

значень величини вимірюваного електричного опору. Розроблений метод дає змогу роз'язати задачу вимірювання малих величин електричного опору в MEMS.

1. Kruglick J. J. *EFAB Technology and Applications* / J. J. Kruglick, A. Cohen, C. Bang // *MEMS: Design and Fabrication* / [Mohamed Gad—el—Hak, ed.]. – 2nd ed. – Boca Raton: CRC Press, 2006. – 664 p.
2. Колпаков Ф. Ф. *Микроэлектромеханические устройства в радиотехнике и системах телекоммуникаций* : [учеб. пособие] / Ф. Ф. Колпаков, Н. Г. Борзяк, В. И. Кортунов. – Харьков : НАУ ХАИ, 2006. – 82 с.
3. *Minhang Bao Analysis and Design Principles of MEMS Devices*, – 1st edition: Elsevier Science, 2005 – 328 p.
4. James J. Allen *Micro Electro Mechanical System Design*, – 1st edition: CRC Press, 2005 – 496 p.
5. Marc J. Madou *Fundamentals of Microfabrication: The Science of Miniaturization*, – 2nd edition: CRC Press, 2002 – 752 p.
6. Пукач А., Іванців Р., Теслюк В. *Методи та схеми вимірювання значення невідомого опору електричного кола*. *Технічні вісті: Науково-публіцистичний часопис "Львівська політехніка"*. Львів: Видавництво НУ"Львівська політехніка". – 2010. – С. 160–163.
7. Джеймс Питерсон *Теория сетей Петри и моделирование систем: Пер. с англ.* – М.: Мир, 1984. – 264 с.
8. Котов В.Е. *Сети Петри*. – М.: Наука. Главная редакция физико-математической литературы, 1984. – 160 с.
9. Васильев В.В., Кузьмук В.В. *Сети Петри, параллельные алгоритмы и модели мультипроцессорных систем*. К.: Наук. думка, 1990. – 216 с.
10. James L. Peterson *A Note on Colored Petri Nets*, *Information Processing Letters*, Volume 11, Number 1, (August 1980), pages 40–43.
11. Kurt Jensen, Lars M. Kristensen. *Coloured Petri Nets: modelling and validation of concurrent systems: 1st edition* – Springer, 2009. – 395 p.

УДК 681.5 , 62-529, 62-58

А. Павельчак¹, В. Самотий^{1,2}, Р. Василів

¹ Національний університет "Львівська політехніка",
кафедра комп'ютеризованих систем автоматики

² Politechnika Krakowska, Katedra Automatyki
i Technik Informatycznych, Polska

АЛГОРИТМ КЕРУВАННЯ ДВОХОСЬОВОЮ ПЛАТФОРМОЮ

© Павельчак А., Самотий В., Василів Р., 2012

Запропоновано алгоритм для керування стабілізацією положення двохосевої платформи з використанням біполярних крокових моторів, акселерометра та керуючого мікроконтролера.

Ключові слова: кроковий двигун, акселерометр, мікроконтролер.

In this paper, the authors have proposed the algorithm to control the stabilization position 2-axial platform using bipolar stepper motors, accelerometer and microcontroller.

Key words: stepper motor, accelerometer, microcontroller.

Вступ

Завдання автоматичної стабілізації положення рухомої платформи у наперед заданій площині є давно відомим і в різні способи реалізованим. Найпоширеніший варіант – це системи з використанням гіроскопів. Потреба у стабілізації об'єктів та окремих їхніх вузлів виникає як в інерційних навігаційних системах, що встановлюються на кораблях та літальних апаратах, габаритному автомобільному транспорті, бурових платформах, так і в компактній побутовій техніці. Залежно від сфери застосування, використовуються, відповідно, різні пристрої, що виконують функції гіроскопа. Тому кожне завдання стабілізації потребує врахування своїх особливостей.

