

Raport z projektu

**MECHANIZM ROBOTA
O 4 STOPNIACH SWOBODY**

Roboty usługowe

Kamil Warchoł

400249

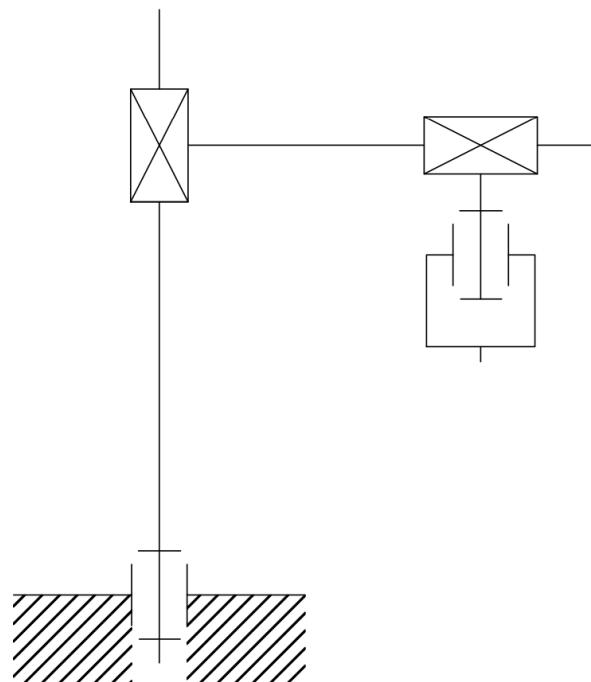


Spis treści

Schemat kinematyczny robota	1
Obliczanie ruchliwości	2
State of art	4
Założenia konstrukcyjne	8
Kinematyka prosta	9
Kinematyka odwrotna	17
Model CAD	24
Rysunki wykonawcze	27
Rysunek złożeniowy	30
Źródła	32

Schemat kinematyczny robota

Poniżej zamieszczony został schemat kinematyczny robota, narysowany w programie Autodesk AutoCad.



Rys. 1 Schemat kinematyczny

Obliczanie ruchliwości

Definicja ruchliwości

Ruchliwość mechanizmu nazywamy liczbę więzów, które należałyby narzucić na człony, aby łańcuch kinematyczny był nieruchomy względem podstawy.

Ruchliwość jest równa liczbie niezależnych napędów, które należy przyłożyć do łańcucha kinetycznego, aby wykonał on ściśle określony ruch, czyli był mechanizmem.

Ruchliwość **w** wyraża się wzorem:

$$w = 6n - \sum_{i=1}^5 i * p_i \text{ gdzie:}$$

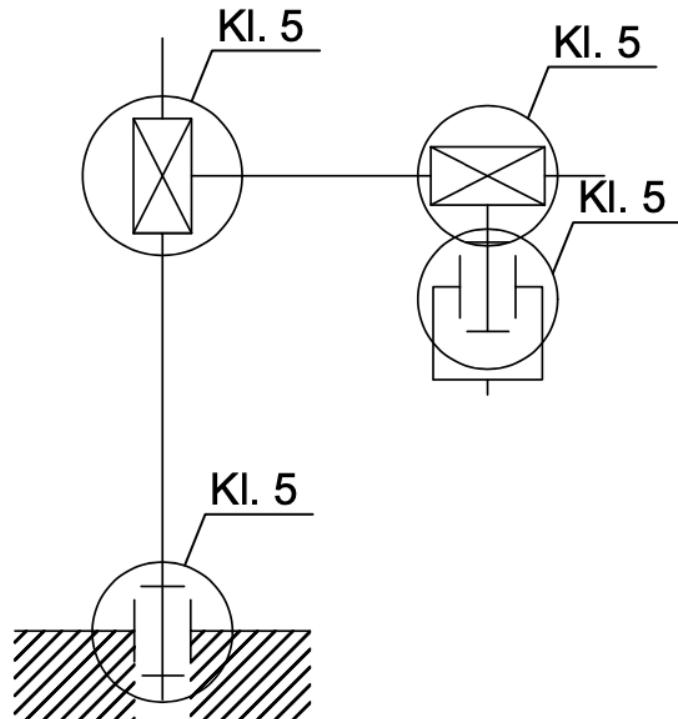
6n – liczba stopni swobody przestrzennego układu swobodnego

n – liczba członów ruchomych

i – klasa par występujących w łańcuchu kinematycznym

p₄, p₅ – liczba par kinematycznych klasy 4 i 5

Schemat z zaznaczonymi klasami par kinematycznych



Rys. 2 Schemat kinematyczny ze stopniami swobody

Obliczenie ruchliwości $p_5 = 4$ - liczba par kinematycznych klasy 5 $n = 4$ - liczba członów ruchomych łańcucha kinetycznego

$$w = 6n - \sum_{i=1}^5 i * p_i$$

Po podstawieniu dostajemy:

$$w = 6 * 4 - 5 * 4 = 4$$

State of art

Przeznaczenie mechanizmu

Proponowany przeze mnie mechanizm będzie przeznaczony do procesu lutowania na liniach produkcyjnych.

Roboty o podobnym przeznaczeniu, które są przedstawione na poniższych grafikach nie mają możliwości obrotu manipulatora (manipulator rusza się tylko w dwóch osiach, ruch trzeciej osi jest możliwy dzięki ruchomemu stołowi). Moje rozwiązanie pozwala na obrót całego manipulatora jak i samego chwytaka. Umożliwi to dostęp grotu lutownicy do wszelkich skomplikowanych i trudno dostępnych miejsc.

Dzięki możliwości obrotu całego robota, możemy też w jednym cyklu pracy obsłużyć kilka stanowisk roboczych, co jest niemożliwe w znalezionych rozwiązaniach. Pozwoli to znacznie przyspieszyć i uprościć produkcję.

Konstrukcyjnie – moje rozwiązanie będzie podobne do zaprezentowanego w punkcie 4 poniżej. Brak konieczności przenoszenia obciążen pozwoli na wydłużenie ramienia oraz zoptymalizowanie wagi całego robota. Poprawi to jego mobilność oraz ułatwi transport.

Wadą jest stopień skomplikowania mechanizmu w odniesieniu do innych lutowniczych robotów, a co za tym idzie na pewno jego cena.

