

POLITECHNIKA POZNAŃSKA

Modelowanie i sterowanie robotów



Raport

Projekt robota w konfiguracji RPR | — —

Adam Nawrocki

Maciej Mak

Grupa L10

Poznań 2022

Spis treści

1. Wstęp	3
2. Zadania	4
2.1. Projekt manipulatora	4
2.1.1. Budowa manipulatora	4
2.1.2. Program w języku Python	6
2.2. Kinematyka prosta	9
2.3. Kinematyka odwrotna	10
2.4. Kinematyka prędkości	11
2.5. Równania dynamiki	13
3. Wnioski	15
Bibliografia	16

1. Wstęp

Projekt obejmuje utworzenie robota w konfiguracji RPR | – –. W raporcie zostały uwzględnione aspekty obliczeniowe takie jak kinematyka prosta, kinematyka odwrotna, kinematyka prędkości, równania Lagrange’a II rodzaju.

W raporcie operujemy na wartościach symbolicznych takich jak np. l_1 , jednak do obliczeń w języku Python[1] zostały użyte rzeczywiste wartości zgodnie z modelem w programie Inventor[2].

Dodatkowo zostały określone takie rzeczy jak przestrzeń robocza manipulatora, tabele notacji Denavit-Hartenberga[3], uproszczony model manipulatora na wykresie Robotics ToolBox[4].

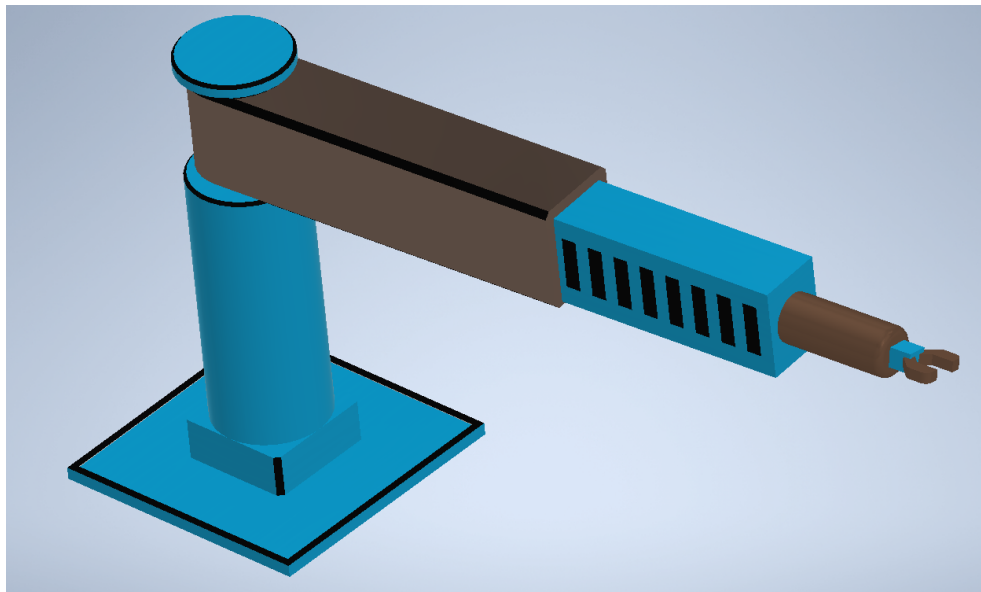
Projekt jest dostępny jako repozytorium na serwisie GitHub[5] pod linkiem <https://github.com/maciejm2517/Projekt-Robot-RPR>

Ustawienia układu stron oraz bibliografii w języku L^AT_EX zostały zainspirowane formatką Pana mgr inż. Bogdana Fabiańskiego[6], od którego otrzymaliśmy plik szablonowy na zajęciach Techniki Cyfrowej.

