

ПРОГРАМА КЕРУВАННЯ МАНІПУЛЯТОРОМ РОБОТА МАРСОХІДА

професор, к.т.н., Новоселов С.П., старший викладач Сичова О.В.

Харківський національний університет радіоелектроніки,
кафедра комп'ютерно-інтегрованих технологій, автоматики і мехатроніки,
e-mail: sergiy.novoselov@nure.ua, oksana.sychova@nure.ua

Abstract. In the given work developed a program to control the manipulator robot «Mars rover» in space. The control program interacts with the manipulator control module via a serial wired or wireless interface. Commands are sent sequentially. Each command affects the operation of a particular motor depending on its type. The main interface of the program combines all visual components and processes events from active components.

Вступ. Найчастіше в мобільних роботах дослідницького призначення в якості інструменту для контактування з об'єктами використовується рука-маніпулятор. Рука робота є гнучким та універсальним інструментом тому, що кінцевий сегмент маніпулятора може бути застосовано для виконання різноманітних завдань або виробництв. Наприклад, шарнірний зварювальний маніпулятор може бути використаний для точкового зварювання, за допомогою сопла-розпилювача можна забарвлювати різні деталі і вузли, а захват може використовуватися для затиску і встановлення предметів.

Для управління пристроєм, що розробляється, потрібне застосування обчислювального пристрою. Таким пристроєм може служити мікроконтролер. Цей мікроконтролер повинен виконувати наступні функції: приймати сигнали зворотних зав'язків по струму і швидкості двигунів, проводити обчислення регуляторів двигунів, управляти двигунами, видаючи сигнали, що управляють, на драйвери двигунів.

Для керування маніпулятором найчастіше використовуються спеціалізовані портативні пульти керування, або застосовують програмне забезпечення на персональному комп'ютері, коли оператор не може знаходитись в прямому візуальному контакті з роботом.

Метою даного дослідження є рішення задачі візуалізації переміщення ланок маніпулятора робота «Марсохід» в просторі за допомогою програми.

Основна частина. З аналізу об'єкту автоматизації було складено ескіз маніпулятора, яким потрібно керувати (рис. 1). Плече 1 має довжину 380 мм, а плече 2 – 200 мм. В якості серводвигуна 1 використовується кроковий двигун, в якості серводвигуна 2 використовується двигун постійного струму з редуктором. Захватом керує серводвигун з вбудованим контролером керування.

Всі двигуни керуються за допомогою спеціальних модулів – драйверів відповідного типу двигунів. Програма, що розроблюється повинна мати можливість зв'язуватися з головним модулем керування за допомогою будь-якого інтерфейсу та передавати спеціально розроблені для цього команди для включення, або виключення певного двигуна.

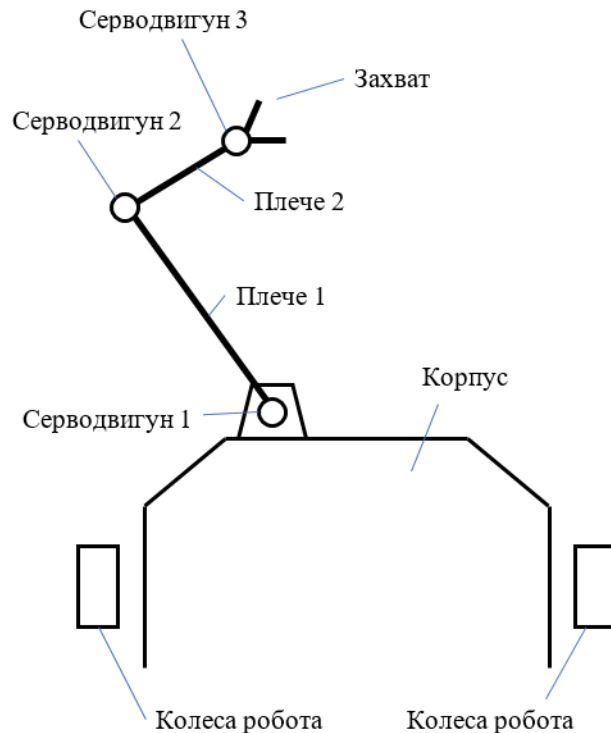


Рисунок 1 – Ескіз маніпулятора робота «Марсохід»

Особливістю даного типу маніпулятора є відсутність реалізованого зворотного зв'язку з модулем керування. Це накладає певні вимоги до програмного модуля керування. Одним з рішень є використання візуального контакту з маніпулятором робота під час керування. Тобто оператор повинен мати можливість за допомогою відеокамери або наочно контролювати процес переміщення маніпулятора.

