



Współfinansowane ze
środków Unii Europejskiej



DOI: 10.5604/20830157.1148055

ZAUTOMATYZOWANY SYSTEM STEROWANIA RUCHEM ROBOTA NA ZADANEJ TRAJEKTORII Z MOŻLIWOŚCIĄ OMINIĘCIA PRZESZKÓD W OPARCIU O LOGIKĘ ROZMYTĄ

Oksana Radion

Narodowy Uniwersytet Techniczny w Łucku, Wydział Automatyknej Kontroli Procesów Produkcyjnych

Streszczenie. Artykuł dotyczy metody projektowania robota mobilnego. Opracowany został algorytm ominięcia przeszkód oraz oprogramowanie Processing co umożliwi ustawienie punktów kontrolnych trajektorii oraz zapisu ich w pamięć robota. Przeanalizowano system ruchu robota z ominięciem przeszkód na podstawie logiki rozmytej.

Słowa kluczowe: automatyczna kontrola, logika rozmyta, sterowanie robotem

АВТОМАТИЗОВАНА СИСТЕМА КЕРУВАННЯ РУХОМ РОБОТА ПО СКЛАДНІЙ ТРАЄКТОРІЇ З ОБМИНАННЯМ ПЕРЕШКОД НА БАЗІ НЕЧІТКОЇ ЛОГІКИ

Анотація. В роботі розглянуто методи проектування мобільного робота. Розроблений алгоритм обминання перешкоди та програмне забезпечення, яке на мові програмування Processing дозволяє задавати опорні точки маршруту та записувати їх в пам'ять робота. Розроблена система руху робота з обминанням перешкод на базі нечіткої логіки.

Ключові слова: автоматичне управління, нечітка логіка, управління роботом

AUTOMATED SYSTEM TO CONTROL A ROBOT'S MOVEMENT ON COMPLEX TRAJECTORY WITH AVOIDING OBSTACLES BASED ON FUZZY LOGIC

Abstract. We consider methods for designing mobile robot. The algorithm of avoiding obstacles and software based on the programming language Processing that allows to set the anchor points of the route and record them in robot's memory were worked out. The system of the robot movement with avoiding obstacles based on fuzzy logic was developed.

Keywords: automatic control, fuzzy logic, robot control

Wstęp

Obecnie istnieje wiele pojazdów. Każdego roku, każdy następny stworzony model samochodu jest wygodniejszy i bardziej wydajny [1]. Jednak automatyczne sterowanie ruchem pojazdu zaczęło nabierać tempa dopiero w ostatnich czasach, pomimo faktu, że rozwój pojazdów to około 110 lat pracy inżynierskiej. Ten artykuł ma na celu dalszy rozwój automatycznego sterowania ruchem pojazdu.

1. Główne wyniki badań

Pojazd można przedstawić, z pewnymi uproszczeniami, jako model fizyczny robota na platformie czołgowej. Platforma czołgowa posiada dwa niezależne silniki, które upraszczają sterowanie ruchem.

W tym artykule, do sterowania ruchem robota została wybrana platforma Arduino z mikroprocesorem rodziny AVR ze względu na łatwość programowania, elastyczność wbudowanego w urządzenie oprogramowania, wysoką wydajność oraz przystępne ceny dla takich celów. Również wykorzystany został komputer osobisty, który służy do wyznaczania trasy ruchu obiektu oraz przekazywania jej do robota. Robot, a dokładniej platforma Arduino, z kolei odbiera trasę, przekształcając ją w wygodną dla siebie formę, steruje ruchem robota i w przypadku napotkania przeszkody reaguje nią objeżdżając ją [2, 3].

Analiza istniejących metod służących do określania odległości do przeszkód pokazała, że najbardziej skuteczną metodą pomiaru odległości jest metoda oparta o zastosowanie czujnika ultradźwiękowego.

Aby wyznaczyć przeszkodę w tym przypadku wykorzystuje się dwa czujniki ultradźwiękowe HC-SR04, które wysyłają fale ultradźwiękowe i oczekują na odbicia (rys. 1).

Вступ

В наш час є багато транспортних засобів. З кожним роком, з кожною наступною моделлю автомобіля створюються все зручніші і ефективніші машини [1]. Однак, автоматизація керування рухом транспортного засобу почала набирати оберти тільки в недавній час, незважаючи на те, що розвиток транспортних засобів нараховує близько 110 років інженерної праці. Ця наукова робота спрямована підштовхнути автоматизацію керування рухом транспортного засобу.

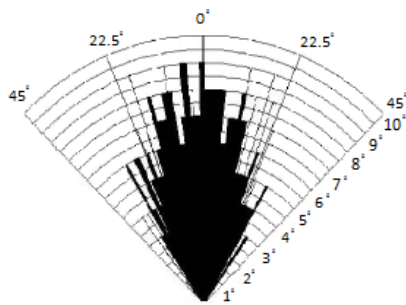
1. Основні результати дослідження

Транспортний засіб можна представити, з деякими спрощеннями, як фізичну модель робота на танковій платформі. Танкова платформа має два незалежних двигуни, які спрощують управління рухом.