2. Zadania

2.1. Projekt manipulatora

2.1.1. Budowa manipulatora

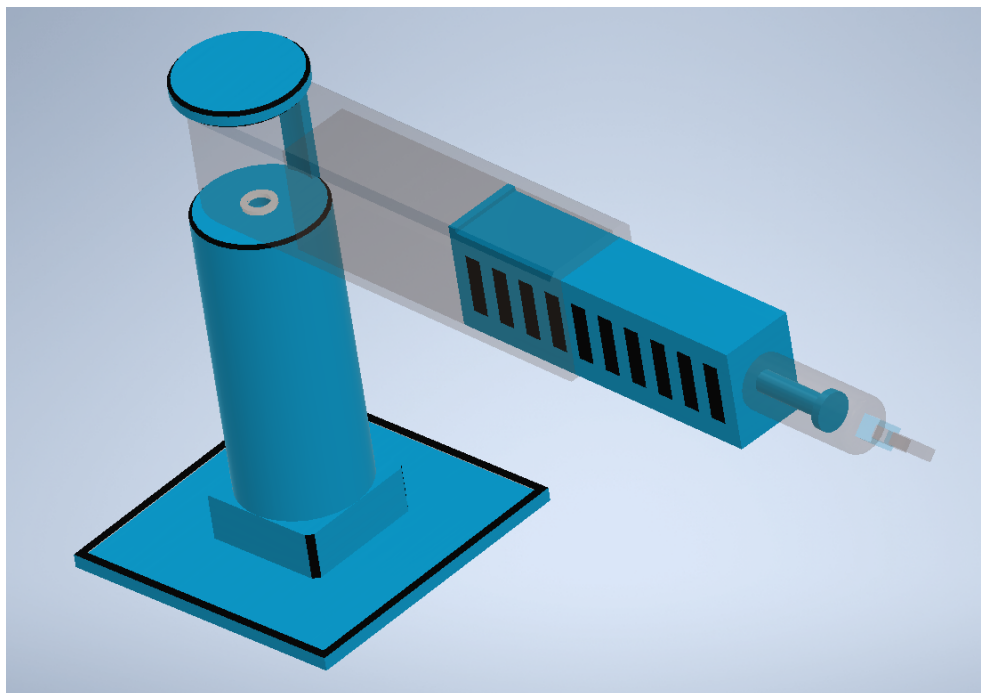


Rysunek 2.1. Projekt manipulatora

Manipulator został zaprojektowany w programie Inventor. Jego konfiguracja to RPR
| — —.

Podczas konstrukcji robota uwzględniliśmy mechaniczne aspekty konstrukcji takie jak:

- ograniczenie stopni swobody
- zabezpieczenia mechaniczne



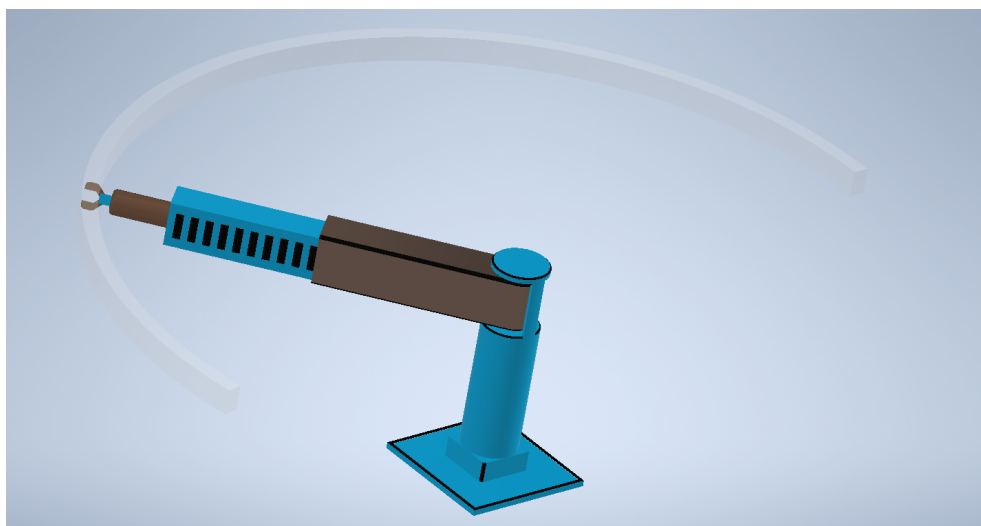
Rysunek 2.2. Konstrukcja mechaniczna manipulatora

Robot posiada ograniczenia w pierwszych dwóch przegubach

- Maksymalne wychylenie kąta θ_1 wynosi 195°
- Maksymalne wysunięcie długości d_1 wynosi 38cm