Сьогодні великої популярності набувають мікромеханічні пристрої, побудовані з використанням мікроелектромеханічної системи (MEMS – Micro Electromechanical Systems), серед них і мікромеханічні акселерометри. З швидким розвитком технологій виготовлення MEMS виникла і потреба в розробленні нових програмних інструментів для проектування та моделювання систем на основі таких пристроїв. У цій статті пропонуємо програмний алгоритм стабілізації положення двохосової платформи на основі даних, що отримуються з мікромеханічного акселерометра.

Аналіз публікацій

Акселерометри [1, 3, 5, 6] можна використовувати у будь-якому пристрої, робота якого пов'язана з переміщенням, нахилом, вібрацією. Наведемо деякі типові застосування акселерометрів. Автомобілебудування. Давачі використовуються для систем безпеки (подушки безпеки, протиугінні системи), короточасної навігації, діагностики автомобіля [2]. Авіоніка. Давачі прискорення широко використовуються у системах керування літальними апаратами, а також в космічних системах. Вимірюються вібрації і квазістатичні прискорення [4]. Побутова техніка. Використовуються для зменшення вібрацій (пральної машини тощо). Системи захисту від падіння. Типовий приклад – захист жорсткого диска в ноутбуках або інших мобільних пристроях. Під час падіння відбувається парковка магнітної головки жорсткого диска. Медицина і спорт. Акселерометри застосовуються в системах діагностики, тренажерах, крокомірах тощо. Пристрої введення інформації. Використовуються для комп'ютерних ігор, мобільних телефонів тощо.

Постановка задачі

На основі запропонованої системи (рис. 1) необхідно розробити програмний алгоритм, який підтримуватиме стабільне положення двохосової платформи для заданих кутів нахилу, незалежно від зміни нахилу основи установки. Кути нахилу при цьому можуть змінюватися програмно. Також передбачається, що осі платформи відхилятимуться лише в межах $\pm 90^\circ$.

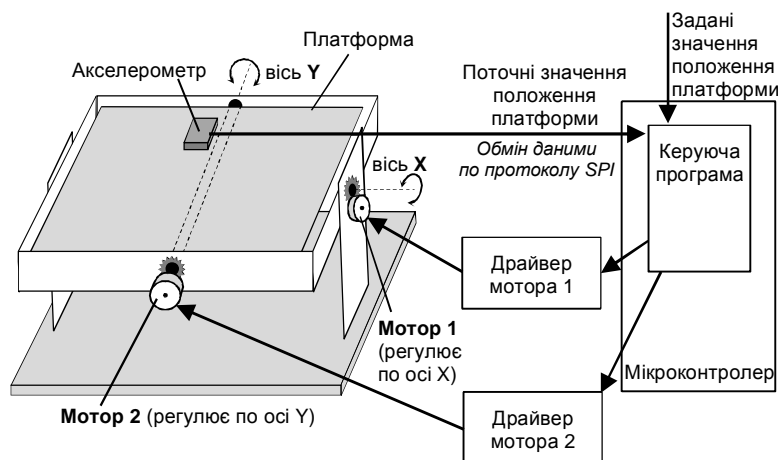


Рис. 1. Структурна схема системи керування для стабілізації положення платформи

Акселерометр генерує кути відхилення відносно горизонту по двохосях X та Y як додатні та від'ємні значення, залежно від вибраної чутливості давача. За наявності заводської похибки калібрування давача відносно горизонтального положення алгоритм повинен передбачати можливість корекції.

Драйвер біполярного крокового мотора повинен підтримувати базові режими роботи моторів: однофазний, двофазний, півкроковий, розгону, гальмування. У режимі гальмування необхідно зменшувати робочий струм обмоток за допомогою ШІМ-регулювання.