Przegląd rozwiązań

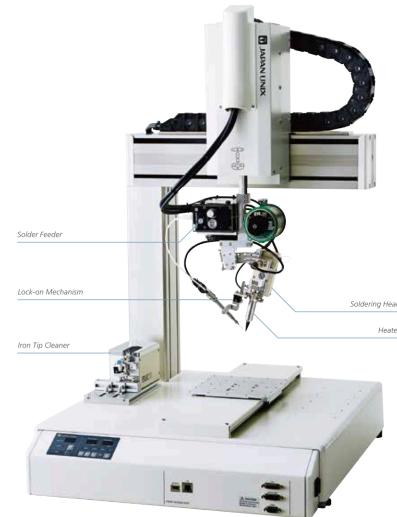
1. Japan Unix – DF304S [3]



UNIX-DF304S

Przestrzeń robocza: 300mm×275mm

2. Japan Unix – DF404S [3]



UNIX-DF404S

Przestrzeń robocza: 400mm×365mm

3. Apollo - J-CAT 330 COMET [2]



Przestrzeń robocza: 300mm×320 mm

4. Wobit - TR300-200-MLxx[1]



Przestrzeń robocza: okrąg o promieniu 300 mm

5.Japan Unix - SOLDER MEISTER UNIX-700FH [4]



Przestrzeń robocza: Charakterystyczna dla robota typu SCARA o długościach ramion 350mm, 450mm i 550mm

6.TBS – T- 6331RM [5]



Przestrzeń robocza: 580mm×280 mm x 100mm

Założenia konstrukcyjne

Przechodząc do konkretów:

-Robot będzie obsługiwał 4 stanowiska lutownicze – każde o wymiarach 180x180

-Podstawą konstrukcji będzie platforma o wymiarach 700x700x200, w której centrum znajdzie się stół obrotowy będący bazą manipulatora,

-Na frontowej ścianie podstawy zostanie umieszczony panel operatorski umożliwiający kalibracje, ręczne kontrolowanie oraz diagnostykę robota,

-Stanowiska lutownicze zostaną umieszczone na przekątnych podstawy, umożliwiając swobodny dostęp końcówce lutowniczej,

-Ruch stołu obrotowego realizowany będzie za pomocą silnika krokkowego NEMA23, co pozwoli uzyskać zakładaną dokładność – jest to priorytetowa sprawa w tego typu robocie,

-Ruch w pionie zostanie zrealizowany za pomocą popularnego rozwiązania np. z drukarek 3D, wykorzystującego gwintowany zewnętrznie pręt oraz wewnętrznie specjalne nakładki podtrzymujące ramię robota. Obrót pręta również obsługiwany jest przez silnik krokkowy NEMA23. Aby utrzymać płynność poruszania zostanie zamocowany profil z wyciętą prowadnicą na ruchome nakładki oraz dwie zewnętrzne prowadnice dodatkowo stabilizujące drgania.

- Ruch w poziomie będzie odbywał się w identyczny sposób co ruch w pionie – tym razem nakładka przesuwająca się po pręcie będzie zespojona z platformą podtrzymującą ostatni element konstrukcji

- Ostatni ruch jest zdublowanym obrotem względem pionowej osi – to właśnie on wpływa na wszechstronność w ustawieniu grotu lutownicy mojego robota. Ten ruch ponownie zrealizowany jest z użyciem silnika krokkowego – NEMA23.

Kinematyka prosta

Notacja Denavita-Hartenberga

Obliczenia kinematyki prostej będą wykonywane przy pomocy notacji Denavita-Hartenberga, pozwalającą bezproblemowo opisać każdy ruch otwartego łańcucha kinematycznego. Każde pojedyncze przekształcenie jednorodne przedstawiamy jako iloczyn czterech przekształceń podstawowych:

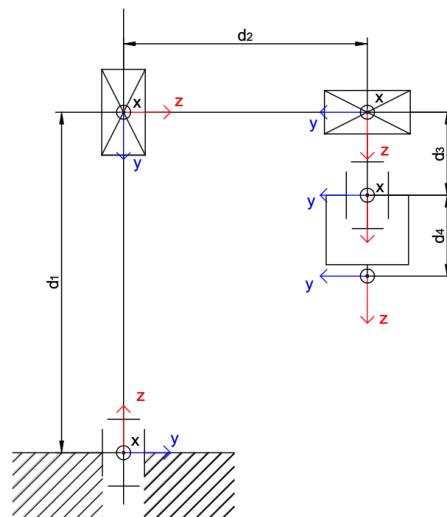
$$A_i = \text{Rot}_{z,\theta_i} \text{Trans}_{z,d_i} \text{Trans}_{x,a_i} \text{Rot}_{x,\alpha_i}$$

Czyli kolejno – rotacja względem osi z o kąt θ , translacja względem osi z o długość d , translacja względem osi x o długość a oraz rotacja względem osi x o kąt α

Przekształcenia te opisujemy za pomocą czterech parametrów:

a_i : długość członu, α_i : skręcenie członu, d_i : odsunięcie przegubu, θ_i : kąt przegubu

Układ	θ_i	d_i	a_i	α_i
1	$\theta_1 var$	$d_1 var$	0	-90°
2	0	$d_2 var$	0	-90°
3	0	d_3	0	0
4	$\theta_r var$	d_4	0	0



Dla układu 1:

$$A_1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Dla układu 2:

$$A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Dla układu 3:

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Dla układu 4:

$$A_4 = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 0 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 0 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{4,0} = A_1 * A_2 * A_3 * A_4$$

$$T_{4,0} = \begin{bmatrix} \cos(\theta_1 - \theta_4) & \sin(\theta_1 - \theta_4) & 0 & d_2 \sin \theta_1 \\ \sin(\theta_1 - \theta_4) & -\cos(\theta_1 - \theta_4) & 0 & -d_2 \cos \theta_1 \\ 0 & 0 & -1 & d_1 - d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Sprawdzenie poprawności obliczeń

Aby sprawdzić poprawność obliczeń przyjmę bardzo proste wartości kątów oraz miary długości ramion. Pozwoli mi to na podstawie macierzy $T_{4,0}$ w łatwy sposób ocenić, czy obliczenia zostały wykonane poprawnie.