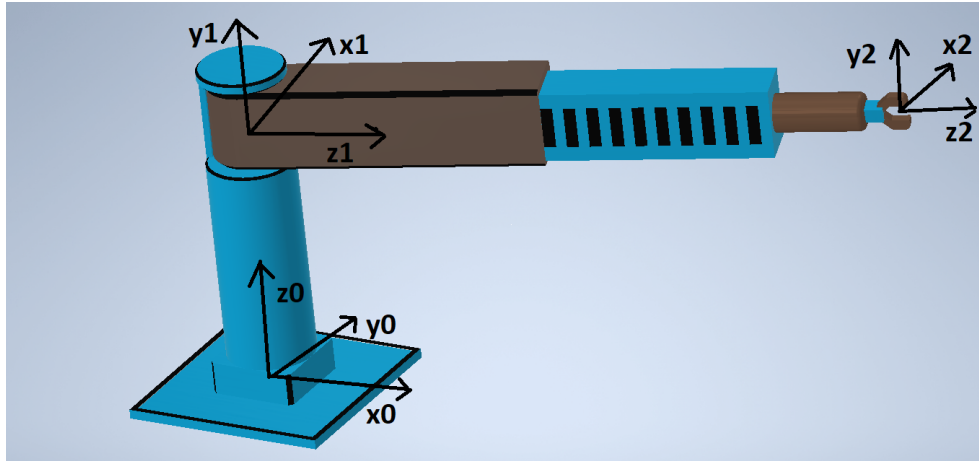
Ostatni przegub jest zaprojektowany na pełen obrót 360° .

Do pierwszego przegubu zostały dodane podkładki, mające na celu polepszenie stabilności obrotu ramienia robota.



Rysunek 2.3. Przestrzeń robocza manipulatora

W naszym modelu wybraliśmy ułożenie końcówki TCP równoległe z ostatnim przegubem przesuwным. Oznacza to, że ten przegub nie odpowiada za pozycję końcówki, a za jej orientację. W związku z tym nasz manipulator może zostać zaprojektowany np. do przykręcania śrub.



Rysunek 2.4. Układy współrzędnych w przegubach manipulatora

2.1.2. Program w języku Python

Do zamodelowania robota skorzystaliśmy z biblioteki RoboticToolBox. Kod znajduje się w funkcji projekt() oraz składa się z trzech etapów:

1. inicjacji zmiennych, utworzeniu robota oraz pokazaniu go na interaktywnym wykresie
2. obliczeniu kinematyki prostej
3. obliczeniu kinematyki prędkości

```

1 import roboticstoolbox as rtb
2 from spatialmath.base.symbolic import *
3 from roboticstoolbox.tools.trajectory import *
4
5
6
7 def projekt():
8
9     #inicjacja
10
11     l1=0.4925
12     l2=0.5
13     alpha1=np.pi/2
14     offset1=np.pi/2
15     theta1_min=0
16     theta1_max=195/360*2*np.pi
17     d2_min=0
18     d2_max=0.38

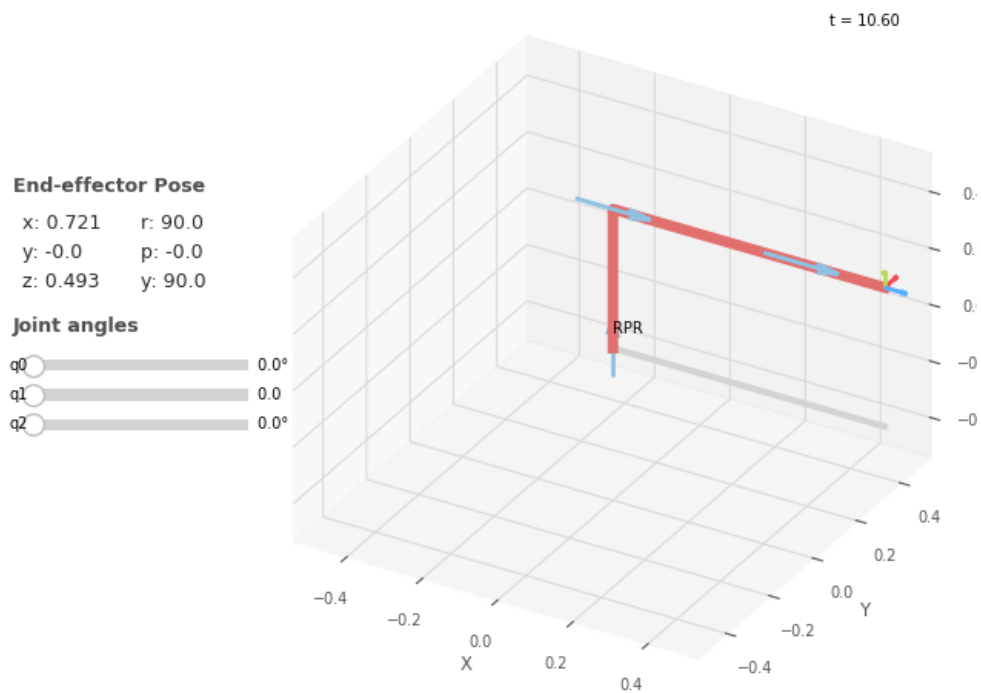
```

```
13=0.221314
19 theta3_min=0
   theta3_max= 2 * np.pi
21
23 q_tmp = [symbol('theta1'), symbol('theta2'), symbol('theta3')]
25
27 #obliczenia na symbolach
29
31 robot=rtb.DHRobot(
33     [
35         rtb.RevoluteDH(d=symbol('l1'), alpha=pi()/2
37             , offset=pi()/2),
39         rtb.PrismaticDH(offset=symbol('l2')),
41         rtb.RevoluteDH(d=symbol('l3'))
43     ], name="RPR")
45
47
49 robot=rtb.DHRobot(
51     [
53         rtb.RevoluteDH(d=l1, alpha=alpha1, offset=offset1,
55             qlim=[theta1_min, theta1_max]),
57         rtb.PrismaticDH(offset=l2, qlim=[d2_min,d2_max]),
59         rtb.RevoluteDH(qlim=[theta3_min, theta3_max], d=l3),
61     ], name="RPR")
63
65 print(robot)
67 robot.teach(robot.q)
69
71 #kinematyka prosta
73
75 T=robot.fkine(q_tmp)
77 print(T)
79
81 #kinematyka predkosci
83 J=simplify(robot.jacob0(q_tmp))
85 print(J)
87
89 #kinematyka odwrotna - na kartce
91 #dynamika - na kartce
93
95 if __name__ == '__main__':
97     projekt()
```

DHRobot: RPR, 3 joints (RPR), dynamics, standard DH parameters

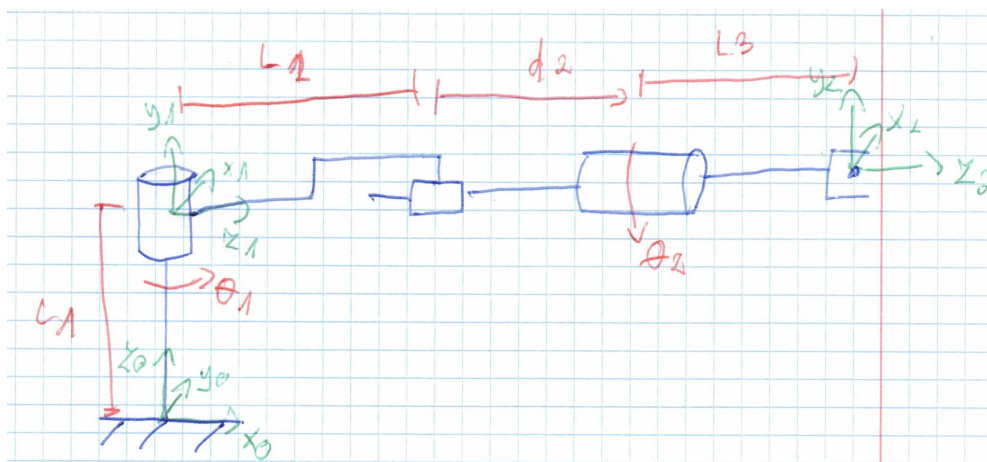
θ_j	d_j	a_j	φ_j	q^-	q^+
$q1 + 90^\circ$	0.4925	0	90.0°	0.0°	200.0°
0.0°	$q2 + 0.5$	0	0.0°	0.0	0.38
$q3$	0.2213	0	0.0°	0.0°	360.0°