На рис. 2 показано реалізацію драйвера. Н-мости виконані на польових транзисторах IRF7105 з N- та P-каналами в одному корпусі та з керуванням логічним рівнем. Для керування

польовими транзисторами як їхні драйвери використані біполярні транзисторні ключі BC337. Ці ключі заодно виконують і функцію захисту від збоїв у програмі мікроконтролера, оскільки забезпечують однакову логіку для N- та P-каналу польової транзисторної зборки. Керування на драйвер крокового мотора подається з мікроконтролера на входи Ain, Bin, Cin та Din. Польові зборки захищені від викидів напруги при відімкненні обмоток двигуна діодами Шоткі. Апаратне ШІМ-регулювання струму реалізоване на логіці І та АБО (мікросхеми 4081, 4071). На вивід PWM драйвера подається ШІМ-сигнал з виводу таймера мікроконтролера, а на вивід Men – дозвіл на активацію ШІМ-регулювання струму обмоток мотора. Для дроблення кута нахилу платформи мотори працюють через редуктори.

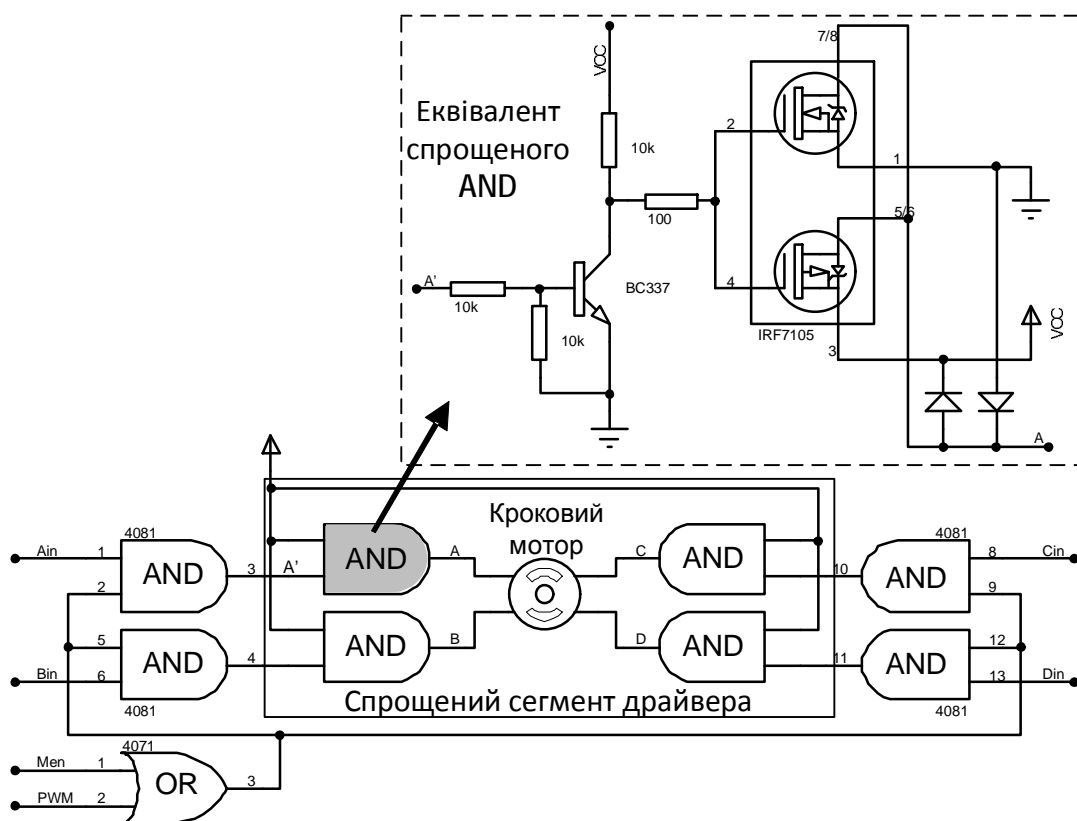


Рис. 2. Реалізація драйвера біполярного крокового мотора

На рис. 3 показані режими, які повинен відпрацювати розроблюваний алгоритм. Базовий режим (А) керування роботою мотора передбачає, що ділянки розгону та гальмування не перетинаються, тобто мотор спершу розганяється до встановленої швидкості, обертається деякий час з номінальною швидкістю, а потім за досягнення кінцевого маршруту сповільнюється. Початок ділянки гальмування визначається згідно з наперед вибраним значенням акселерометра.