Іншим варіантом є визначення кута повороту плеча за допомогою часу, що сплинув після отримання команди керування. Якщо знати час, який витрачається на весь шлях, то можна з певною похибкою підрахувати необхідний проміжок часу для зміщення механізму на заданий кут.

Головне вікно програми керування маніпулятором робота «Марсохід» наведено на рис. 2. В інтерфейсі виділяються дві області: область керування та область відображення поточного стану маніпулятора. На рисунку позначено: 1 – елемент керування великим плечем маніпулятора; 2 – кнопка «Вліво» для керування великим плечем маніпулятора; 3 – кнопка «Вправо» для керування великим плечем маніпулятора; 4 – кнопка «Взяти» для керування захватом; 5 – кнопка «Відпустити» для керування захватом; 6 – кнопка «Вліво» для керування маленьким плечем

маніпулятора; 7 – кнопка «Вправо» для керування маленьким плечем маніпулятора; 8 – елемент керування нижнім плечем маніпулятора.

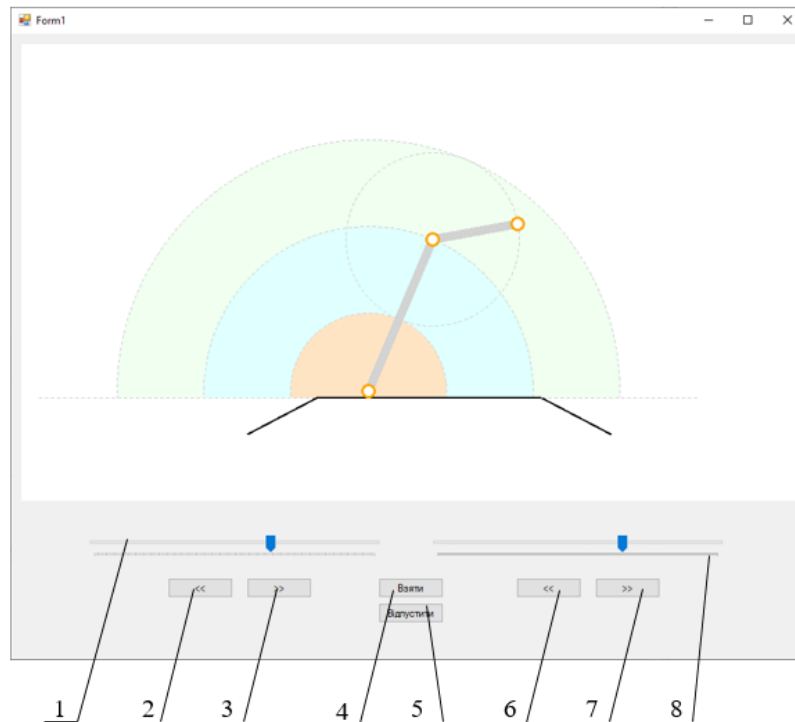


Рисунок 2 – Головна форма програми

В інтерфейсі програми використовуються такі візуальні компоненти, як кнопки та слайдери. Слайдери 1 та 8 призначені для точного позиціонування маніпулятора за допомогою комп'ютерної миші. Слайдер 1 керує великим плечем маніпулятора, слайдер 2 – для керування маленьким плечем. З переміщенням слайдеру відбувається перебудова поточного стану маніпулятора на екрані ПК. Беручи до уваги відсутність енкодерів на двигунах маніпулятора, зображення на екрані не завжди відповідає поточному стану маніпулятора на роботі марсоході. Тому для більш точного керування використовуються кнопки 2-7.