Для керування рухом робота була вибрана плата Arduino з мікропроцесором сімейства AVR за легкість програмування, гнучкість прошивки, високу швидкодію і помірну ціну для таких цілей. Також використовується ПК, який бере на себе роботу по встановленню маршруту руху об'єкта і передачі цього самого маршруту до робота. Робот, а точніше плата Arduino, в свою чергу, виконує роботу по прийому маршруту, перетворення його в зручний для себе вигляд, управління рухом робота, і в разі зустрічі перешкоди реагувати на це, об'їздом [2, 3].

Аналіз існуючих методів визначення дистанції до перешкоди показав найбільш ефективний метод вимірювання відстані, який базується на використанні ультразвукового датчика.

Для визначення перешкоди в цьому випадку використовується два ультразвукові датчики відстані HC-SR04, які посилають механічні хвилі ультразвукової частоти і очікують відбиття рис. 1



Rys. 1. Wykres kierunkowości czujnika HC-SR04
 Рис.1. Диаграма спрямованості датчика HC-SR04

Jeśli odbicie miało miejsce, czujniki wysyłają do platformy Arduino sygnał w postaci prostokątnego impulsu o pewnej częstotliwości zależnej od odległości do przeszkody. Dlatego można określić odległość do przeszkody znając charakterystykę częstotliwości impulsu od odległości do przeszkody:

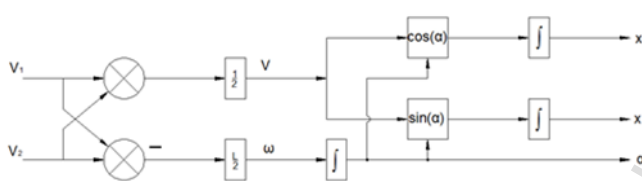
$$D = \frac{S}{58} \quad (1)$$

gdzie D – odległość do badanego obiektu, cm; S – szerokość impulsów sygnału, mikrosekundy.

Do sterowania ruchem silnika elektrycznego w lewo i prawo został wybrany układ L293D, który jest łatwy w użyciu oraz do zamontowania w panelu.

Rysunek 2 przedstawia schemat skonstruowany na podstawie wytycznych opisanych powyżej [4].

Aby przeanalizować ruch robota został wykorzystany model kinematyczny z napędem na dwa koła. Rysunek 3 pokazuje model kinematyczny robota, a na rys. 4 robot jest przedstawiony schematycznie z dołączonymi wektorami prędkości.



Rys. 3. Kinematyczny model robota
 Рис. 3. Кінематична модель робота

Matematycznie dany model może być opisany następująco:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(v_1 + v_2) \cos \alpha \\ \frac{1}{2}(v_1 + v_2) \sin \alpha \\ \frac{L}{2}(v_1 - v_2) \end{pmatrix} \quad (2)$$

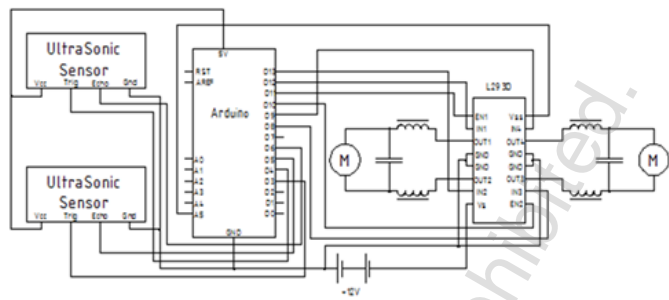
gdzie v_1 , v_2 – prędkość liniowa odpowiednio pierwszego i drugiego koła napędowego; L – długość ramienia obrotowego; x , y – współrzędne środka masy robota; α – kąt obrotu robota względem kierunku północ.

Wprowadźmy oznaczenie:

$$\begin{aligned} v &= \frac{1}{2}(v_1 + v_2) \\ \omega &= \frac{1}{2}(v_1 - v_2) \end{aligned} \quad (3)$$

Następnie można uprościć formułę podaną wyżej:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{pmatrix} = \begin{pmatrix} \cos \alpha \\ \sin \alpha \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega \quad (4)$$



Rys. 2. Schemat elektryczny robota
 Рис.2. Електрична схема робота

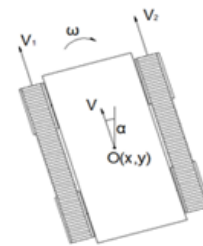
Якщо відбиття відбулося, то сенсори посилюють сигнал у виді прямокутного імпульсу до плати Arduino певної тривалості, залежно від відстані до перешкоди. Тому можна визначити відстань до перешкоди, знаючи закономірність тривалості імпульсу від відстані до перешкоди:

де D – довжина до вимірюваного об'єкта, см; S – ширина сигналу, мкс.

Для керування електродвигунами приводів лівого та правого рушіїв була обрана мікросхема L293D, як проста в використанні, що монтується в панель.

На рис. 2 зображено принципову електричну схему побудовану на основі ресурсів про які згадується вище [4].

Щоб відслідковувати рух робота, було використано кінематичну модель з двома ведучими колесами. На рис. 3 зображена кінематична модель робота, а на рис. 4 зображено схематично робот із прикладеними векторами швидкостей.



Rys. 4. Схематичне представлення робота
 Рис. 4. Схематичне зображення робота

Математично дану модель можна описати так:

де v_1 , v_2 – лінійні швидкості відповідно першого і другого ведучого колеса; L – довжина плеча обертання; x , y – координати центра маси робота; α – кут повороту робота відносно півночі.