Однак базовий режим не враховує деякі нюанси. У разі старту руху поблизу InhibitionZone мотор ще не встиг розігнатися, а вже потрібно гальмувати (рис. 3, Б). Може трапитися, що мотор більшу частину ділянки InhibitionZone пройде на мінімальній швидкості. Або, якщо мотор стартуватиме у самій ділянці, то, відповідно, він буде обертатися до кінцевого пункту на мінімальній швидкості. А це знижує швидкісні характеристики та динаміку цілої системи.

Інший варіант поведінки платформи – це різка зміна координат з + на -, і навпаки (рис. 3, В). Наприклад, платформа перебувала у стані спокою і отримала показ +20 з акселерометра. Згідно з алгоритмом мотор починає поступово прискорюватися до номінальної швидкості. І тут стрибком показ акселерометра змінюється на від’ємне значення, наприклад -23. Наш алгоритм попросту змінить напрям руху, хоча, логічно, він мав би поступово пригальмувати, а потім почати прискорюватися у протилежний бік.

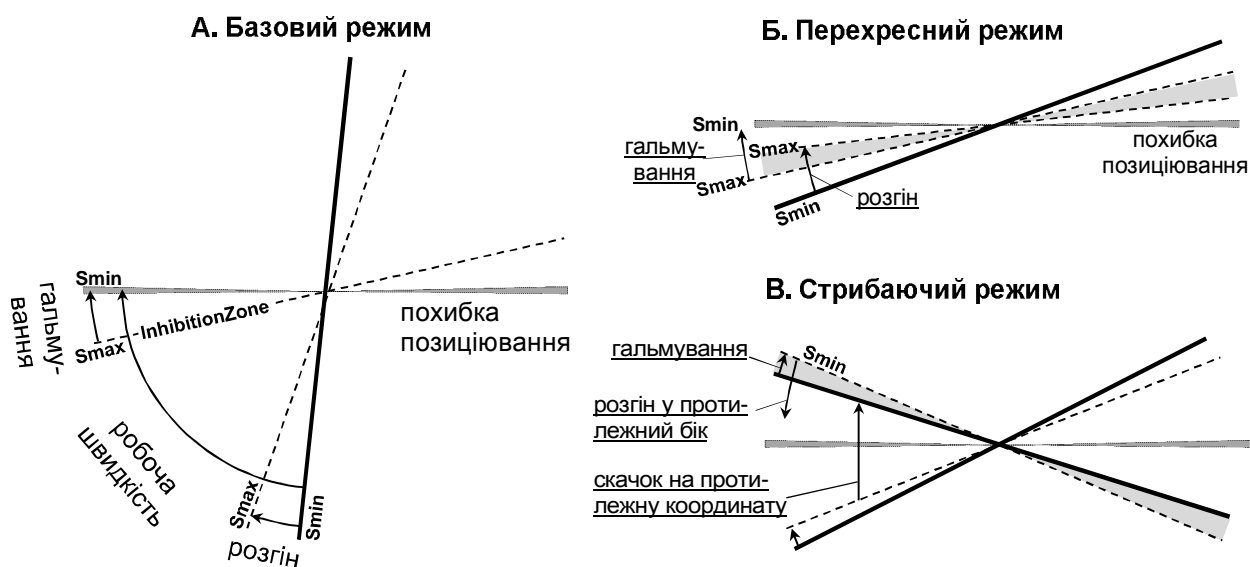


Рис. 3. Режими роботи крокового мотора

Реалізація алгоритму

Режими роботи драйвера крокового мотора (однофазний, двофазний, півкроковий) реалізуємо за допомогою масиву даних, представлених у двійковому вигляді