Для обробки стану слайдера використовується обробник події `ValueChanged`. Приклад функції обробника показано нижче:

```
private void trackBar6_ValueChanged(object sender, EventArgs e)
{
    TrackBar tr = (TrackBar)sender;
    angle2 = 360 - tr.Value;
    panel1.Refresh();
}
```

Даний код обчислює кут оберту маленького плеча маніпулятора та присвоює його змінній `angle2`. Аналогічна функція використовується і для

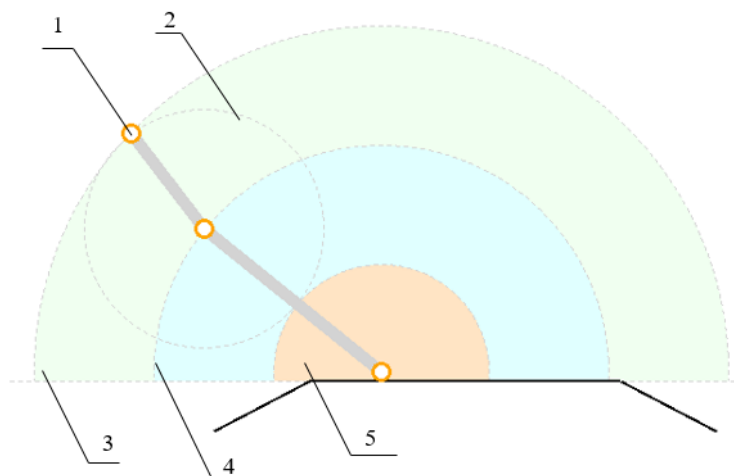
першого слайдера, тільки обчислює кут зміщення великого плеча маніпулятора та присвоює його змінній `angle1`.

При старті програми задаються початкові значення слайдерам:

```
private void Form1_Load(object sender, EventArgs e)
{
    trackBar5.Maximum = 180;
    trackBar5.Value = angle1;
    trackBar6.Minimum = -180;
    trackBar6.Maximum = 180;
    trackBar6.Value = angle2;
}
```

Для обробки стану кнопок використовуються події: `button_MouseDown` та `button_MouseUp`. Перша подія запускає таймер 1, а друга вимикає його. Перша кнопка працює з таймером 1, друга с таймером 2. Кнопка «<<<» для керування маленьким плечем маніпулятора працює з таймером 3, кнопка «>>>» для керування маленьким плечем маніпулятора – з таймером 4.

На рис. 3 показано графічний інтерфейс програми, який відображає поточний стан маніпулятора робота.



1 – положення захвату; 2 – траєкторія руху захвату маніпулятора;
3 – максимально можлива траєкторія руху; 4 – траєкторія руху середнього суглобу маніпулятора; 5 – зона недосяжності маніпулятора.

Рисунок 3 – Графічний інтерфейс програми, який відображає поточний стан маніпулятора

Для розрахунку позиції суглобів маніпулятора скористуємося рис. 4. Як можна бачити, перший суглоб L1 закріплений на основі і повернений

на кут Q_1 . Другий суглоб L_2 , кріпиться до кінця першого суглоба і повернений відносно нього на кут Q_2 . Робочий орган (захват) маніпулятора знаходиться на кінці другого суглоба.