Введемо заміну:

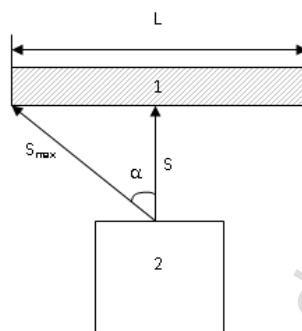
Тоді можна спростити вище наведену формулу:

Widząc więc, że system jest prosty matematycznie, ale niedoskonały fizycznie.

Na podstawie przeprowadzonego eksperymentu stwierdzono, że czujnik przestaje rejestrować sygnał odbity pod kątem $17 \dots 24^\circ$ pomiędzy wektorem emisji sygnałów i normalną do płaszczyzny odbicia. Biorąc pod uwagę, że przeszkody to równoległościanny, a maksymalna długość precyzyjnego działania czujników wynosi 100 cm, możliwe jest określenie odległości, od której można rozpocząć pomiar szerokości przeszkody. Rys. 5 przedstawia diagram pomiaru szerokości przeszkody, gdzie $\alpha = 17 \dots 24^\circ$, $S_{\max} = 100$ cm. Za pomocą cosinusa kąta można znaleźć minimalną długość S , z której można rozpocząć pomiar szerokości przeszkody L :

$$S = S_{\max} \cos \alpha = 100 \text{ cm} \cdot 0.91 = 91 \text{ cm} \quad (5)$$

Tak więc pomiar szerokości można dokonać od 91 cm.



Rys. 5. Schemat pomiaru szerokości przeszkody (1 - przeszkoda, 2 - robot)

Aby zmierzyć szerokość przeszkód wystarczy użyć formuły tangensa kąta:

$$L = S \cdot \tan \alpha \quad (6)$$

W celu pomiaru całej długości przeszkód, potrzeba zmierzyć kąt obrotu w prawo i w lewo w stosunku do położenia początkowego robota. Zakładając, że robot jest w równej odległości od obu krawędzi przeszkody można znaleźć maksymalną szerokość przeszkody, którą będzie mógł zaakceptować robot i uniknąć jej. W tym przypadku formuła jest następująca:

$$L = S \cdot \tan \alpha = 2 \cdot 91 \cdot 0.445 = 81 \text{ cm} \quad (7)$$

W przypadku gdy szerokość przeszkody jest większa, robot nie będzie mógł jej uniknąć ze względu na niedoskonałości czujników ultradźwiękowych.

Po odnalezieniu przeszkody, obliczeniu jej szerokości i wyborze najlepszej strony obrotu, do trajektorii dodaje się kolejny punkt o współrzędnych (x_m, y_m) , które są obliczane według następującego wzoru:

$$\begin{aligned} x_m &= x_0 + \left(S + \frac{W}{2} + E \right) \sin \alpha \\ y_m &= y_0 + \left(S + \frac{W}{2} + E \right) \cos \alpha \end{aligned} \quad (8)$$

gdzie x_0, y_0 – współrzędne środka masy robota przed objazdem przeszkody; S – odległość do przeszkody; H, W – szerokość i wysokość robota; E – rezerwa odległości pomiędzy robotem a przeszkodą ($\sim 1 \dots 5$ cm); α – optymalny kąt objazdu przeszkody.

Algorytm uniknięcia przeszkody (objazdu) wykorzystuje logikę rozmytą [5, 6]. Logika rozmyta, w przeciwieństwie do logiki boolowskiej, ma nie dwie wartości (prawdziwą i fałszywą), a zbiór wartości. W logice boolowskiej wystarczyłoby stwierdzić czy jest przeszkoda, a następnie zacząć odchodzić od niej. Ten sposób jest tani ale nieskuteczny. Algorytm obejścia przeszkody z wykorzystaniem logiki rozmytej polega na tym, aby zmienić prędkość i kąt obrotu względem kierunku do przeszkody, umożliwiając płynne i bardziej efektywne uniknięcie przeszkody.

Отже, як бачимо, система є математично простою, але фізично не досконалою.

Експериментальним шляхом було визначено, що датчик перестає сприймати випромінений сигнал при куті $17 \dots 24$ градуси між вектором випромінювання сигналу і нормаллю площини відбиття. Якщо врахувати що перешкоди являють собою паралелепіпеди, і максимальну довжину точної дії датчика 100 см, то можна визначити з якої відстані можна почати вимірювання ширини перешкоди. На рис. 5 зображена схема вимірювання ширини перешкоди. Тут $\alpha = 17 \dots 24^\circ$, $S_{\max} = 100$ см. З використанням формули косинуса кута можна знайти мінімальну відстань S , з якої можна починати вимірювання ширини перешкоди L :

Отже починати вимірювання ширини можна з 91 см.

Рис. 5. Схема виміру ширини перешкоди (1 - перешкода, 2 - робот)

Щоб виміряти ширину перешкоди достатньо скористатися формулою тангенса кута:

Щоб виміряти всю довжину перешкоди, потрібно виміряти кут повороту направо і наліво відносно початкового положення робота. Припускаючи що робот знаходиться на однаковій відстані від двох країв перешкоди можна знайти максимальну ширину перешкоди, яку зможе сприйняти робот, а отже і об'їхати. В цьому випадку формула буде така:

У випадку, коли ширина перешкоди буде більшою, робот не зможе її об'їхати через недоліки ультразвукового датчика.