SM[3][8] = { {0001₂, 0100₂, 0010₂, 1000₂, 0001₂, 0100₂, 0010₂, 1000₂}, /1-фазний/
 {0101₂, 0110₂, 1010₂, 1001₂, 0101₂, 0110₂, 1010₂, 1001₂}, /2-фазний/
 {0001₂, 0101₂, 0100₂, 0110₂, 0010₂, 1010₂, 1000₂, 1001₂}}; /півкроковий/

Логіку розгону/гальмування реалізуємо на затримці за рахунок пропусків тактів таймера (сам алгоритм керування має бути реалізований у підпрограмі переривання таймера мікроконтролера). Тобто, спершу пропускаємо n кроків, потім $n-1$, $n-2$ і до 0. У алгоритмі ця логіка буде реалізована на змінних Speed та N. У разі гальмування Speed щокроку зменшуватиметься, а у разі розгону – збільшуватиметься до значення Smax. Для режиму гальмування необхідно також вказувати межі його зони (InhibitionZone), що визначатиметься з показів акселерометра. Обчислюють значення InhibitionZone за такою формулою:

$$\text{InhibitionZone} = \frac{\text{Accel}_{90^\circ} \times \text{Nstep}_X}{\text{Nstep}_{90^\circ}}, \quad (1)$$

де Accel_{90° – статичні покази акселерометра за відхилення на 90° , Nstep_{90° – кількість кроків, які виконує двигун у разі повороту осі платформи на 90° , Nstep_X – кількість кроків двигуна, які необхідні для зупинки у зоні стабілізації.

Напрямки руху мотором задаються за допомогою елементів переліку
enum {Stop, Forward, Back} DirM, DirMnew;

Алгоритм керування

1. Початкова ініціалізація: Xcor, Ycor, Epos, InhibitionZone, Smax, Smin=1, Speed=Smin, N=Smax, mode, sw=0, PORTC=SM[mode][7], ШІМ включений, K=0, logic=1, DirM=Stop.

2. Отримання значень кутів відхилень X та Y.

3. Xpos = X – Xcor. /корекція положення/

4. **якщо** |Xpos| > Epos, **тоді**: /мотор обертається/

4.1. виключення ШІМ-регулювання струму.

4.2. **якщо** N=Speed, **тоді**: /виконуємо крок двигуном/

4.2.1. N=Smax. /поч. знач. для лічильн. пропуску тактів таймера/

4.2.2. **якщо** DirM=Stop, **тоді**:

- 4.2.2.1. **якщо** $X_{pos} > 0$, **тоді:** DirM=Forward; DirMnew=Forward
інакше: DirM=Back; DirMnew= Back
- інакше:**
- 4.2.2.2. **якщо** $X_{pos} > 0$, **тоді:** DirMnew=Forward
інакше: DirMnew= Back
- 4.2.3. **якщо** DirM \neq DirMnew, **тоді:** /зупиняємо мотор гальмуючи/
4.2.3.1. **якщо** Speed $>$ Smin, **тоді:** Speed-- /гальмуємо/
інакше: DirM=Stop
- інакше:**
- 4.2.3.2. **якщо** $|X_{pos}| \leq \text{Inhibition Zone}$, **тоді:** /зона гальмування/
4.2.3.2.1. **якщо** logic=1, **тоді:** /1-й раз у зоні гальмування/
4.2.3.2.1.1. $K = (S_{max} - \text{Speed}) / 2$;
4.2.3.2.1.2. logic=0;
4.2.3.2.2. **якщо** $K < 0$, **тоді:**
4.2.3.2.2.1. K--;
4.2.3.2.2.2. Speed++; /прискорюємо/
інакше:
4.2.3.2.2.3. **якщо** Speed $>$ Smin, **тоді** Speed--
/гальмуємо/
інакше:
4.2.3.2.3. **якщо** Speed $<$ Smax, **тоді** Speed++
/прискорюємо/
4.2.3.2.4. logic=1;
- 4.2.4. **якщо** DirM=Forward, **тоді:** /обертання вперед/
перемикач (sw)
0: PORTC=SM[mode][0]; sw=1
1: PORTC=SM[mode][1]; sw=2
2: PORTC=SM[mode][2]; sw=3
3: PORTC=SM[mode][3]; sw=4
4: PORTC=SM[mode][4]; sw=5
5: PORTC=SM[mode][5]; sw=6
6: PORTC=SM[mode][6]; sw=7
7: PORTC=SM[mode][7]; sw=0
інакше: /обертання назад/
перемикач (sw)
0: PORTC=SM[mode][0]; sw=1
1: PORTC=SM[mode][1]; sw=2
2: PORTC=SM[mode][2]; sw=3
3: PORTC=SM[mode][3]; sw=4
4: PORTC=SM[mode][4]; sw=5
5: PORTC=SM[mode][5]; sw=6
6: PORTC=SM[mode][6]; sw=7
7: PORTC=SM[mode][7]; sw=0
інакше (до п. 4.2): N-- /пропускаємо такт таймера/
інакше (до п. 4): /мотор не рухається/
4.3. Speed=Smin; N=Smax; /скидання до початкових параметрів/
4.4. включення ШІМ-регулювання струму.
4.5. logic=1; DirM=Stop.
5. аналогічна модифікація пп.3-4 для координати Y.
6. при наступному такті таймера алгоритм продовжується з п.2.

У цьому алгоритмі використані змінні мають такий зміст:

Базовий режим. **X** та **Y** – значення кутів відхилень від заданого положення, які зчитуються з акселерометра по SPI-протоку; **Xcor** та **Ycor** – задані відхилення для фіксації платформи у довільному положенні, якщо ж дорівнюють 0, тоді платформа стабілізується в горизонтальній площині; **Xpos** та **Ypos** – відкалібровані величини кутів, що враховують задані значення відхилень; **Epos** – значення похибки показів акселерометра (на рис. 3 ця похибка позиціонування позначена сірою ділянкою); **InhibitionZone** – величина, що задає розмір для зони гальмування; **Smax** та **Smin** – задають межі прискорення (фактично їхня різниця відповідає кількості кроків двигуна, протягом яких має виконатися прискорення); **N** – змінна, що відповідає за прискорення (за допомогою неї реалізується логіка пропуску тактів таймера); **Speed** – визначає поточну швидкість; **mode** – вказує у якому з трьох режимів (однофазний, двофазний, півкроковий) має працювати мотор; **sw** – змінна для реалізації логіки перемикачів фазами двигуна; **SM** – двовимірний масив, у якому розміщені дані для реалізації логіки перемикачів фазами двигуна для вказаних трьох режимів; **PORTC** та **PORTD** – регістри мікроконтролера, що відповідають за вихідні рівні на його портах C та D, та до яких під'єднані драйвери крокових біполярних двигунів (для спрощення алгоритму передбачається, що решта виводів цих портів не задіюються);

Перехресний режим. **logic** – якщо змінна дорівнює 1, то це означає, що ми або ще не зайшли в зону гальмування, або перебуваємо у ній перший раз, якщо дорівнює 0 – ми виконуємо обертання двигуном у зоні гальмування; **K** – змінна, в якій зберігається кількість кроків прискорення, що необхідно здійснити мотору.

Стрибаючий режим. За допомогою змінних **DirM** та **DirMnew** фіксується напрямок руху мотора, як у попередньому такті таймера, так і у поточному. Якщо змінні містять різні значення, то це означає, що трапився різкий стрибок на протилежну за знаком координату, і необхідно плавно змінити напрямок обертання мотора. Тобто, мотор має спершу пригальмувати і зупинитися, а потім, поступово прискорюючись, почати обертатися у протилежний бік.

