Uniwersytet Zielonogórski
Instytut Sterowania i Systemów Informatycznych

Laboratorium Sterowania Robotów

Lokalizacja robota Lego Mindstorms NXT przy użyciu odometrii

Uwagi wstępne

1. Wszystkie przykłady i zadania wykonujemy w środowisku Matlab z użyciem skrzynki narzędziowej `RWTHMindstormsNXT`.
2. Komunikację z robotem rozpoczynamy od podłączenia poprzez kabel USB. Następnie wykorzystujemy połączenie poprzez Bluetooth, zgodnie z zaleceniami prowadzącego.
3. Poniższy opis został przygotowany na podstawie informacji umieszczonych na stronach:

   (a) `http://www.mindstorms.rwth-aachen.de/`.

   (b) `http://www.inpharmix.com/jps/PID_Controller_For_Lego_Mindstorms_Robots.html`

Budowa i oprogramowanie robota

1. Celem ćwiczenia jest określenie pozycji robota lub zmian jego pozycji, czyli współrzędnych x, y oraz orientacji (kąt θ) w odniesieniu do czasu. Najważniejsze będzie jednak określenie pozycji i orientacji robota na podstawie znanych prędkości kół.
2. Wszystkie poniższe równania matematyczne zawierają poniższe zmienne
   - x_0, y_0 - współrzędne początkowe robota (pozycja początkowa),
   - $x(t), y(t)$ - współrzędne robota jako funkcje czasu,
   - $\theta_0, \theta(t)$ - początkowa orientacja robota i orientacja jako funkcja czasu,
   - v_R, v_L - prędkość odpowiednio prawego i lewego koła,
   - s_R, s_L - odległość pokonana odpowiednio przez kolo prawe i lewe (np.: obliczona na podstawie odczytu z enkoderów i danego obwodu koła),
   - b - odległość pomiędzy kołami (ich środkami),
   - r - promień skrętu robota,
   - t - czas w sekundach,
3. W każdej chwili czasu, współrzędne robota zmieniają się w zależności od prędkości robota i jego orientacji.
4. Przy poruszaniu się do przodu z pozycji (x, y, θ) o dystans d, nowe współrzędne robota będą opisane równaniem

$$\begin{bmatrix} x_{nowy} \\ y_{nowy} \\ \theta_{nowy} \end{bmatrix} = \begin{bmatrix} x + d\cos(\theta) \\ y + d\sin(\theta) \\ \theta \end{bmatrix}$$

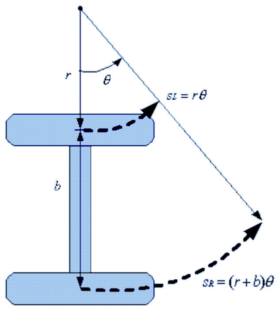
5. Przy obrocie robota z pozycji (x, y, θ) o kąt α, nowe współrzędne robota będą opisane równaniem

$$\begin{bmatrix} x_{nowy} \\ y_{nowy} \\ \theta_{nowy} \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta + \alpha \end{bmatrix}$$

6. Ponieważ używamy dwóch niezależnie sterowanych kół napędowych to przy wykonaniu skrętu o promieniu r każde z kół pokona następujące odległości

$$\begin{aligned} s_R &= (r + b)\theta \\ s_L &= r\theta \end{aligned}$$

   Powyższe wyrażenia odnoszą się do sytuacji przedstawionej na rysunku (czyli skrętu w lewą stronę)



7. W celu wyznaczenia trajektorii robota na podstawie znajomości prędkości jego kół, możemy posłużyć się następującymi równaniami

$$\begin{aligned} \frac{dx}{dt} =& v(t)\cos(\theta(t)) \\ \frac{dy}{dt} =& v(t)\sin(\theta(t)) \end{aligned} \tag{1}$$

gdzie $v(t)$ jest funkcją prędkości robota.

