

Imię i nazwisko	Nr indeksu	Kierunek	Wydział (skrót)	Data	Wersja sprawozdania
Dawid Królak Michał Matuszak	145383 145403	Informatyka	WliT	05.12.2020	1.0
Nr ćwiczenia	Tytuł ćwiczenia				
Ćw.3	Kinematyka i lokalizacja dwukołowego robota mobilnego				

0. Opis ćwiczenia.

Celem ćwiczenia jest wyprowadzenie modelu kinematyki dwukołowego robota mobilnego oraz poznanie budowy i zasady działania systemu pomiarowego odometrii inkrementalnej służącego do określania bieżącej pozycji i orientacji platformy robota w globalnym układzie współrzędnych.

1.1 Wyznaczenie macierzowego równania stanu robota dwukołowego (wektor stanu $q = [\varphi \ x \ y]^T$, sterowanie $u = [\omega \ v]^T$.

$$q = \begin{bmatrix} 1 & 0 \\ 0 & \cos(\varphi) \\ 0 & \sin(\varphi) \end{bmatrix} \cdot \begin{bmatrix} \omega \\ v \end{bmatrix}$$

```
def WyznaczMacierzStanu1(fi, u):
    m = [[1, 0],
          [0, math.cos(fi)],
          [0, math.sin(fi)]]

    q = np.matmul(m, u)
    return q
```

Zrzut ekranu 1: Kod źródłowy funkcji wyznaczającej macierz stanu ze sterowania $u = [\omega \ v]^T$.

1.2 Wyznaczenie macierzowego równania stanu robota dwukołowego (wektor stanu $q = [\varphi \ x \ y]^T$, sterowanie $u = [\omega_P \ \omega_L]^T$.

$$q = \begin{bmatrix} \frac{R}{D} & \frac{-R}{D} \\ \frac{R \cdot \cos(\varphi)}{2} & \frac{R \cdot \cos(\varphi)}{D} \\ \frac{R \cdot \sin(\varphi)}{2} & \frac{R \cdot \cos(\varphi)}{D} \end{bmatrix} \cdot \begin{bmatrix} \omega_P \\ \omega_L \end{bmatrix}$$

```
def WyznaczMacierzStanu2(fi, u):
    m = [[R/D, -R/D],
          [R*math.cos(fi)/2, R*math.cos(fi)/D],
          [R*math.sin(fi)/2, R*math.sin(fi)/D]]

    q = np.matmul(m, u)
    return q
```

Zrzut ekranu2: Kod źródłowy funkcji wyznaczającej macierz stanu ze sterowania $u = [w \ v]$.

2. Obliczanie pozycji i orientacji robota.

Wartości pozycji i orientacji obliczono wykorzystując dyskretyzację metodą Eulera wstecz. Startowe wartości orientacji i pozycji przyjęto jako 0, okres próbkowania $T_p = 0.01$, parametr $R = 0.021\text{m}$, parametr $D = 0.073\text{m}$.

```
T = [0] # Czas
Fi = [0] # Orientacje robota
X = [0] # Pozycje X robota
Y = [0] # Pozycje Y robota
W = [0] # Prędkość katowa robota
V = [0] # Prędkość liniowa robota
U1 = [] # Lista macierzy stanu u = [w v]
U2 = [] # Lista macierzy stanu u = [wp wl]

Tp = 0.01 # Okres próbkowania
R = 0.021 # Promień kół robota
D = 0.073 # Szerokość robota

with open("profile_prędkości_V2.txt") as file:
    lines = file.readlines()
    i = 0
    for line in lines:
        t, wl, wp = line.split("\t")

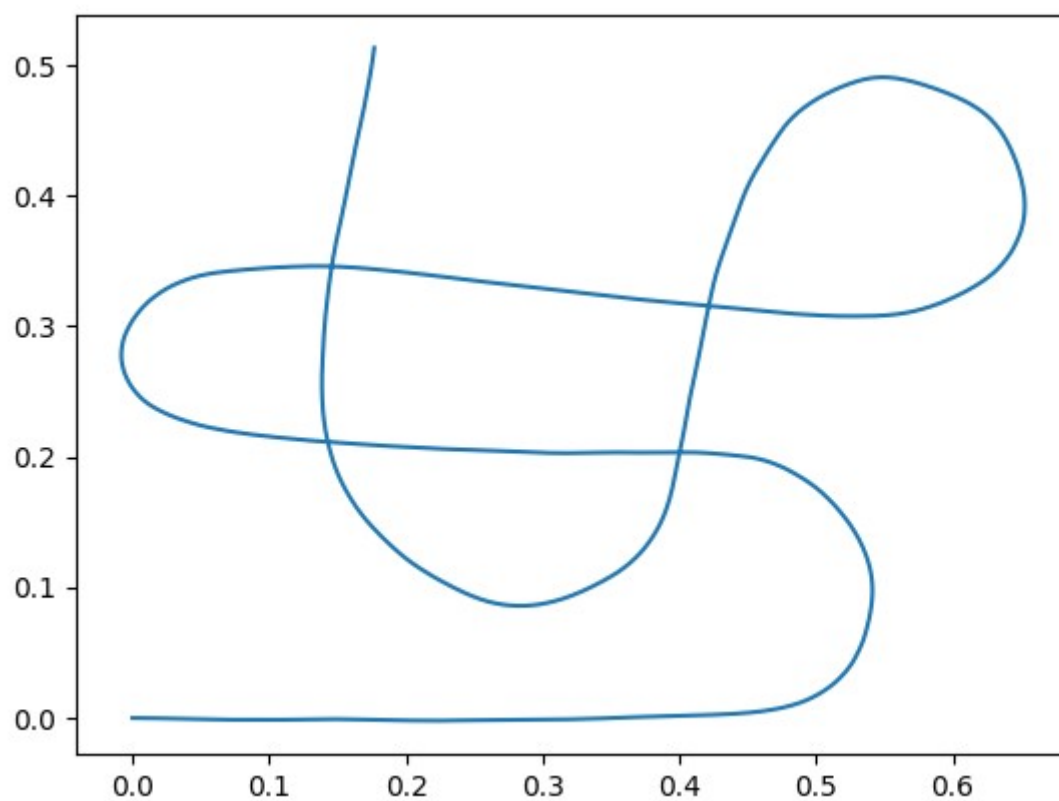
        T.append(t)
        W.append((float(wp)-float(wl))*R/D) # Wyprowadzenie zależności w(t) oraz v(t)
        V.append((float(wp)+float(wl))*R/2) # na podstawie danych

        U1.append([[W[-1]], [V[-1]]]) # Utworzenie listy kolejnych
        U2.append([[float(wp)], [float(wl)]] # macierzy sterowań

    # Ustalenie funkcji fi(t), x(t), y(t)
    for i in range(1, len(lines)):
        Fi.append(Fi[i - 1] + Tp * (W[i]))
        X.append(X[i - 1] + Tp * (V[i]*math.cos(Fi[i])))
        Y.append(Y[i - 1] + Tp * (V[i]*math.sin(Fi[i])))
```

Zrzut ekranu 3: Kod obliczający wartości pozycji i orientacji oraz prędkości kątowych i liniowych robota w zależności od czasu

3. Wykres przemieszczenia robota oraz pozycja i orientacja końcowa.



Zrzut ekranu 4: Rysunek trajektorii przemieszczania się robota

Orientacja i pozycja końcowa robota:

$\varphi = 1.5118217299787955$

$x = 0.176832378573989$

$y = 0.5133492914017785$