Для реалізації цього алгоритму на нашій фізичній установці необхідно попередньо визначити значення величин, що потребують початкової ініціалізації.

Тестування алгоритму проводилося за допомогою системи симулювання мікропроцесорних пристроїв Proteus та середовища розробника AVR Studio v.4.19. Програма керування була написана мовою Cі (компілятор AVR Toolchain). Для апробації алгоритму був виготовлений макет установки з такою конфігурацією (рис. 4):

- біполярні крокові мотори M42SP-6NKA 6.3 Ом, 7,5 град. (MITSUMI);
- передатна кількість редукторів: вісь X – 21,44; вісь Y – 4,7;
- драйвера моторів на базі мікросхем польових зборок IRF7105;
- акселерометр LIS302DL з передачею даних по SPI-протоколу;
- керуючий мікроконтролер ATmega32A;
- LCD-дисплей RAYSTAR RC1602A-GGN-CSX.

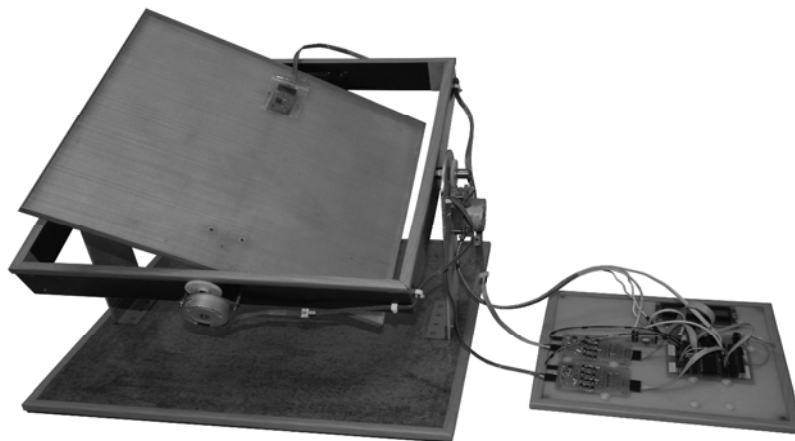


Рис. 4. Макет для натурної апробації алгоритму керування

Усі необхідні файли проектів для тестування алгоритму можна знайти на сайті автора <http://pavelchak.at.ua>.

Висновки

Розроблений програмний алгоритм керування дає можливість виконувати стабілізацію положення двохосової платформи для заданих кутів нахилу та є оптимізований для роботи у разі незначних відхилень та різких стрибків. Також його можна використати у разі певної його модифікації для подібних систем, що використовують крокові двигуни. Необхідно зазначити, що для досягнення високої точності позиціонування платформи потрібно використовувати акселерометри, що мають низькі шуми та велику роздільну здатність на виході (такі моделі доволі дорогі).

1. Волович А., Волович Г. *Интегральные акселерометры // Компоненты и технологии.* – 2002. – № 1. – С. 67–72. 2. Сысоева С. *Автомобильные акселерометры. Части 1-6 // Компоненты и технологии.* – 2005. – № 8–9; – 2006. – № 2–5. 3. Сысоева С. *Ключевые сегменты рынка МЭМС-компонентов. Акселерометры // Компоненты и технологии.* – 2010. – № 3. – С. 20–26. 4. А.Казакевич. *Акселерометры Analog Devices. Устройство, применение и непрерывное обновление // Компоненты и технологии.* – 2007. – № 5. – С. 46–50. 5. William C. Dunn. *Introduction to instrumentation, sensors, and process control.* – ARTECH HOUSE, INC. – 2006. – 331 p. 6. *Modern sensors handbook / edited by Pavel Ripka, Alois Tipek.* – ISTE Ltd. – 2007. – 518 p.