Obliczenia zostaną przeprowadzone przy użyciu środowiska MATLAB za pomocą następującego algorytmu:

```
%% Kąty rotacji
teta1 = pi/2;
teta2 = -pi/2;
teta3 = -pi/2;
teta4 = pi/2;
%% Długości członów/wielkości translacji
d1 = 2;
d2 = 1;
d3 = 1;
d4 = 1;
%% Macierze przekształceń jednorodnych
a1rotz = [cos(teta1) -sin(teta1) 0 0;
           sin(teta1) cos(teta1) 0 0;
           0 0 1 0;
           0 0 0 1];
a1transz = [1 0 0 0;
             0 1 0 0;
             0 0 1 d1;
             0 0 0 1];
a1rotx = [1 0 0 0;
           0 0 -1 0;
           0 1 0 0;
           0 0 0 1];
a2transz = [1 0 0 0;
             0 1 0 0;
             0 0 1 d2;
             0 0 0 1];
a2rotx = [1 0 0 0;
           0 0 -1 0;
           0 1 0 0;
           0 0 0 1];
a3transz = [1 0 0 0;
             0 1 0 0;
             0 0 1 d3;
             0 0 0 1];
a4rotz = [cos(teta4) -sin(teta4) 0 0;
           sin(teta4) cos(teta4) 0 0;
           0 0 1 0;
           0 0 0 1];
```

```

a4transz = [ 1  0  0  0;
             0  1  0  0;
             0  0  1 d4;
             0  0  0  1];
%% Obliczenia macierzy Ai
a1 = a1rotz * altransz * alrotx;
a2 = a2transz * a2rotx;
a3 = a3transz;
a4 = a4rotz * a4transz;
%% Macierz transformacji T40
T40 = a1 * a2 * a3 * a4;

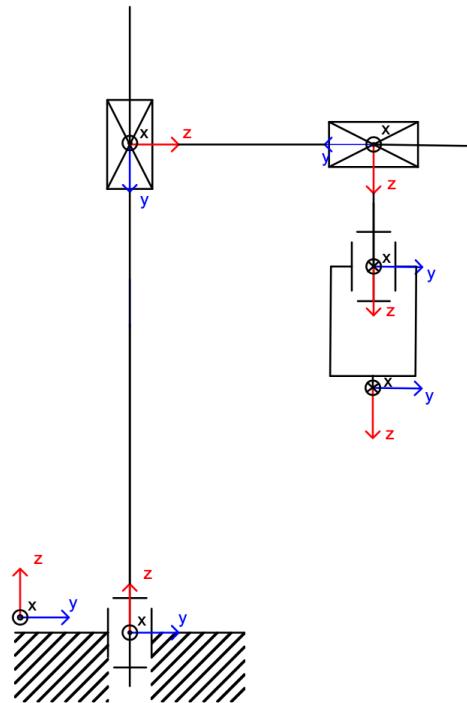
```

Korzystając z powyższego algorytmu przeprowadzam analizę poprawności wyników na poniższych przykładach.

Przykład 1:

$$\theta_1 = 0^\circ, \theta_4 = 180^\circ, d_1 = 2, d_2 = 1, d_3 = 0.5, d_4 = 0.5$$

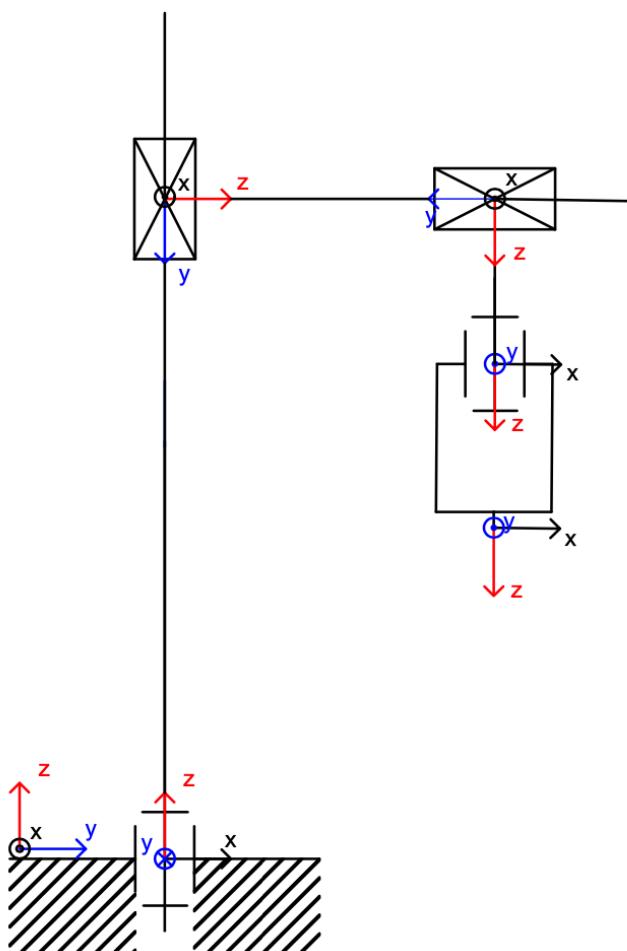
$$T_{4,0} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Przykład 2 :

$$\theta_1 = 90^\circ, \theta_4 = 90^\circ, d_1 = 2, d_2 = 1, d_3 = 0.5, d_4 = 0.5$$

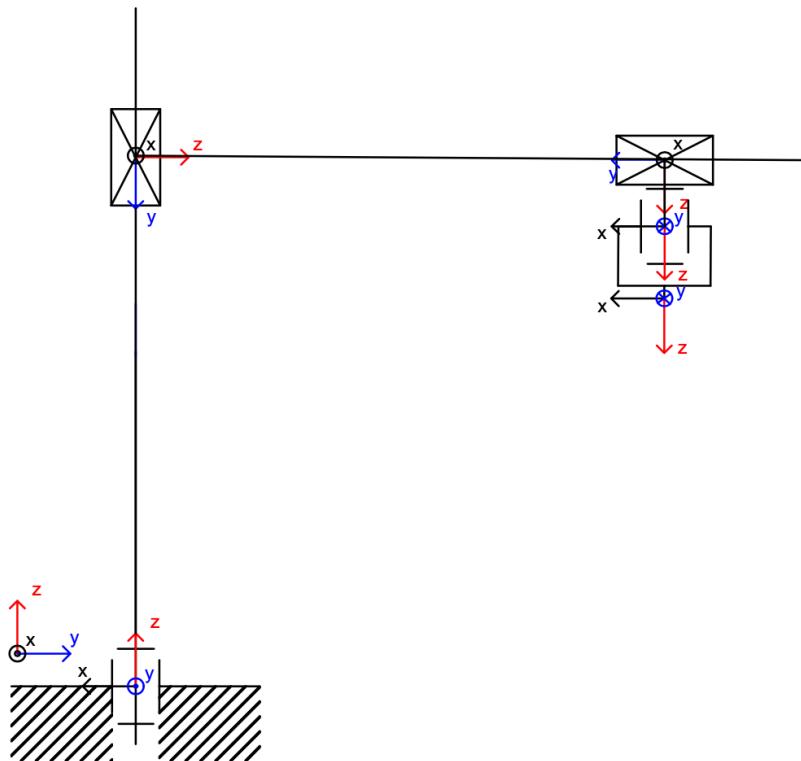
$$T_{4,0} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Przykład 3:

$$\theta_1 = -90^\circ, \theta_4 = 90^\circ, d_1 = 4, d_2 = 4, d_3 = 0.5, d_4 = 0.5$$

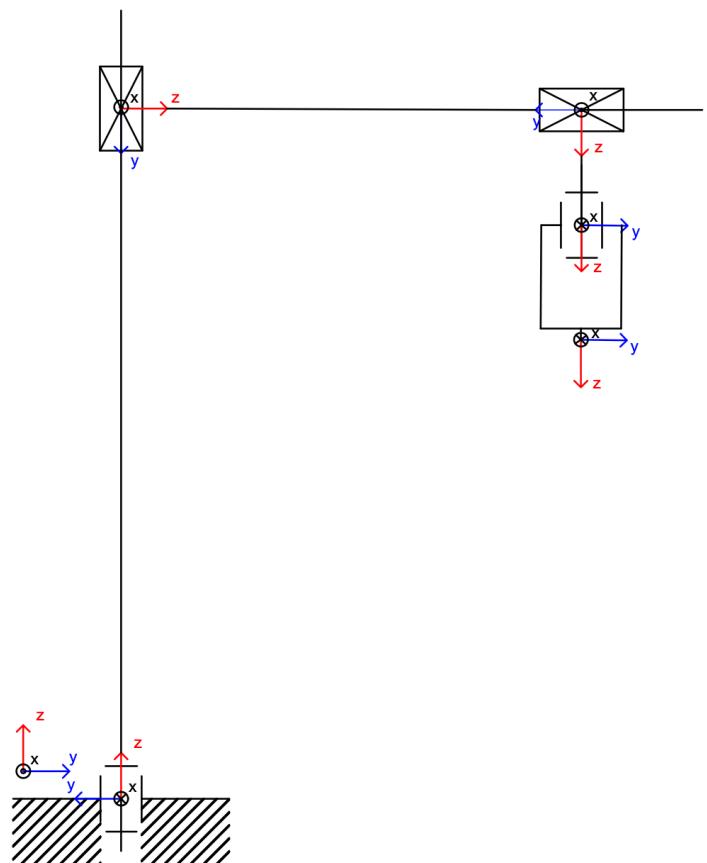
$$T_{4,0} = \begin{bmatrix} -1 & 0 & 0 & -4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Przykład 4:

$$\theta_1 = 180^\circ, \theta_4 = 180^\circ, d_1 = 3, d_2 = 2, d_3 = 0.5, d_4 = 0.5$$

$$T_{4,0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 2 \\ 0 & 0 & -1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Kinematyka prosta na realnych danych

W poprzednim punkcie korzystałem z prostych wartości kątów oraz uproszczonych długości boków w celu sprawdzenia poprawności obliczeń. Teraz zostaną wykonane kolejne dwa przykłady na realnych już danych oraz „trudniejszych” kątach.

Przykład 1:

$$\theta_1 = 13^\circ, \theta_4 = 10^\circ, d_1 = 400, d_2 = 300, d_3 = 105, d_4 = 90$$

$$T_{4,0} = \begin{bmatrix} 0.9986 & 0.052 & 0 & 67.4 \\ 0.052 & -1 & 0 & -292.3 \\ 0 & 0 & -1 & 205 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Przykład 2:

$$\theta_1 = 63^\circ, \theta_4 = 32^\circ, d_1 = 200, d_2 = 100, d_3 = 105, d_4 = 90$$

$$T_{4,0} = \begin{bmatrix} 0.8572 & 0.5150 & 0 & 89.1 \\ 0.5150 & -0.8572 & 0 & -34.399 \\ 0 & 0 & -1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Kinematyka odwrotna

Do wszystkich obliczeń zostanie użyty następujący skrypt napisany w środowisku Matlab:

```
%% Obliczenia
eqn1= T40(1,4)==x;
eqn2= T40(2,4)==y;
eqn3= T40(3,4)==z;
eqn4= T40(1,1)==a;
eqn5= T40(2,2)==b;
eqns = [eqn1,eqn2,eqn3];

wynik = solve(eqns,[d1 d2 teta1,]);
eqna = teta1 ==wynik.teta1;
eqns2 = [eqna,eqn5];
wynikteta = solve(eqns2, [teta1, teta4])
```

W zależności od przykładu zmieniane będą wartości parametrów x, y, z.

Rozwiążane zostaną w ten sposób wszystkie przykłady umieszczone w poprzednim rozdziale – pozwoli to na szybką weryfikację poprawności uzyskanych wyników.

Przykład 1

```
%% Wartości z macierzy T40
x=0;
y=-1;
z=1;
a=-1;
b=1;
```

ResultD1 =

2

ResultD2 =

1

ResultTeta1 =

0

ResultTeta4 =

pi

Zrzut ekranu uzyskanego wyniku dla przykładu 1.

Wynik uzyskany w tym przykładzie jest poprawny.

Przykład 2

```
%% Wartości z macierzy T40
x=1;
y=0;
z=1;
a=1;
b=-1;
```

ResultD1 =

2

ResultD2 =

1

ResultTeta1 =

$\pi/2$

ResultTeta4 =

$\pi/2$

Zrzut ekranu uzyskanego wyniku dla przykładu 2.

Wynik uzyskany w tym przykładzie jest poprawny.