Rysunek 2.5. Tablica DH z konsoli programu w języku Python



Rysunek 2.6. Interaktywny układ robota dla programu w języku Python

2.2. Kinematyka prosta



Rysunek 2.7. Schemat robota

i	θ_i	d_i	a_i	α_i
1	$\theta_1 + 90$	l_1	0	90
2	0	$l_2 + d_2$	0	0
3	θ_3	l_3	0	0

Rysunek 2.8. Tabela DH

$$A_1^0 = \begin{bmatrix} -s_1 & 0 & c_1 & 0 \\ c_1 & 0 & s_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

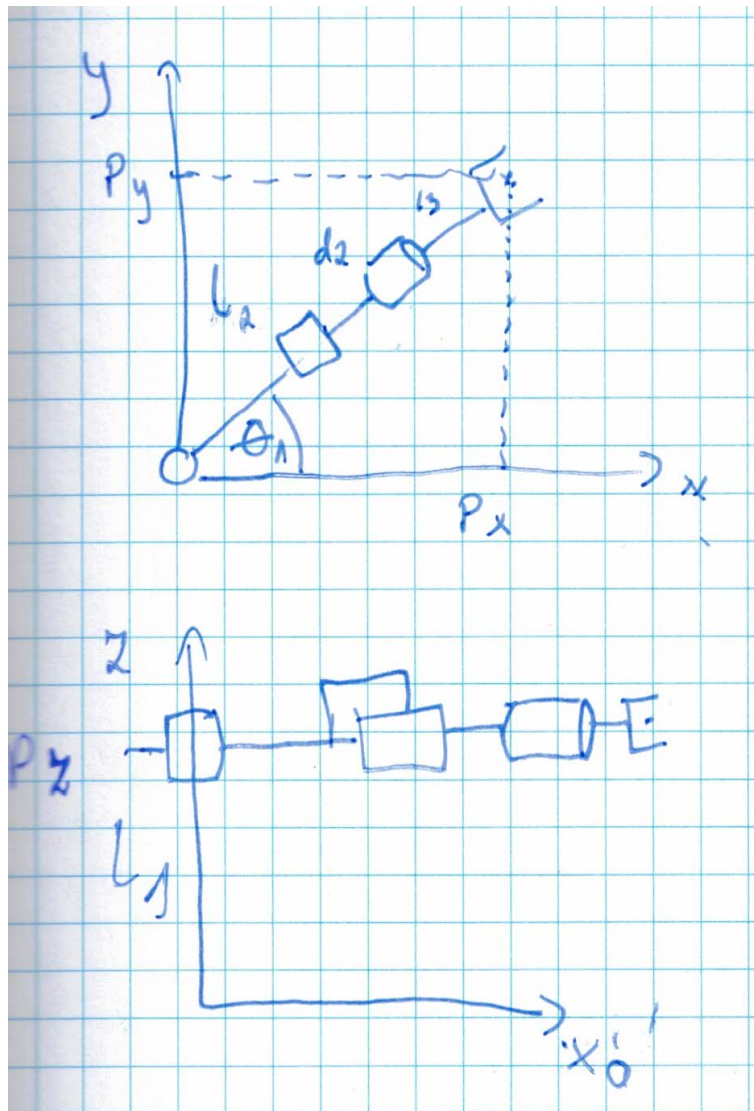
$$A_2^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_2 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & l_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^0 = \begin{bmatrix} -c_3 s_1 & s_3 s_1 & c_1 & c_1(l_2 + d_2 + l_3) \\ c_1 c_3 & -c_3 c_1 & s_1 & s_1(l_2 + d_2 + l_3) \\ s_3 & c_3 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rysunek 2.9. Macierz transformacji robota

2.3. Kinematyka odwrotna



Rysunek 2.10. Widok z góry i boku robota wykorzystane podczas obliczania kinematyki odwrotnej

$$q = \begin{bmatrix} \theta_1 \\ d_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} \text{atan2}(\cos(\sqrt{p_x^2 + p_y^2}), \sin(\sqrt{p_x^2 + p_y^2})) \\ \sqrt{p_x^2 + p_y^2} - l_2 - l_3 \\ \theta_3 \end{bmatrix}$$

Rysunek 2.11. Macierz transformacji robota

θ_3 nie wpływa na zmianę położenia końcówki (jedynie na jej orientację względem układu bazowego).

