв'язання навігаційної задачі оцінки псевдодальностей, псевдошвидкостей і псевдоприскорень, отримані на етапі первинної обробки за чотирма або більше видимими НС, використовуються для розрахунку оцінок координат та векторів швидкості і прискорення МР, що є сутністю вторинної обробки інформації в прийнятному пристрої (або розв'язання навігаційної задачі). При проведенні необхідних розрахунків використовується інформація про координати і складові вектора швидкості (ефемериди) кожного НС, яка виділяється з навігаційного повідомлення. Точність цієї ефемеридної інформації безпосередньо впливає на точність НЧВ.

Список використаних джерел:

1. Nevliudov I., Novoselov S., Sychova O., Tesliuk S. Development of the Architecture of the Base Platform Agricultural Robot for Determining the Trajectory Using the Method of Visual Odometry, 2021 *IEEE XVIIIth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)*, Polyana (Zakarpattya), 2021. P. 64–68. doi: 10.1109/MEMSTECH53091.2021.9468008.
2. Рудик А. В. Використання стелс-технологій в мобільних робототехнічних комплексах та методи виявлення малопомітних об'єктів // Вимірювальна та обчислювальна техніка в технологічних процесах. 2016. № 2. С. 146-150.

АЛФАВІТНИЙ СПИСОК

А		З	
Александров І. О.	85	Зарубін І. С.	19
Алістратов О. М.	107	Здорик Н. В.	154
Апатов Ф. В.	88	Зіад К.	70
Антоненко Ю. О.	136	Зінченко Д. О.	22
		Зуєва А. Д.	72
Б		К	
Багаєв Д. О.	111	Карпович Б. О.	98
Басюк В.С.	139	Ключник С. С.	24
Білоконь М.А.	5	Коваленко І. С.	26
Бін Д.	7	Кожевникова В. Г.	74
Бойко А. Ю.	59	Кравченко Р. С.	156
Брехов Д.О.	141	Крещук М. О.	100
Бураковська М. С.	70	Кудря Т. К.	77
	90		
В		Л	
Вжесневський М.О.	9	Леонова А. О.	102
Внуков Т. С.	144	Літвін В. О.	159
Вяліна А. В.	93		
Г		М	
Гапон Н. Я.	151	Мельник А. О.	151
Гонтаренко І. О.	61, 63	Мешков А.Ю.	105
Горева М. М.	11	Мігаль С. Д.	107
Гродецький А. Ю.	146	Морозова К. О.	29
Гуржій С. В.	149		162
Гусак О. А.	151		
	65, 80	Н	
Д		Назаренко С. В.	32
Дзюба С. С.	13	Небрат В. В.	65, 80
Дмитренко Д. М.	67	Носик Д. О.	34
Долгошея І. Д.	16		
Ж		П	
Жезлова А. С.	17	Павленко Ю. В.	110
	95	Пара І. І.	36
		Посвалюк М. В.	80
		Проценко А.А.	39

