

# Programowanie robota mobilnego E-puck w języku Python\*

## Zadania do wykonania

Joanna Ratajczak      Mirela Kaczmarek  
Aktualizował: Filip Dyba

### 1 Wymagania wstępne

- Zapoznanie się z zasadami bezpieczeństwa pracy na stanowisku laboratoryjnym.
- Zapoznanie się z treścią instrukcji *Programowanie robota mobilnego E-puck w języku Python*.
- Zapoznanie się z dokumentem *Modele kinematyki E-pucka*.

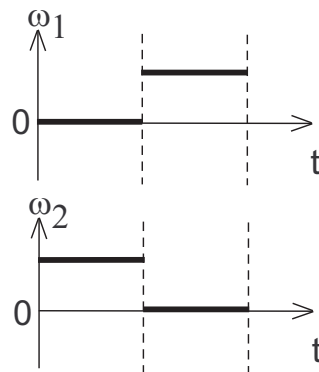
### 2 Zadania do wykonania

1. Zidentyfikować parametry geometryczne robota. Jaką interpretację mają parametry wejściowe funkcji `set_speed(lewy,prawy)`?
2. Sterując prędkościami kół przemieścić robota wzdłuż ścieżki schodkowej (przemieścić robota wzdłuż prostej o stały odcinek, wykonać obrót o  $90^\circ$  w prawo wokół głównej osi pionowej robota, następnie przemieścić robota wzdłuż prostej o taki sam odcinek i wykonać obrót o  $90^\circ$  w lewo).
3. Powtórzyć poprzednie zadanie wykonując obroty wokół osi pionowej odpowiedniego koła.

---

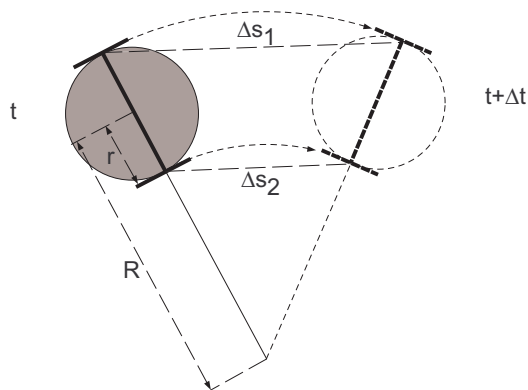
\*Data ostatniej modyfikacji: 27.02.2024. Ćwiczenie w ramach kursu Robotyka (2)

4. Zmodyfikować program kreślący krzywą schodkową w taki sposób, aby robot czekał przed wykonaniem obrotu 2 sekundy.
5. Napisać funkcję umożliwiającą zadawanie prędkości postępowej i obrotowej środka robota.
6. Sterując odpowiednio prędkościami zakreslić ósemkę.
7. Naturalnymi sterowaniami robota są prędkości kątowe  $\omega_1, \omega_2$  jego kół. Niech  $u = (\omega_1, \omega_2) \in R^2$  będzie pewną funkcją sterowań jak na rysunku 1. Sterowania te przeprowadzają robota z położenia początkowego **S** do końcowego **G** na płaszczyźnie ruchu.



Rysunek 1: Przebiegi prędkości kół robota E-puck w czasie

- Zadać sterowania z przeciwnym znakiem. Gdzie znalazł się robot? Jak należy interpretować sterowanie przeciwne  $-u$ ?
  - Jak powinno wyglądać sterowanie odwrotne, tzn. przeprowadzające robota z punktu **G** do punktu **S**?
8. Celem zadania jest napisanie programu, który przeprowadzi robota od punktu startowego **S** do końcowego **G** po zadanej ścieżce (np. takiej, jak na rysunku 3). Ścieżka składa się z kilku łuków o znanych promieniach krzywizny. Zadanie można rozwiązać bazując na poniższym przykładzie. Po wykonaniu zadania należy porównać wyniki doświadczalne z teoretycznymi oraz zastanowić się co wpływa na rozbieżności pomiędzy tymi wynikami.