Після того як було виявлено перешкоду, обчислено її ширину і обрано в яку сторону краще повертати, в траєкторію руху додається ще одна точка з координатами (x_m, y_m) , які обчислюються за наступною формулою:

де x_0, y_0 – координати центра маси робота, перед об'їздом перешкоди; S – відстань до перешкоди; H, W – відповідно висота і ширина робота; E – запас ходу між перешкодою і роботом ($\sim 1 \dots 5$ см); α – оптимальний кут обходу перешкоди.

В якості алгоритму по обминанню перешкод використовується нечітка логіка [5, 6]. Нечітка логіка, на відміну від булевої логіки, має не два значення (true і false), а множини значень. При булевій логіці достатньо було б визначити, чи є перешкода, і тоді почати відхилятися від неї. Це є дешевий і неефективний спосіб. Алгоритм обходу перешкоди з застосуванням нечіткої логіки заключається в тому, щоб змінювати швидкість, а отже і кут повороту відносно відстані до перешкоди, що дозволяє плавно і більш ефективно рухатися в обхід перешкоди.

W tym przypadku użyto algorytm logiki rozmytej zaproponowany przez E.Mamdani. Formalnie algorytm Mamdaniego jest zdefiniowany w następujący sposób [7, 8]:

- kształtowanie bazy reguł wnioskowania rozmytego;
- stopniowanie zmiennych wejściowych;
- łączenie warunków w regułach rozmytych;
- uruchomienie wniosków w regułach rozmytych;
- akumulacja wniosków w regułach rozmytych;
- defuzyfikacja zmiennych wyjściowych.

Jako dane wejściowe wybrano „przesunięcie na osi x”, „przesunięcia na osi pionowej”, które są zdefiniowane jako różnica pomiędzy następnym punktem trajektorii i aktualną pozycją robota. Jako wyjściowe - zmienne bezwymiarowe opisujące „wymaganą prędkość” i „wymaganą prędkość kątową”.

Dla wszystkich zmiennych wejściowych i wyjściowych budujemy zmienne lingwistyczne.

Y_1 = „przesunięcie na osi odciętych”.

Termy: ujemna duża; ujemna mała; zerowa; dodatnia mała; dodatnia duża.

Y_2 = „przesunięcie na osi rzędnych”.

Termy: ujemna duża; ujemna mała; zerowa; dodatnia mała; dodatnia duża.

U_1 = „wymagana prędkość”.

Termy: zerowa; mała; średnia; duża.

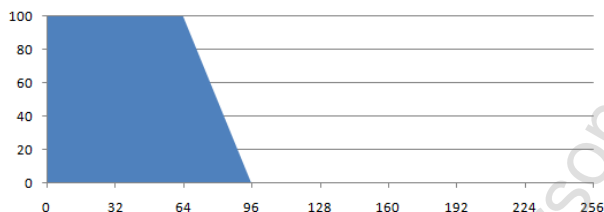
U_2 = „wymagana prędkość kątową”.

Termy: ujemna duża; ujemna mała, zerowa, dodatnia mała, dodatnia duża.

W tym przypadku system wnioskowania rozmytego będzie miał 25 rodzajów rozmytych reguł produkcji typu:

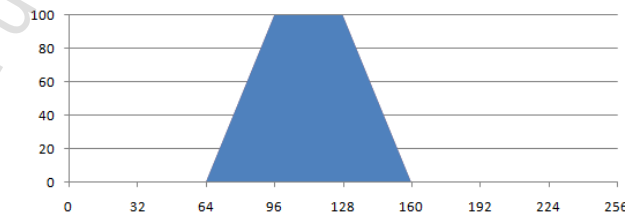
„Zasada #: Jeśli <warunek>, to <akcja>”

W przypadku danego robota zmienna wyjściowa „wymagana prędkość” zawiera zmienne rozmyte a_1 = „zerowa prędkość”, a_2 = „niska prędkość”, a_3 = „średnia prędkość”, a_4 = „wysoka prędkość”. Zdefiniujemy funkcje graficznie w postaci zależności odcinkowo-liniowych. Wykresy dla „wymaganej prędkości” pokazano na rys. 6, 7, 8, 9.



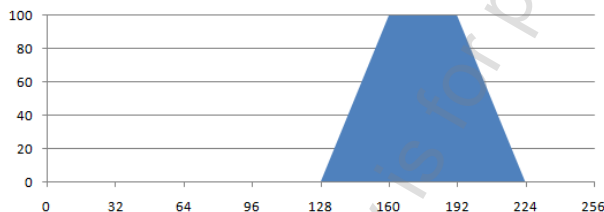
Rys. 6. Wykres funkcji „prędkość zero”

Rys. 6. Wykres funkcji „prędkość zero”



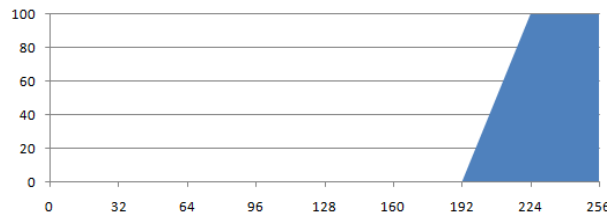
Rys. 7. Wykres funkcji „prędkość mała”