Przykład 3

```
%% Wartości z macierzy T40
x=-4;
y=0;
z=3;
a=-1;
b=1;
```

ResultD1 =

4

ResultD2 =

4

ResultTeta1 =

$-\pi/2$

ResultTeta4 =

$\pi/2$

Zrzut ekranu uzyskanego wyniku dla przykładu 3.

Wynik uzyskany w tym przykładzie jest poprawny.

Przykład 4

```
%% Wartości z macierzy T40
x=0;
y=2;
z=2;
a=1;
b=-1;
```

ResultD1 =

3

ResultD2 =

2

ResultTeta1 =

pi

ResultTeta4 =

pi

Zrzut ekranu uzyskanego wyniku dla przykładu 4.

Wynik uzyskany w tym przykładzie jest poprawny.

Kinematyka odwrotna na realnych danych

Podobnie jak w przypadku kinematyki prostej – teraz przejdę do policzenia analogicznych przykładów oraz sprawdzenia wyników dla realnych parametrów uzyskiwanych w ruchu robota.

Przykład 1:

%% Wartości z macierzy T40

```
x=67.4;
y=-292.4;
z=205;
b=-1;
```

```
ResultD1 =
400
400
400
400

ResultD2 =
-2251013^(1/2)/5
-2251013^(1/2)/5
2251013^(1/2)/5
2251013^(1/2)/5

ResultTeta1 =
atan(337/1462) - pi
atan(337/1462) - pi
atan(337/1462)
atan(337/1462)

ResultTeta4 =
-log(- 157460610363^(1/2)*(337/11255065000 - 731i/5627532500) - 2251013^(1/2)*(3649883/56275:
    -log(157460610363^(1/2)*(337/11255065000 - 731i/5627532500) - 2251013^(1/2)*(3649883/56275:
    -log(- 157460610363^(1/2)*(337/11255065000 - 731i/5627532500) + 2251013^(1/2)*(3649883/56275:
    -log(157460610363^(1/2)*(337/11255065000 - 731i/5627532500) + 2251013^(1/2)*(3649883/56275:
```

Matlab wyświetla wyniki w dziwnej postaci, ale po przeliczeniu ich do wartości liczbowych i wybraniu odpowiednich rozwiązań otrzymujemy:

$$D1 = 400\text{mm}$$

$$D2 = 300.06\text{mm}$$

$$Teta1 = 9.95^\circ$$

$$Teta4 = 12.98^\circ$$

Wyniki zgadzają się z założeniami kinematyki prostej.

Przykład 2:

```

%% Wartości z macierzy T40
x=89.1;
y=-34.399;
z=5;
a=0.8572;
b=-0.8572;

ResultD1 =
200
200
200
200

ResultD2 =
9122101201^(1/2)/1000
9122101201^(1/2)/1000
-9122101201^(1/2)/1000
-9122101201^(1/2)/1000

ResultTeta1 =
pi/2 - atan(34399/89100)
pi/2 - atan(34399/89100)
-2*atan(9122101201^(1/2)/89100 + 34399/89100)
-2*atan(9122101201^(1/2)/89100 + 34399/89100)

ResultTeta4 =
-log(9122101201^(1/2)*(73717057/22805253002500 + 1909413i/228052530025) - 1512034796781875)
-log(9122101201^(1/2)*(73717057/22805253002500 + 1909413i/228052530025) + 1512034796781875)
-log(- 9122101201^(1/2)*(73717057/22805253002500 + 1909413i/228052530025) - 1512034796781875)
-log(- 9122101201^(1/2)*(73717057/22805253002500 + 1909413i/228052530025) + 1512034796781875)

```

Ponownie Matlab wyświetla wyniki w dziwnej postaci, ale po przeliczeniu ich do wartości liczbowych i wybraniu odpowiednich rozwiązań otrzymujemy:

$$D_1 = 200\text{mm}$$

$$D_2 = 96.5\text{mm}$$

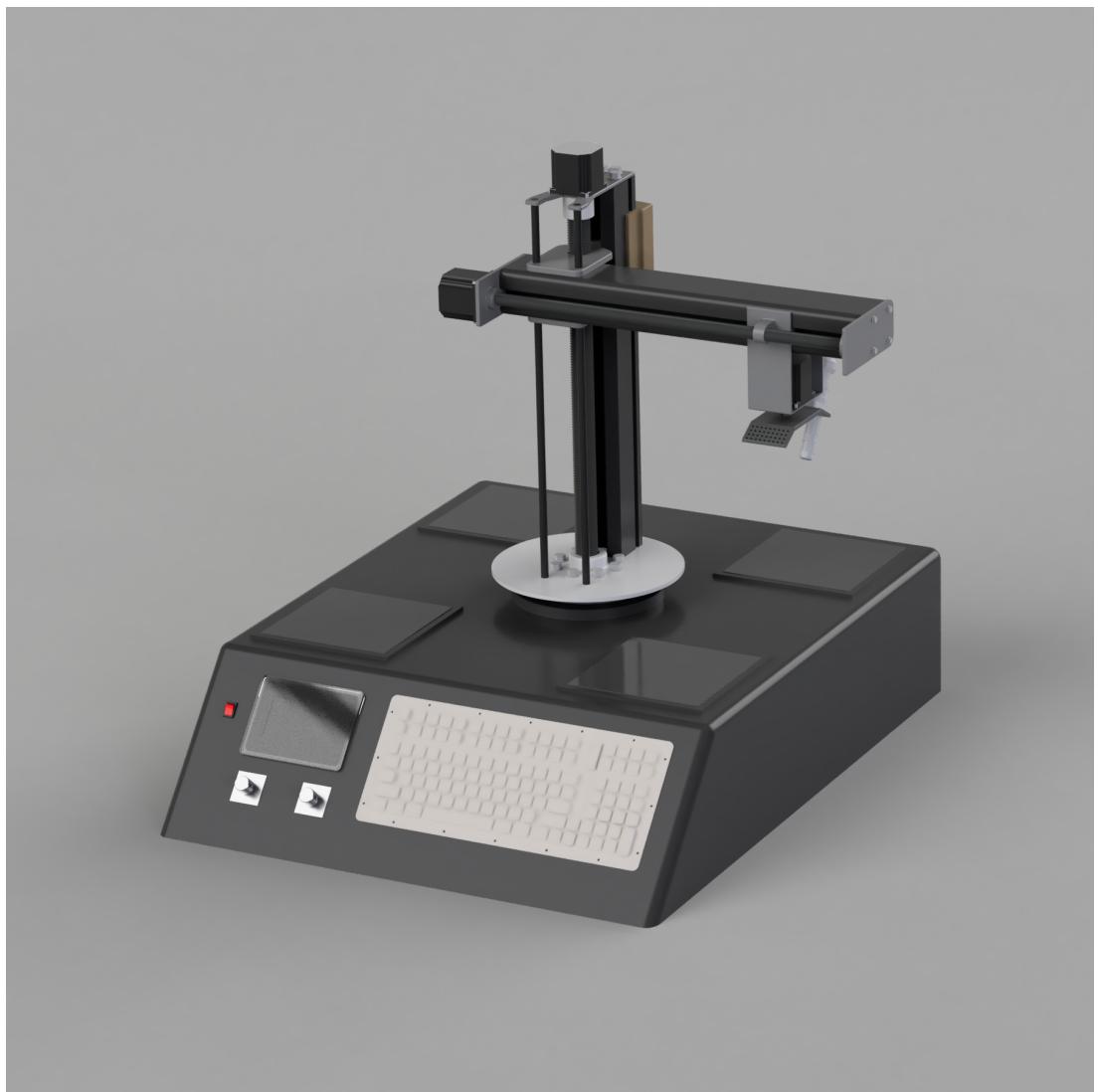
$$\Theta_1 = 66.3^\circ$$

$$\Theta_4 = 34.79^\circ$$

Wyniki zgadzają się z założeniami kinematyki prostej, uznajemy błędy za wynikające z przybliżeń liczb.

Powyższe przykłady potwierdzają poprawność wykonania modelu.