2.4. Kinematyka prędkości

$$J_1 = \begin{bmatrix} \vec{z}_0 \times \overrightarrow{(0_n - 0_0)} \\ \vec{z}_0 \end{bmatrix}$$

$$z_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$0_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$O_n = \begin{bmatrix} l_3 \times c_1 + c_1 \times (l_2 + d_2) \\ l_3 \times s_1 + s_1 \times (l_2 + d_2) \\ l_1 \end{bmatrix}$$

$$J_1 = \begin{bmatrix} -l_3 \times s_1 - s_1 \times (l_2 + d_2) \\ l_3 \times c_1 + c_1 \times (l_2 + d_2) \\ 0 \end{bmatrix}$$

$$J_2 = \begin{bmatrix} \vec{z}_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} c1 \\ s1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$J_3 = \begin{bmatrix} \overrightarrow{z_2} \times \overrightarrow{0_n - 0_2} \\ \overrightarrow{z_2} \end{bmatrix}$$

$$\overrightarrow{0_n - 0_2} = \begin{bmatrix} c_1 \times (l_2 + d_2) \\ s_1 \times (l_2 + d_2) \\ l_1 \end{bmatrix}$$

$$\overrightarrow{z_2} = \begin{bmatrix} c_1 \\ s_1 \\ 0 \end{bmatrix}$$

$$J_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ c1 \\ s1 \\ 0 \end{bmatrix}$$

$$J = \begin{bmatrix} -l_3 \times s_1 - s_1 \times (l_2 + d_2) & c_1 & 0 \\ l_3 \times c_1 + c_1 \times (l_2 + d_2) & s_1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & c1 \\ 0 & 0 & s1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} -l_3 \times s_1 - s_1 \times (l_2 + d_2) & c_1 & 0 \\ l_3 \times c_1 + c_1 \times (l_2 + d_2) & s_1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & c1 \\ 0 & 0 & s1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{d}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

Rysunek 2.12. Wektory prędkości linowej i kątowej końcówki względem układu zerowego

2.5. Równania dynamiki

$$J_{OC1} = \begin{bmatrix} -\frac{1}{2}s_1l_2 \\ \frac{1}{2}c_1l_2 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \begin{matrix} \vec{0} & \vec{0} \end{matrix}$$

$$J_{OC2} = \begin{bmatrix} -\frac{1}{2}s_1l_2 & c_1 \\ \frac{1}{2}c_1l_2 & s_1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \quad \begin{matrix} \vec{0} \end{matrix}$$

$$J_{OC3} = \begin{bmatrix} -\frac{1}{2}s_1l_2 & c_1 & 0 \\ \frac{1}{2}c_1l_2 & s_1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & s_1 \\ 0 & 0 & c_1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$D_1 = \begin{bmatrix} -\frac{1}{4}(l_2^2m_2) + I_{1YY} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$D_2 = \begin{bmatrix} \frac{1}{4}(l_2^2c_1^2m_2 + l_2^2s_1^2m_2) + I_{2YY} & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$D_3 = \begin{bmatrix} \frac{1}{4}(l_2^2c_1^2m_3 + l_2^2s_1^2m_3) + s_3^2I_{3XX} + s_3^2I_{3YY} & 0 & c_3s_3I_{3XX}(c_1^2 - s_1^2) + i_{3YY}(c_3s_3s_1^2 - c_3^2c_1^2) \\ 0 & m_3 & 0 \\ I_{3XX}(c_3s_3c_1^2 - c_3s_3s_1^2) + I_{3YY}(c_3s_3s_1^2 - c_3^2c_1^2) & 0 & I_{XX}c_3^2(c_1^2 - s_1^2)^2 + I_{YY}(c_3c_1^2 - s_3s_1^2)^2 + I_{ZZ}(4s_1^2c_1^2) \end{bmatrix}$$