Rys. 7. Wykres funkcji „prędkość mała”



Rys. 8. Wykres funkcji „prędkość średnia”

Rys. 8. Wykres funkcji „prędkość średnia”



Rys. 9. Wykres funkcji „prędkość wysoka”

Rys. 9. Wykres funkcji „prędkość wysoka”

Podobnie, możemy wprowadzić inne zmienne lingwistyczne za pomocą wykresów. Na rysunkach 10, 11, 12, 13 przedstawione są wykresy lingwistycznych zmiennych, a mianowicie „wymagana prędkość”, „wymagana prędkość kątową”, „przesunięcie na osi odciętych”, „przesunięcie na osi rzędnych”. Na osi odciętych w wykresach prędkości są liczby od 0 do 256, wykazujące prędkość w zależności od modulacji szerokości impulsów, które są podawane na wejście sterownika silników. Na wykresach pokazane są przemieszczenia wzdłuż osi odciętych i rzędnych wyrażone w cm. Na osi rzędnych wykresu zaznaczono poziom przynależności.

В даному випадку використаємо систему нечіткого виводу Мамдані. Формально алгоритм Мамдані визначається наступним чином [7, 8]:

- формування бази правил систем нечіткого виводу;
- фазифікація вхідних змінних;
- агрегація підумов в нечітких правилах продукції;
- активізація підвисновків в нечітких правилах продукції;
- акумуляція висновків в нечітких правилах продукції;
- дефазифікація вихідних змінних.

В якості вхідних даних вибираємо «зміщення по осі абсцис», «зміщення по осі ординат», які визначаються як різниця між наступною точкою маршруту і теперішнім положенням робота. На виході будуть безрозмірні змінні, які характеризують «потрібну швидкість» і «потрібну куту швидкість».

Для всіх вхідних і вихідних змінних будемо лінгвістичні змінні.

Y_1 = «зміщення по осі абсцис».

Терми: від’ємна велика; від’ємна мала; нульова; додатна мала; додатна велика.

Y_2 = «зміщення по осі ординат».

Терми: від’ємна велика; від’ємна мала; нульова; додатна мала; додатна велика.

U_1 = «потрібна швидкість».

Терми: нульова; мала; середня; велика.

U_2 = «потрібна куту швидкість».

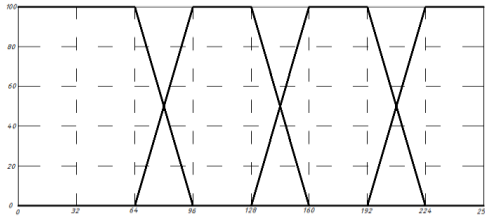
Терми: від’ємна велика; від’ємна мала; нульова; додатна мала; додатна велика.

В цьому випадку система нечіткого виводу буде мати 25 правил нечітких продукцій виду:

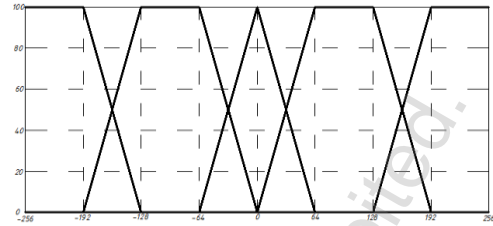
«Правило #: Якщо <умова>, то <дія>»

Для розглядуваного робота вихідна змінна «потрібна швидкість» містить в собі нечіткі змінні a_1 = «нульова швидкість», a_2 = «мала швидкість», a_3 = «середня швидкість», a_4 = «висока швидкість». Задамо функції належності графічно з допомогою кусково-лінійних функцій. Зображення графіків для «потрібної швидкості» приведено на рис. 6, 7, 8, 9.

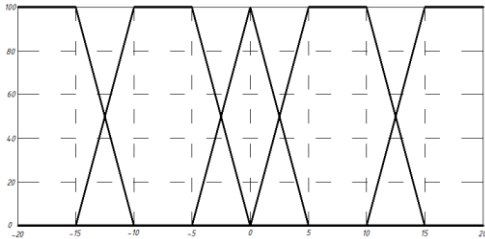
Аналогічно можна представити і інші лінгвістичні змінні в якості графіків. На рис. 10, 11, 12, 13 зображені графіки належностей лінгвістичних змінних відповідно «потрібна швидкість», «потрібна куту швидкість», «зміщення по осі абсцис», «зміщення по осі ординат». По осі абсцис на графіках з швидкостями наведені числа від 0 до 256, що показують швидкість в залежності від ШИМ-сигналу який подається на вхід драйвера двигунів. На графіках з зміщенням по осях абсцис і ординат відкладають відстані в см. По осях ординат на графіках відзначають рівень належності.