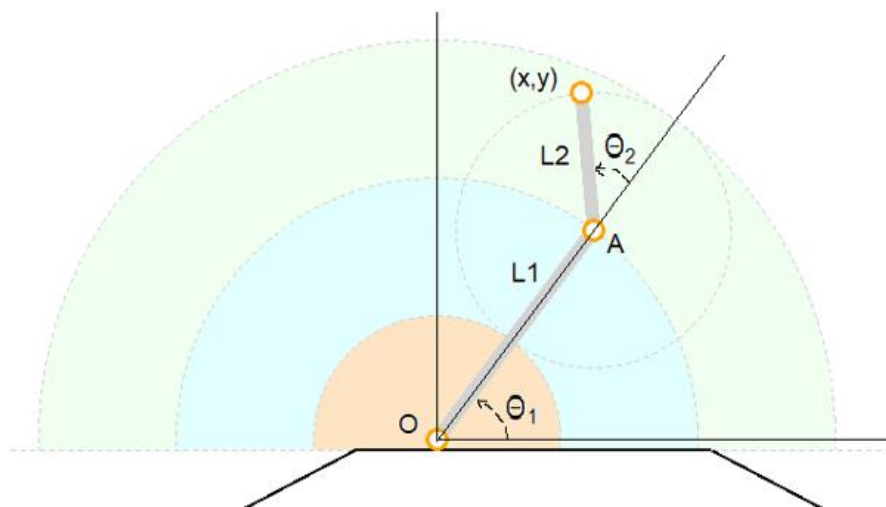


Рисунок 4 – Вихідні дані для розрахунку позиції суглобів маніпулятора

Пряме завдання кінематики полягає в знаходженні координат робочого органу (x, y) за заданим L_1, L_2, Q_1, Q_2 . L_1 і L_2 – це, відповідно, довжини плеча і ліктя маніпулятора, які в нашому випадку визначені конструкцією маніпулятора.

Для рішення даної задачі ми маємо дві системи відліку – перша, пов'язана з точкою кріплення плеча $L_1 - O$, а друга – з початком координат в точці кріплення ліктя – A . Для графічного зображення маніпулятора, знайдемо зміщення малого плеча відносно великого:

$$XA = L_1 \cdot \cos(Q_1) \quad (1)$$

$$YA = L_1 \cdot \sin(Q_1) \quad (2)$$

Координати розташування захвату (x, y) в системі відліку ліктя виразимо формулами

$$x'' = L_2 \cdot \cos(Q_2) \quad (3)$$

$$y'' = L_2 \cdot \sin(Q_2) \quad (4)$$

З рис. 4 можна бачити, що лікоть L_2 повернуто відносно плеча на кут Q_1+Q_2 , тому:

$$x' = L_2 \cdot \cos(Q_1 + Q_2) \quad (5)$$

$$y' = L_2 \cdot \sin(Q_1 + Q_2) \quad (6)$$

Таким чином координати захвату (x, y) можна обчислити наступним чином:

$$x = XA + x' = L_1 \cdot \cos(Q_1) + L_2 \cdot \cos(Q_1 + Q_2) \quad (5)$$

$$y = YA + y' = L_1 \cdot \sin(Q_1) + L_2 \cdot \sin(Q_1 + Q_2) \quad (6)$$

Виходячи з цього, позиція кожного плеча обчислюється окремо за допомогою наступної функції:

```
private void DreawLine(Graphics g)
{
    Point plecho_point1 = new Point(plecho1_start_x,
plecho1_start_y);

    plecho_Middle = new Point((int)(plecho1_start_x - plecho1 *
Math.Cos(angle1 * Math.PI / 180)),
(int)(plecho1_start_y - plecho1 * Math.Sin(angle1 * Math.PI /
180)));

    Point plecho_point2 = new Point((int)(plecho_Middle.X - plecho2
* Math.Cos((angle1 - angle2) * Math.PI / 180)),
(int)(plecho_Middle.Y - plecho2 * Math.Sin((angle1 - angle2) *
Math.PI / 180)));
    plecho_Start = plecho_point1;
    plecho_End = plecho_point2;
    DrawNormal(g, plecho_Start, plecho_End, plecho_Middle, 25);
}
```

Висновки. Таким чином, в даній роботі представлена розроблена програма для візуалізації переміщення ланок маніпулятора робота «Марсохід» в просторі. Дана програма взаємодіє з модулем керування маніпулятором через послідовний дротовий або бездротовий інтерфейс. Команди надсилаються послідовно. Кожна команда впливає на роботу конкретного двигуна в залежності від її типу. Головний інтерфейс програми об'єднує всі візуальні компоненти та обробляє події від активних компонентів.

Список використаних джерел.

1. Петров И. В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования [Текст] / Под ред. проф. В. П. Дьяконова. – М. : СОЛОН-Пресс, 2004 г. – 266 с.
2. Багаева Т. А. Автоматизация процесса реконфигурации сложной технической системы // Автоматизация в промышленности. 2009. №10. с. 11-14.
3. Барщевский Е. Г., Зубарев Ю. Я., Солдатенко С. А. Активная идентификация автоматизированных систем на основе вычислительного эксперимента Текст. // Программные продукты и системы. 2009. №1. с. 15-17.