Rysunek 2: Ruch E-pucka po łuku o promieniu  $R$

**Przykład** Zakreślenie okręgu o promieniu  $R = 5$  cm przez punkt leżący w środku symetrii robota.

*Rozwiązanie:*

Długość promienia podstawy robota  $r$  wynosi 25 mm. Chcemy znaleźć takie prędkości kół  $v_1$  oraz  $v_2$ , aby środek podstawy robota poruszał się po okręgu o promieniu  $R = 5$  cm. Na rysunku 2 przedstawiono położenie robota w chwili  $t$  oraz  $t + \Delta t$ . Korzystając z twierdzenia Talesa, dla nieskończenie krótkiej chwili czasu  $dt$  możemy napisać, że

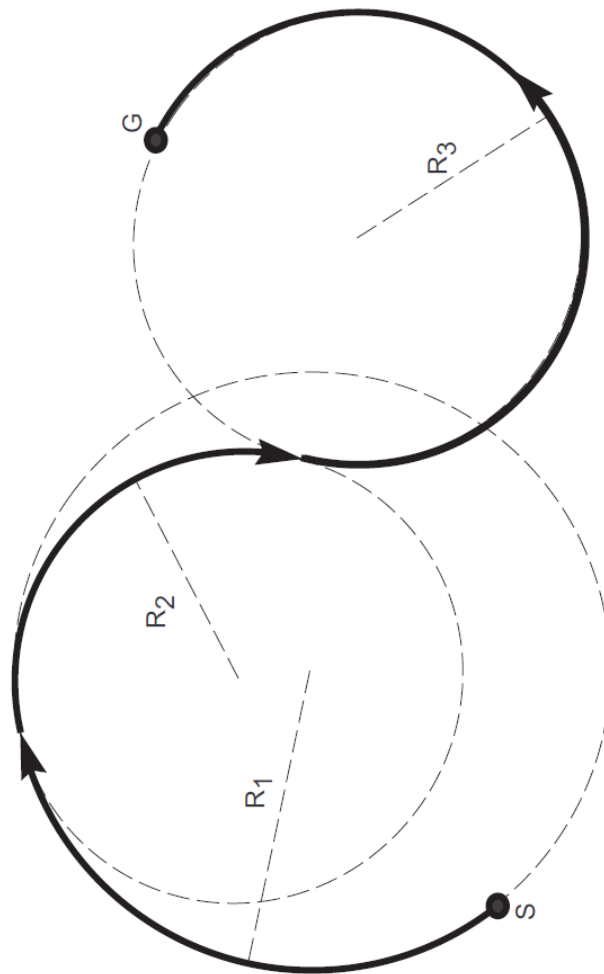
$$\frac{ds_1}{R + r} = \frac{ds_2}{R - r}, \quad (1)$$

gdzie  $ds_1$ ,  $ds_2$  oznaczają odpowiednio odcinki  $\Delta s_1$  oraz  $\Delta s_2$  w chwili  $t + \Delta t$ . Dzieląc równanie (1) przez  $dt$  (korzystając z tego, że  $\frac{ds}{dt} = v$ ) otrzymujemy

$$\frac{v_1}{v_2} = \frac{R + r}{R - r}. \quad (2)$$

Podstawiając do powyższego równania dane otrzymujemy  $\frac{v_1}{v_2} = \frac{3}{1}$ . Zatem należy zadbać o to, aby pomiędzy zadanymi wartościami prędkości kół spełniona była zależność  $v_1 = 3v_2$ .

9. Napisać program symulujący parkowanie prostopadłe (tyłem, przodem) oraz parkowanie równoległe. Porównać te manewry pod kątem trudności wykonania. Wyjaśnić przyczyny trudności.
10. Zweryfikować działanie czujników zbliżeniowych. Wyznaczyć charakterystykę tych czujników.
11. Napisać program, który umożliwi robotowi zatrzymanie się przed przeszkodą.



Rysunek 3: Przykładowa ścieżka

12. Napisać program, który umożliwi robotowi unikanie przeszkód podczas jazdy.
13. Napisać program, który umożliwi robotowi znalezienie wolnego miejsca oraz zaparkowanie.
14. Napisać program, który umożliwi robotowi autonomiczne poruszanie się po labiryncie.