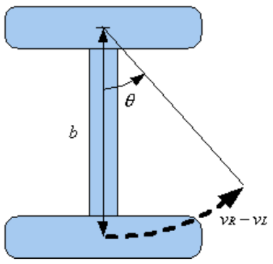
8. Zmiana orientacji robota, gdy mamy różne ale stałe prędkości kół, jest opisana następującym równaniem różniczkowym

$$\frac{d\theta}{dt} = \frac{(v_R - v_L)}{b}$$

Całkując powyższe równanie i biorąc pod uwagę początkową orientację robota $\theta(0) = \theta_0$ otrzymujemy równanie do obliczania pozycji robota

$$\theta(t) = \frac{(v_R - v_L)t}{b} + \theta_0 \tag{2}$$

Opisana tutaj sytuacja jest przedstawiona na poniższym rysunku



9. Znając orientację robota (czyli wzór (2)), możemy wyznaczyć jego pozycję na podstawie poniższych równań różniczkowych (przyjmujemy, że punkt odniesienia znajduje się w połowie odległości pomiędzy kołami, dlatego jego prędkość to $\frac{(v_R+v_L)}{2}$)

$$\begin{aligned} \frac{dx}{dt} =& \frac{(v_R + v_L)}{2}\cos(\theta(t)) \\ \frac{dx}{dt} =& \frac{(v_R + v_L)}{2}\sin(\theta(t)) \end{aligned} \tag{3}$$

10. Równanie (3) ma taką samą postać jak (1). Wykonując teraz operację całkowania i przyjmując, że początkowa pozycja robota to $x(0) = x_0$ i $y(0) = y_0$ otrzymujemy

$$\begin{aligned} x(t) =& x_0 + \frac{b(v_R + v_L)}{2(v_R - v_L)}\left(\sin\left(\frac{(v_R - v_L)t}{b} + \theta_0\right) - \sin(\theta_0)\right) \\ y(t) =& y_0 - \frac{b(v_R + v_L)}{2(v_R - v_L)}\left(\cos\left(\frac{(v_R - v_L)t}{b} + \theta_0\right) - \cos(\theta_0)\right) \end{aligned} \tag{4}$$

11. Korzystając z równania (4) mamy możliwość określenia aktualnej pozycji robota. Podstawiając za v_R i v_L odpowiednio s_R i s_L oraz opuszczając zmienną t (czyli czas, gdyż s_R i s_L przechowują aktualnie pokonany dystans przez koła a nie ich prędkości) możemy otrzymać dobre przybliżenie pozycji robota.

12. W programie komputerowym symulującym równania (4) musimy oddzielnie zaimplementować przypadek gdy $v_R = v_L$ lub $s_R = s_L$ (czyli gdy mianownik dąży do zera). W takim przypadku robot musi poruszać się po linii prostej.

13. Ponieważ w wybranych aplikacjach nie mamy dostępu do obliczeń zmiennoprzecinkowych lub moc obliczeniowa pokładowego sterownika jest ograniczona to wtedy do obliczeń możemy używać uproszczonych wyrażeń. Jednym z najczęściej stosowanych przybliżeń są wzory przedstawione poniżej

$$
\begin{aligned}
\overline{s} =& \frac{(s_R + s_L)}{2} \\
\theta =& \frac{(s_R - s_L)}{b} + \theta_0 \\
x =& \overline{s}\cos(\theta) + x_0 \\
y =& \overline{s}\sin(\theta) + y_0
\end{aligned} \tag{5}
$$

14. W powyższych rozważaniach przyjmowaliśmy, iż prędkości kół są stałe albo wolno zmienne. Jeśli jednak taka sytuacja nie ma miejsca, musimy dodatkowo rozważyć wpływ przyspieszenia. W najprostszym przypadku prędkości poszczególnych kół mozemy zapisać jako

$$
\begin{aligned}
v_R(t) =& \alpha_R t + w_R \\
v_L(t) =& \alpha_L t + w_L
\end{aligned} \tag{6}
$$

gdzie α_R, α_L są stałymi przyspieszeniami odpowiednio koła prawego i lewego, a w_R i w_L są wartościami początkowych prędkości.