$$D = D_1 + D_2 + D_3$$

$$c_{111} = 1/2(-1/2l_2^2c_1s_1m_2 - 1/2l_2^2c_1s_1m_2 + 1/2l_2^2s_1c_1m_2) + 1/2(-1/2l_2^2c_1s_1m_3 - 1/2l_2^2c_1s_1m_3 + 1/2l_2^2s_1c_1m_3)$$

$$c_{113} = I_{3XX}(c_3s_3 - 2c_1s_1 - c_3s_32s_1c_1) + I_{3YY}(c_3s_32s_1c_1 + 2c_3^2c_1s_1) - s_3c_3I_{3XX} - s_3c_3I_{3YY}$$

$$c_{133} = 1/2(I_{3XX}8c_1s_1(s_1^2-s_1^2) + I_{3YY}(4(c_3+s_3)c_1s_1(s_3s_1^2-c_3s_1^2)) + I_{3ZZ}(-8c_1s_1(s_1^2-c_1^2)))$$

$$c_{333} = 1/2(I_{3XX}(-2)s_3c_3(c_1^2-s_1^2)^2) + I_{3YY}(-s_3c_1^2-c_3s_1^2)^2)$$

$$c_{313} = c_{133}$$

$$c_{311} = c_{131}$$

$$c_{131} = c_3s_3(I_{3XX} + I_{3YY})$$

$$c_{331} = I_{3XX}(s_3^2-c_3^2)(c_1^2-s_1^2) + I_{3YY}(-s_1^2(s_3^2-c_3^2) - c_3s_3c_1^2) - c_{133}$$

$$C = \left[\begin{array}{c} c_{111}\dot{\theta}_1^2 + c_{131}\dot{\theta}_1\dot{\theta}_3 + c_{311}\dot{\theta}_1\dot{\theta}_3 \\ 0 \\ c_{113}\dot{\theta}_1^2 + c_{133}\dot{\theta}_1\dot{\theta}_3 + c_{333}\dot{\theta}_3^2 + c_{313}\dot{\theta}_1\dot{\theta}_3 \end{array} \right]$$

$$E_p = E_{p1} + E_{p2} + E_{p3} = m_1gl_1 + m_2gl_1 + m_3gl_1$$

$$G=\frac{\delta E_p}{\delta q_k}=\begin{bmatrix}0\\0\\0\end{bmatrix}$$

$$D(q)q + C(q,\dot{q}) + G(q) = \tau$$

3. Wnioski

Budowa robota w praktyce różni się znacząco od obliczeń prowadzonych na kartce. Trzeba uwzględnić różne opory, mechanizmy, przeszkody w wykonywaniu modelu. Mogliśmy tego również doświadczyć na laboratoriach z tego przedmiotu, gdzie pracowaliśmy z prawdziwymi maszynami **KUKA** czy **STÄUBLI**

Nasz projekt również wymagał od nas uwzględnienia takich przeciwności jak rzeczywista konstrukcja połączeń czy sposoby blokady mechanizmów, aby robot mógł zostać złożony w rzeczywistości.

Możemy również zauważyć, że parametry takie jak przestrzeń robocza różnią się od obliczeń prowadzonych na kartce - teoretycznie przy naszej konfiguracji robota, przestrzenią roboczą powinien być dysk, jednak w rzeczywistości jest nią spłaszczony walec (bez środka). Wynika to z faktu że końcówka manipulatora nie jest punktem w przestrzeni, a ma swój rozstaw końcówek.

Bibliografia

- [1] "Python." [Online]. Available: <https://www.python.org/>
- [2] "Inventor." [Online]. Available: <https://www.autodesk.pl/products/inventor/overview>
- [3] "Denavit–hartenberg." [Online]. Available: https://en.wikipedia.org/wiki/Denavit%E2%80%93Hartenberg_parameters
- [4] "Robotics toolbox." [Online]. Available: <https://petercorke.com/toolboxes/robotics-toolbox/>
- [5] "Github." [Online]. Available: <https://github.com/>
- [6] "Bogdan fabiański." [Online]. Available: <https://sin.put.poznan.pl/people/details/bogdan.fabianski>