ДОДАТОК Б

Вихідний код програми

Вихідний код програми для Arduino:

```
#include "ModbusRtu.h"
#include <Wire.h>
#include "MPU6050.h"
#include "Kalman.h"

MPU6050 mpu;
#define ID 10
#define TXEN 4

#define EN 5
#define DIR 3
#define CLK 2

uint16_t cur_step = 0;
uint16_t target_step = 0;

// data array for modbus network sharing
//au16data[0] -> Start | Stop route
//au16data[1] -> calc angle
//au16data[2] -> calc dir
//au16data[3] -> Step fo step motor
//au16data[4] -> Kalman angle
uint16_t au16data[5] = {1, 2, 3, 0, 0};

/**
 * Modbus object declaration
 * u8id : node id = 0 for master, = 1..247 for slave
 * port : serial port
```

```
* u8txenpin : 0 for RS-232 and USB-FTDI
*           or any pin number > 1 for RS-485
*/
Modbus slave(ID, Serial, TXEN); // this is slave @10 and RS-485

// Timers
unsigned long timer = 0;
float timeStep = 0.01;

// Pitch, Roll and Yaw values
float pitch = 0;
float roll = 0;
float yaw = 0;

long time_zero = 0;
long cur_time_zero = 0;
int incomingByte = 0;
bool isRotate = false;
uint16_t angle = 0;
float angle_start = 0;
uint16_t dir = 0;

int metallVal = 0;

// Ініціалізація фільтрів Калмана для акселерометра та гіроскопа
Kalman kalmanX; // Для осі X
Kalman kalmanY; // Для осі Y
int Kalman_angle;

// Змінні для зберігання значень
```

```
float ax, ay, az; // Прискорення з акселерометра
float gx, gy, gz; // Кутова швидкість з гіроскопа
float angleX = 0; // Кут по осі X
float angleY = 0; // Кут по осі Y

void setup()
{
  Serial.begin(115200);
  slave.start();

  pinMode(EN, OUTPUT);
  pinMode(DIR, OUTPUT);
  pinMode(CLK, OUTPUT);

  digitalWrite(EN, HIGH);
  digitalWrite(DIR, HIGH);
  digitalWrite(CLK, LOW);

  // Initialize MPU6050
  while(!mpu.begin(MPU6050_SCALE_2000DPS,
MPU6050_RANGE_2G))
  {
    Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
    delay(500);
  }

  // Calibrate gyroscope. The calibration must be at rest.
  // If you don't want calibrate, comment this line.
  mpu.calibrateGyro();
```

```
// Set threshold sensivty. Default 3.
// If you don't want use threshold, comment this line or set 0.
mpu.setThreshold(0);
//mpu.setScale(MPU6050_SCALE_250DPS);

time_zero = millis();
isRotate = false;
Serial.println("Start sensor!");
}

void loop()
{
  slave.poll( au16data, 4 );
  timer = millis();

  // Read normalized values
  Vector norm_G = mpu.readNormalizeGyro();
  Vector norm_A = mpu.readNormalizeAccel();

  // Calculate Pitch, Roll and Yaw
  pitch = pitch + norm_G.YAxis * timeStep;
  roll = roll + norm_G.XAxis * timeStep;
  yaw = yaw + norm_G.ZAxis * timeStep;

  ax = norm_A.ZAxis * timeStep;

  //Kalman filter
  float kalAngleZ = kalmanX.getAngle(ax, yaw, millis());
  Kalman_angle = (int)kalAngleZ;
```

```
if (isRotate == true)
{
    if (angle_start < yaw) {
        angle = (int)(yaw - angle_start) + 1;
        dir = 0;
    }
    else if (angle_start > yaw) {
        angle = (int)(angle_start - yaw) + 1;
        dir = 1;
    }
}

//step
target_step = au16data[3];
if (cur_step < target_step)
{
    while (cur_step < target_step)
    {
        digitalWrite(EN, LOW);
        digitalWrite(DIR, LOW);
        digitalWrite(CLK, HIGH);
        delayMicroseconds(550);
        digitalWrite(CLK, LOW);
        delayMicroseconds(550);
        cur_step++;
    }
}
else if (cur_step > target_step)
{
    while (cur_step > target_step)
```

```
{
    digitalWrite(EN, LOW);
    digitalWrite(DIR, HIGH);
    digitalWrite(CLK, HIGH);
    delayMicroseconds(550);
    digitalWrite(CLK, LOW);
    delayMicroseconds(550);
    cur_step--;
}
}
else {digitalWrite(EN, HIGH);}

// Wait to full timeStep period
while (((timeStep*1000) - (millis() - timer)) > 0){
    io_poll();
}
}

void io_poll() {

    //1 - start route
    //2 - stop route
    if (au16data[0] == 1)
    {
        isRotate = true;
        angle_start = yaw;
        angle = 0;
        dir = 0;
        au16data[0] = 0;
    }
}
```

```
}  
else if (au16data[0] == 2)  
{  
    isRotate = false;  
    mpu.calibrateGyro();  
    angle_start = yaw;  
    angle = 0;  
    dir = 0;  
    au16data[0] = 0;  
}  
  
au16data[1] = angle;  
au16data[2] = dir;  
au16data[4] = Kalman_angle;  
}
```