15. Podstawiając wzór (6) w odpowiednie miejsca otrzymujemy

$$
\begin{aligned}
A =& \frac{(\alpha_R + \alpha_L)}{2} \\
B =& \frac{(w_R + w_L)}{2} \\
C =& \frac{(\alpha_R - \alpha_L)}{2b} \\
D =& \frac{(w_R - w_L)}{b}
\end{aligned}
$$

Teraz, wzory na polozenie i orientację robota będą miały następującą postać

$$
\begin{aligned}
\theta(t) =& Ct^2 + Dt + \theta_0 \\
\frac{dx}{dt} =& (At + B)\cos(Ct^2 + Dt + \theta_0) \\
\frac{dy}{dt} =& (At + B)\sin(Ct^2 + Dt + \theta_0)
\end{aligned} \tag{7}
$$

Oczywistym jest, że łatwo jest wyznaczyć orientację robota. Jednak analityczne rozwiązanie równań różniczkowych pozycji robota jest bardzo skomplikowane. Dlatego musimy rozwiązać je w sposób numeryczny. W programie komputerowym skorzystamy metod całkownia numerycznego (np. metody Simpsona).

Zadania do wykonania

1. Zbudować robota według wskazówek umieszczonych w instrukcji oraz przekazanych przez prowadzącego.

2. Napisać program do pomiaru charakterystyki silnika. Istotne jest aby zmierzyć prędkość obrotową silnika (liczba obrotów na minutę) w zależności do zadanej mocy silnika. Następnie wyznaczyć odległość pokonywaną przez robota w zadanej jednostce czasu (przyjąć, że średnica koła napędowego to 56 [mm]).

3. Sprawdzić czy dla zadanych, stałych prędkości kół, gdzie prędkość jednego koła różni się od prędkości drugiego koła, robot będzie poruszał się po okręgu o promieniu $(b/2)\frac{(v_R+v_L)}{v_R-v_L}$.

4. Napisać program do sterowania roborem tak aby poruszał się według danej trajektorii położenia lub prędkości.Przyjąć, że robot ma się poruszać po trajektorii której segmenty są tylko liniami prostymi i wycinkami koła.

5. Napisać program umożliwiający wizualizację aktualnej pozycji robota na ekranie komputera dla zadanych trajektorii prędkości kół.

6. Porównać wyniki określania pozycji robota na podstawie wzorów (4) i (5).

7. Podstawiając $v = \frac{v_L + v_R}{2}$ oraz $\omega = \frac{v_R - v_L}{b}$ model kinematyki robota o napędzie różnicowym można zapisać jako:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{8}$$

Przy założeniu, że v oraz ω są stałe, robot powinien poruszać się po okręgu o promieniu $R = v/\omega$ z okresem $T = 2\pi/\omega$.

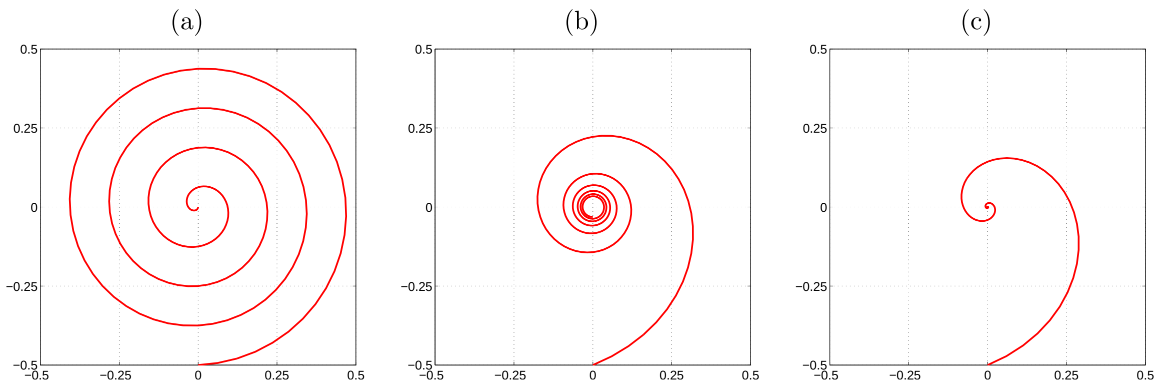
Korzystając ze środowiska Simulink zaimplementować model kinematyki robota i dobrać sterowania (v, ω) w taki sposób, żeby w okresie $T = 5$ s robot wykreślił okrąg o promieniu $R = 0,5$ m. Następnie dokonać korekt sterowań tak aby robot poruszał się po:

(a) spirali Archimedesa o równaniu (współrzędne biegunowe) $\theta = a/r$, $, r = x^2 + y^2$, $a = const$ (rys. 1(a)),

(b) spirali hiperbolicznej o równaniu $\theta = a \cdot r$ (rys. 1(b)),

(c) spirali logarytmicznej o równaniu $\theta = \ln(r/a)$ (rys. 1(c)).

Częstotliwość objazdu bieguna spirali w każdym przypadku powinna wynosić $0,2$ Hz. Następnie zaimplementować wybrany przypadek spirali (a)–(c) na robocie LEGO i sprawdzić praktycznie realizację zaplanowanej trajektorii.

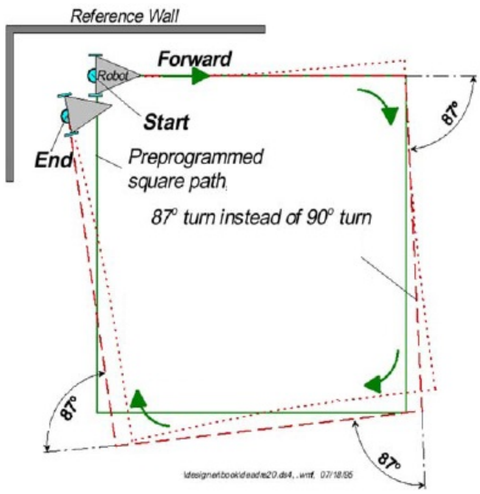


Rysunek 1: Trajektorie do zadania 8.

8. (*) Droga robota zakręca o 90^o po łuku okręgu. Promienie wewnętrznej i zewnętrznej krawędzi jezdni wynoszą odpowiednio 80 i 100. Korzystając z modelu kinematyki z poprzedniego zadania, zaprojektować sterownik, który dokona przejazdu pojazdu po trajektorii minimalizującej siłę odśrodkową w trakcie jazdy po zakręcie.

   - Wskazówka 1 (dla dociekliwych): przy założeniu stałej prędkości jazdy maksymalna siła odśrodkowa występuje w szczycie zakrętu,
   - Wskazówka 2 (dla mniej dociekliwych): minimalizacja maksymalnej siły odśrodkowej prowadzi do trajektorii hiperbolicznej, przechodzącej przez szczyt zakrętu.

9. W klasycznym teście Borensteina na błąd lokalizacji w odometrii robot wykonuje zaprogramowany przejazd po kwadracie o ustalonym promieniu (patrz rys. 2), przy czym mierzy się błędy odległości i kąta w każdym wierzchołku kwadratu. Błąd ma źródła deterministyczne (np. niedokładne wymiary robota) oraz stochastyczne (np. błędy pomiaru enkoderów robota). Teoretycznie kluczowy wpływ w odometrii ma pomiar azymutu robota, który wpływa nieliniowo na pomiar odległości. Sprawdzić czy i na ile użycie żyroskopu do dokładniejszego bezpośredniego pomiaru zmian kąta (zamiast pomiaru pośredniego z enkoderów, np. we wzorze 5) poprawi dokładność jazdy po zadanej trajektorii kwadratu. Bok kwadratu ustalić jako 1m, a błędy lokalizacji (czysta odometria vs odometria+żyroskop) przedstawić na wykresach.



Rysunek 2: Test błędu odometrii