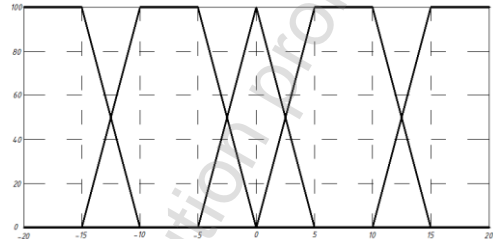
Rys. 10. Wykres funkcji przynależności „wymagana prędkość”
 Рис. 10. Графік функції належності «потрібна швидкість»



Rys. 11. Wykres funkcji przynależności „wymagana prędkość kąta”
 Рис. 11. Графік функції належності «потрібна кутова швидкість»



Rys. 12. Wykres funkcji przemieszczenia na osi odciętej
 Рис. 12. Графік функції належності «зміщення по осі абсцис»



Rys. 13. Wykres funkcji przemieszczenia na osi rzędnych
 Рис. 13. Графік функції належності «зміщення по осі ординат»

Dla każdej z reguł jest obliczony stopień prawdopodobieństwa. Następnie, dla każdej z reguł, są obliczone poziomy odcięcia według formuły:

$$a_i = A_i(x_0) \cap B_i(y_0) \tag{9}$$

gdzie x_0, y_0 – ściśle określone wartości danych wejściowych; A_i, B_i – odpowiednio rozmyte wartości danych wejściowych; a_i – poziomy odcięcia; i – numer reguły.

Znak „ \cap ” oznacza funkcję logicznego minimum. Następnie dla każdej reguły są znajdowane odcięcia funkcji według wzoru:

$$C'_i = a_i \cap C_i(z) \tag{10}$$

gdzie C'_i – wartość odcięcia funkcji przynależności i-tej reguły; C_i – rozmyte wartości wyniku i-tej reguły.

Następnym etapem jest kompozycja, w której odbywa się połączenie odnalezionych odciętych wartości funkcji przynależności, prowadzące do uzyskania końcowej rozmytej wartości zgodnie z formułą:

$$\mu_z(z) = C(z) = C'_1(z) \cup C'_2(z) \dots \cup C'_i(z) \tag{11}$$

gdzie μ_z – wartość końcowa rozmytego zbioru.

Znak „ \cup ” oznacza maksymalną funkcję logiczną.

Następnym etapem jest defuzyfikacji czyli określenie dokładnej wartości rozmytego zbioru. Jest wiele sposobów przeprowadzenia defuzyfikacji. W pracy zastosowano metodę w stosunku do uśrednionego centrum, które jest obliczane według wzoru:

$$y = \frac{\sum_{i=1}^M \mu(c_i) \cdot c_i}{\sum_{i=1}^M \mu(c_i)} \tag{12}$$

gdzie y – wynik defuzyfikacji; c_i – centrum i-tego obszaru; $\mu(c_i)$ – funkcji przynależności centrum i-tego obszaru.

Ogólna struktura systemu rozmytego sterowania robotem mobilnym jest przedstawiona na rys. 13.

Dla każdego pravidła znajdują się stopieny istinności. Potim dla każdego pravidła znajdują się rıvni vıdсіcnenıya za formuloju:

de x_0, y_0 – vıdpovidno cıtkı znaczenıya vıhıdnıx danıx; A_i, B_i – vıdpovidno necıtkı znaczenıya vıhıdnıx danıx; a_i – rıvni vıdсіcnenıya; i – nomer pravidla;

Znak « \cap » poznaacze funkcıju logıcnoho mınuımu. Potim dla każdego pravidla znajdują się usıceni funkcıı nalejnosdı za formuloju:

de C'_i – usıceni znaczenıya funkcıı nalejnosdı кожного i-ogo pravidla; C_i – necıtkı znaczenıya rezulıtatu кожного i-ogo pravidla.

Potim ıde etap kompozıcıı, de vıdbuvaetsıya ob'ıdnannı znayıdnıx usıceni znaczenıy funkcıı nalejnosdı, cıo prıvodıty do otrımannıya pıdsıumkovej necıtkı mnojıny, za formuloju:

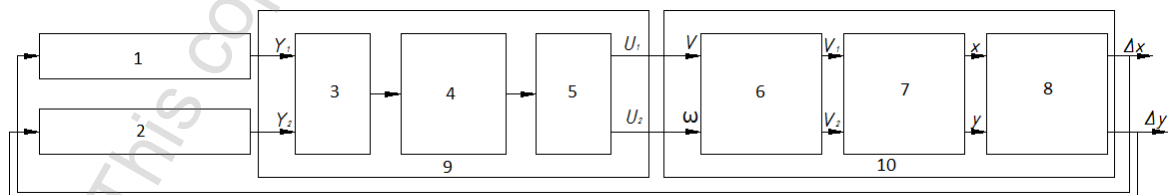
de μ_z – rezulıtuoja pıdsıumkoja necıtkı mnojına.

Znak « \cup » poznaacze funkcıju logıcnoho maksimuma.

Potim ıde etap defazıfıkacıı, tobo vıznaczenıya cıtkogo znaczenıya vıd necıtkı mnojıny. Dla defazıfıkacıı ı baato metodıv. Ya vıkorıstav metod vıdnosno serednoho centru, yakıy rozrahojujeť za formuloju:

de y – rezulıtat defazıfıkacıı; c_i – centru i-oı oblastı; $\mu(c_i)$ – funkcıja nalejnosdı centru i-oı oblastı.

Zagalna struktura sıstemı necıtkogo upravlnınnı mobılnoju robotom zobrazena na rys. 13.



Rys. 13. Schemat ogólny sterowania rozmytego robotem (1 - różnica na osi odciętych, 2 - różnica na osi rzędnych, 3 - fazyfikacja, 4 - baza reguł - inferencja, 5 - defuzyfikacja, 6 - wartość równania kinematycznego, 7 - kinematyczny model robota, 8 - przejście do zmiennych trajektorii, 9 - regulator rozmyty, 10 - moduł robota)
 Рис. 13. Загальна схема нечіткого управління роботом (1 - різниця по осі абсцис, 2 - різниця по осі ординат, 3 - фазифікація, 4 - база правил - інференція, 5 - дефазифікація, 6 - розв'язок оберненої кінематичної задачі, 7 - кінематична модель робота, 8 - перехід до траєкторних змінних, 9 - фuzzi-регулятор, 10 - модуль робота)

2. Przykłady wyznaczenia prędkości przez regulator rozmyty

Założmy, że mamy następujące dane: $\Delta x = 10$; $\Delta y = 12,5$. Dokonujemy fazyfikacji wartości danych, na wyjściu otrzymujemy tablicę wartości funkcji przynależności każdej zmiennej lingwistycznej. Wartość zero nie wpływa na obliczenie - omijamy termy, które mają takie wartości.

$$Y_1 = \{„dm” = 1\},$$

$$Y_2 = \{„dm” = 0,5; „dd” = 0,5\},$$

gdzie „dm” – terma „dodatnia mała”;

„dd” – terma „dodatnia duża”.

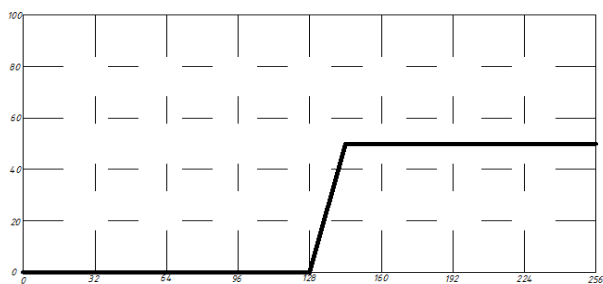
Wybierając reguły, w których dane termy nie są równe zero uzyskujemy poziom odcięcia dla każdej z reguł:

$$P19: U_1 = \{„s” = 0,5\}, i U_2 = \{„VM” = 0,5\};$$

$$P20: U_1 = \{„w” = 0,5\}, i U_2 = \{„VM” = 0,5\};$$

Teraz musimy znaleźć funkcję przynależności odciętej funkcji za pomocą funkcji logicznego minimum i połączyć je za pomocą funkcji logicznego maksimum. Na wyjściu otrzymujemy nowy zestaw rozmyty (rys. 14 i 15).

wymagana prędkość || потрібна швидкість



Rys. 14. Rozmyty zbiór zmiennej „wymagana prędkość”

Рис. 14. Нечітка множина змінної «потрібна швидкість»

Teraz defazyfikujemy rozmyte zbiory żeby znaleźć dokładne wartości prędkości liniowej i kątowej. Aby to zrobić, należy użyć opisanej powyżej metody w stosunku do uśrednionego centrum:

$$v = \frac{176 \cdot 0,5 + 224 \cdot 0,5}{0,5 + 0,5} = 200$$

$$\omega = \frac{-80 \cdot 0,5 + (-80) \cdot 0,5}{0,5 + 0,5} = -80$$

Na podstawie tych wartości można uzyskać wartość prędkości liniowej każdego koła napędowego, przy rozwiązywaniu układu równań (14):

$$\begin{cases} \frac{v_1 + v_2}{2} = 200 \\ \frac{(v_1 - v_2) \cdot L}{2} = -80 \end{cases} \quad (14)$$

W naszym przypadku L (długość obrót ramienia) równa się 5 cm.

В нашому випадку L (довжина плеча обертання) дорівнюватиме 5 см.

$$\begin{cases} v_2 = 400 - v_1 \\ v_1 - 400 + v_1 = -\frac{160}{5} \end{cases} \quad (15)$$

$$\begin{cases} v_1 = 184 \\ v_2 = 216 \end{cases} \quad (16)$$

W związku z tym, sterownik szerokości impulsów silnika należy wysterować wartościami 184 i 216. Powoduje to, że pierwszy silnik jest sterowany napięciem 6,5V, a drugi – 7,62V. Dalej następuje obliczenie nowych współrzędnych i proces ten powtarza się.

Отже, треба подати на драйвер двигунів через широтно-імпульсну модуляцію числа 184 i 216. Таким чином на перший двигун спаде напруга 6,5В, а на другий – 7,62В. Потім відбувається обчислення нових координат і процес повторюється.

2. Приклад знаходження швидкостей через нечіткий регулятор

Припустимо, що в нас є такі дані: $\Delta x = 10$; $\Delta y = 12,5$. Фазифікуємо дані значення, на виході отримаємо масив значень функції належності кожної лінгвістичної змінної. Так як значення нуля не впливає на розрахунок, пропустимо терми які мають таке значення.

$$Y_1 = \{«дм» = 1\},$$

$$Y_2 = \{«дм» = 0,5; «дв» = 0,5\},$$

де «дм» - терма «додатня мала»;

«дв» - терма «додатня велика».

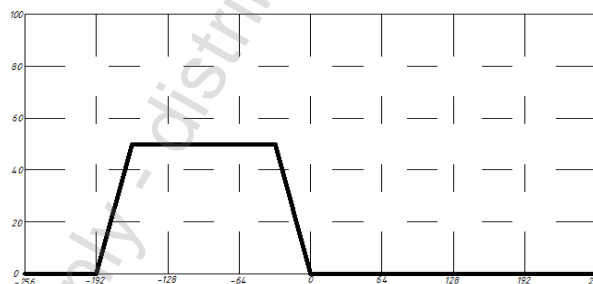
Вибираємо правила в яких дані терми не є нульовими, і знаходимо рівні відсічення для кожного з правил:

$$P19: U_1 = \{«с» = 0,5\}, i U_2 = \{«вм» = 0,5\};$$

$$P20: U_1 = \{«в» = 0,5\}, i U_2 = \{«вм» = 0,5\};$$

Тепер треба знайти усічені функції належності за допомогою логічної функції мінімуму, і об'єднати їх за допомогою логічної функції максимуму. На виході отримаємо нову нечітку множину (рис. 14, рис. 15).

wymagana prędkość kątowa || потрібна кутлова швидкість



Rys. 15. Rozmyty zbiór zmiennej „wymagana prędkość kątowa”

Рис. 15. Нечітка множина змінної «потрібна кутлова швидкість»

Тепер дефазифікуємо нечіткі множини щоб знайти точні значення лінійної і кутової швидкості. Для цього використаємо вище згадуваний метод відносно середнього центра:

3. Wnioski

- 1) W pracy rozważono metody projektowania robota mobilnego oraz został uzasadniony wybór robota mobilnego z napędem gąsienicowym.
- 2) Analiza istniejących metod wyznaczenia dystansu do przeszkody wykazała, że najbardziej efektywną metodą pomiaru odległości jest metoda wykorzystująca zastosowaniu czujniki ultradźwiękowe.
- 3) Do sterowania napędowym silnikiem elektrycznym został wybrany układ L293D.
- 4) Opracowany algorytm omijania przeszkody oraz oprogramowanie, które w języku programowania Processing pozwala na ustawienie punktów kontrolnych trasy i ich zapisanie w pamięci robota.
- 5) Badany system śledzenia ruchu nie wykorzystuje wbudowanej nawigacji.
- 6) Opracowany system ruchu robota z uniknięciem przeszkód oparty został na logice rozmytej.

Podziękowania

Praca powstała w ramach projektu PL-NTU Transgraniczna wymiana doświadczeń PBU.03.01.00-06-386/11-00 współfinansowanego w ramach Programu Współpracy Transgranicznej Polska-Białoruś-Ukraina 2007-2013 finansowanego ze środków Unii Europejskiej w ramach Europejskiego Instrumentu Sąsiedztwa i Partnerstwa.

Niniejsza publikacja została stworzona przy pomocy Unii Europejskiej. Wyłączną odpowiedzialność za zawartość niniejszej publikacji ponosi Oksana Radion oraz w żaden sposób nie może być ona postrzegana jako odzwierciedlenie poglądów Unii Europejskiej.

Literatura || Література

- [1] Eberkard R., Simpson P., Dobbins R.: Computational Intelligence PC Tools. AP Professional, 1996.
- [2] Gradetskiy V.G., Veshnikov V.B., Kalinicheko S.V. Upravlyayemye dvizheniya mobil'nykh robotov po proizvol'no oriyentirovannym v rostranstve poverkhnostyam, Nauka, Moskva, 2001. (In Russian).
- [3] Laumond J.P., Sekhavat S., Lamiroux F.: Guidelines in Nonholonomic Motion Planning for Mobile Robots, Springer, 1998.
- [4] Novikov YU.V.: Vvedeniye v tsifrovuyu skhemotekhniku. INTUIT.ru, 2006. (In Russian).
- [5] Reas C., Fry B. (Foreword by Maeda J.): Processing: A Programming Handbook for Visual Designers and Artists, MIT Press, 2007.
- [6] Yakh'yayeva G.E.: Nechetkiye mnozhestva i neyronnyye seti. Binom. Laboratoriya znaniy, Moskva, 2008. (In Russian).
- [7] Zade L.: Ponyatiye lingvisticheskoy peremennoy i yego primeneniye k prinyatiyu priblizhennykh resheniy. Mir, Moskva, 1976. (In Russian).
- [8] Zadeh L.A.: Fuzzy sets. Information and Control, vol. 8, N 3, 1965, 338-353.

Mgr Oksana Radion
e-mail: oksval.girl@gmail.com

Radion Oksana urodziła się w 1992 roku. Magistrantka Wydziału Automatycznej Kontroli Procesów Produkcyjnych w Narodowym Uniwersytecie Technicznym w Łucku. Dwukrotnie otrzymała stypendium Funduszu Charytatywnego Igora Pałyci. Posiada doświadczenie w zakresie zintegrowanej technologii informatycznej i komputerowej, w szczególności w zakresie rozwoju oprogramowania i sprzętu komputerowego. Uczestniczka wielu konferencji.



Магістр Оксана Радіон
e-mail: oksval.girl@gmail.com

Радіон Оксана Олександрівна народилась в 1992 році. Магістр кафедри автоматизованого управління виробничими процесами Луцького національного технічного університету. Дворазовий стипендіат благодійного фонду «Фонд Палиці». Має досвід роботи в галузі комп'ютерно-інтегрованих технологій, зокрема в розробці програмно-апаратних комплексів. Є учасником багатьох конференцій.

otrzymano/отримано/received: 04.03.2015

przyjęto do druku/ прийнято до друку/accepted: 15.03.2015