Model CAD



Widok izometryczny

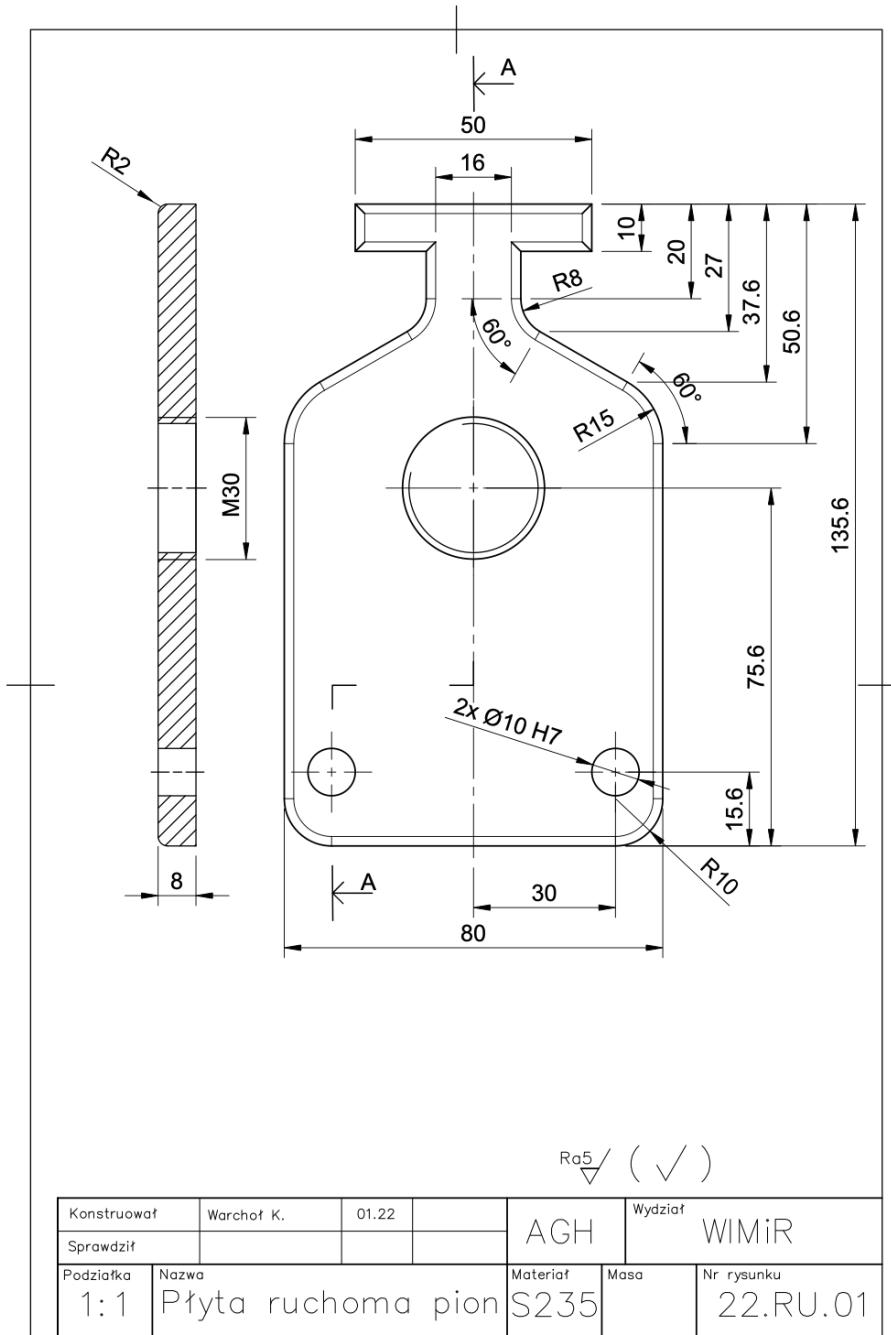


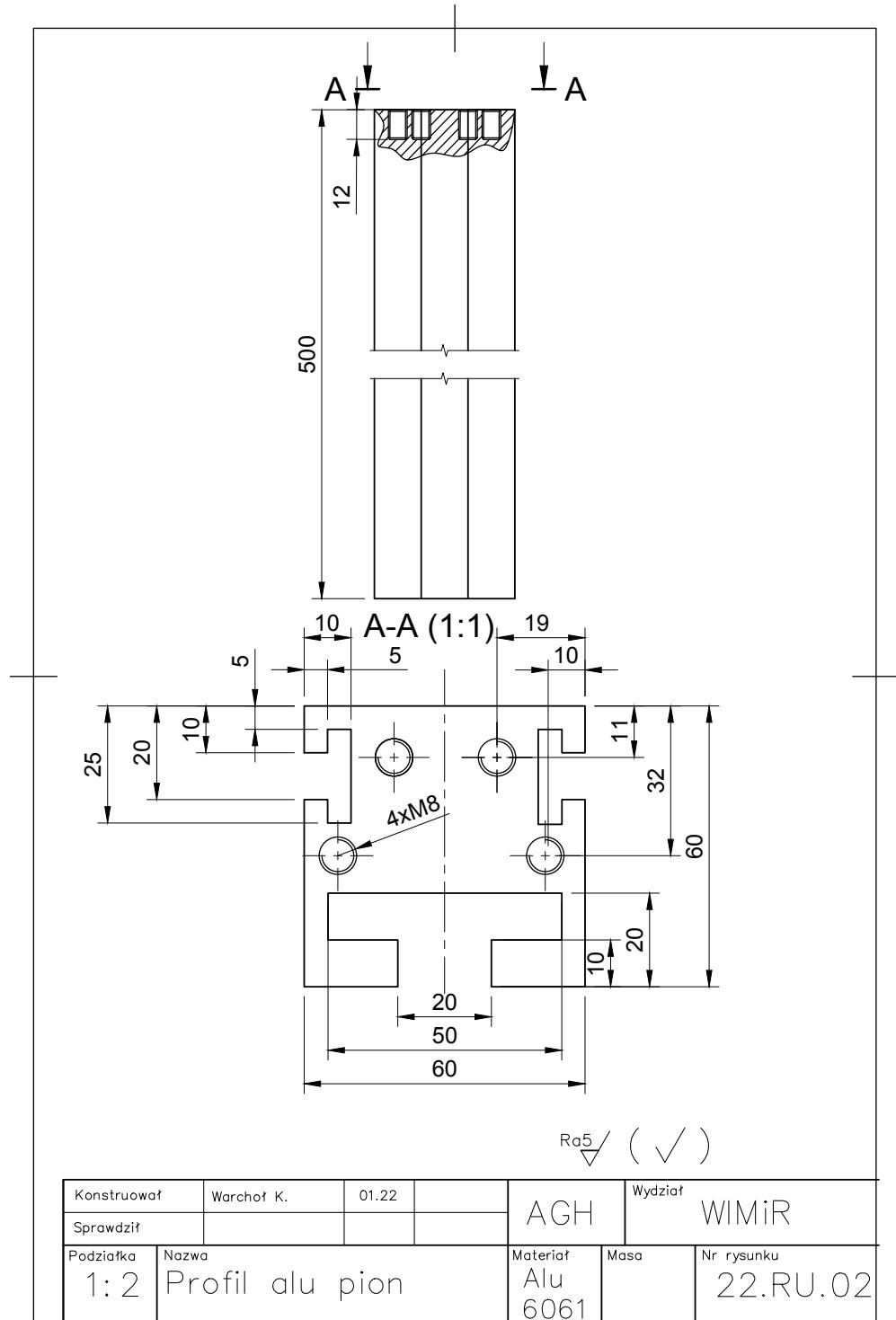
Widok z przodu

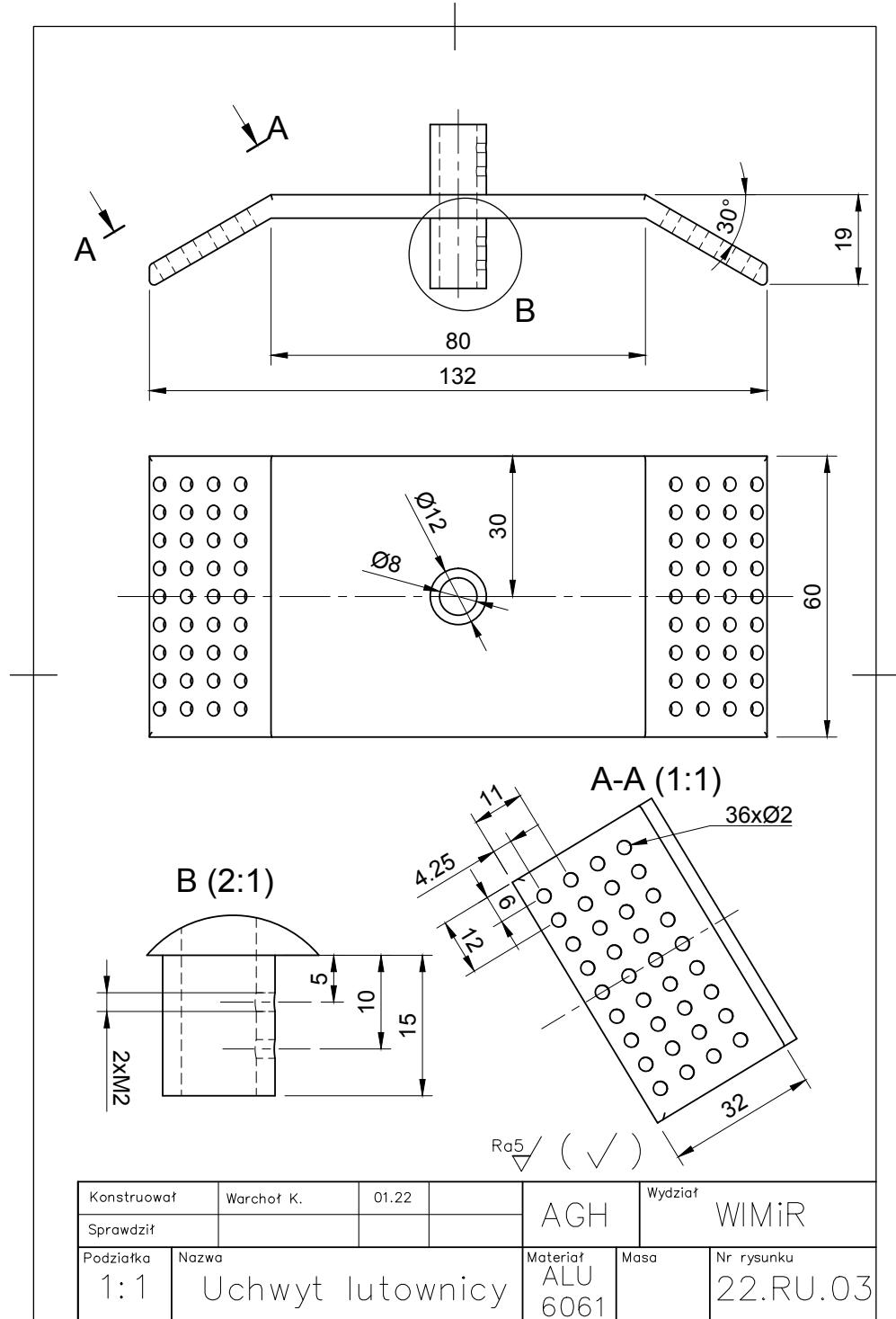


Widok z prawej

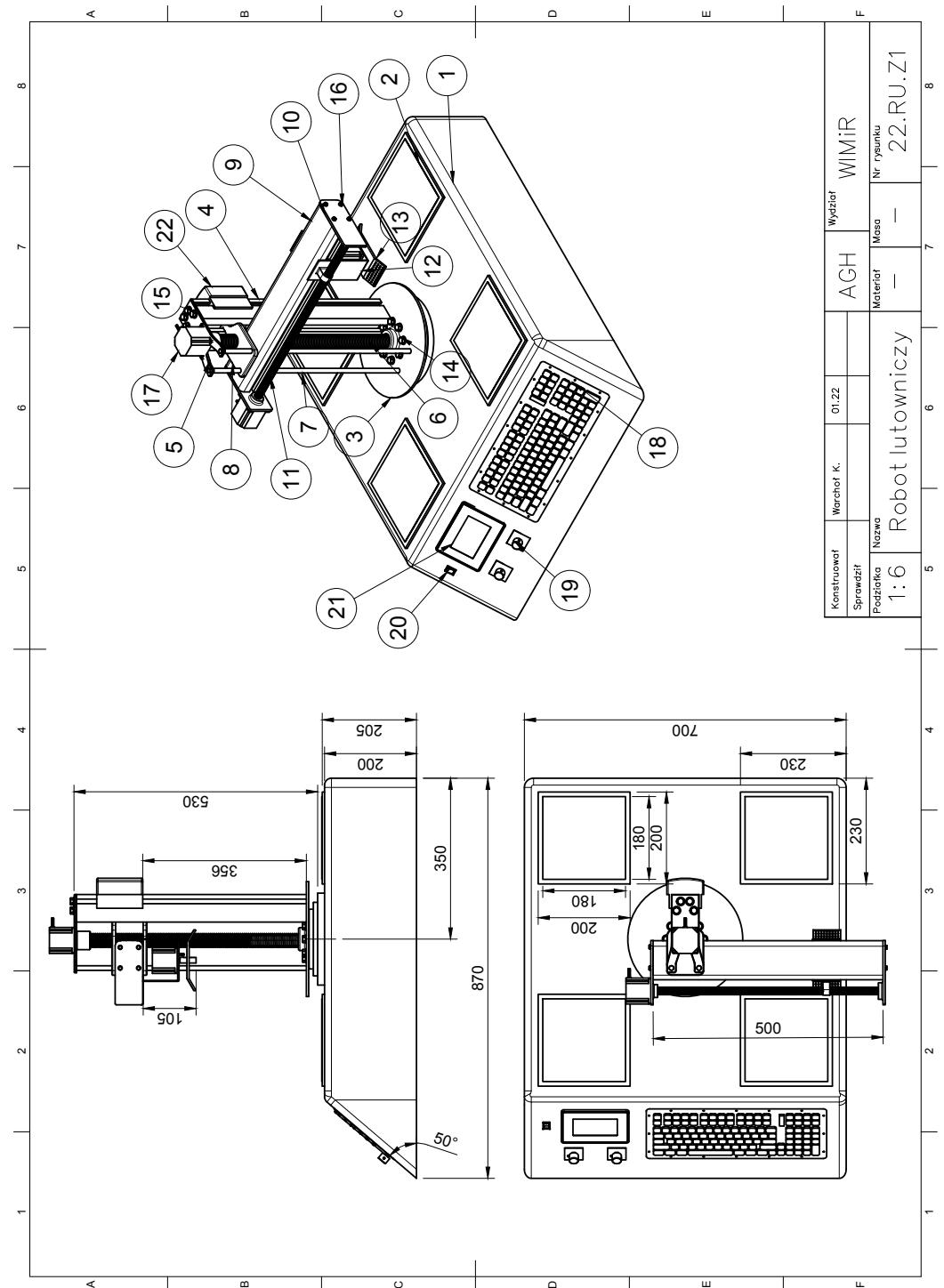
Rysunki wykonawcze







Rysunek złożeniowy



22	Sterownik lutowniczy		1	–	–	–	–
21	Wyświetlacz		1	–	–	–	[9]
20	Włącznik		1	–	–	–	[10]
19	Enkoder		2	–	–	–	[11]
18	Klawiatura		1	–	–	–	[8]
17	Silnik NEMA23		4	–	–	–	[7]
16	Śruba M4 x 8		8	DIN 7500	C35	–	[12]
15	Śruba M8 x 12		4	DIN 6921	C35	–	[12]
14	Śruba M8 x 20		8	DIN 6921	C35	–	[12]
13	Uchwyt lutowniczy		1	22.RU.03	Alu 6061	–	–
12	Przesuwna platforma		1	–	Alu 6061	–	–
11	Pręt gwintowany M16		1	DIN 976	Stal A2	–	–
10	Nakładka boczna		2	–	Alu 6061	–	–
9	Profil poziomy		1	–	Alu 6061	–	–
8	Płyta ruchoma pion gwintowana M30		2	22.RU.01	Alu 6061	–	–
7	Prowadnica Ø8		2	–	S235JR	–	–
6	Pręt gwintowany M30		1	DIN 976	Stal A2	–	–
5	Nakładka górnna		1	–	Alu 6061	–	–
4	Profil pionowy		1	22.RU.02	Alu 6061	–	–
3	Stół obrotowy		1	–	Alu 6061	–	–
2	Podkładka lutownicza		4	–	Alu 6061	–	–
1	Podstawa		1	–	Alu 6061	–	–
Nr poz	Konstruował			Liczba sztuk	Nr rysunku/normy	Materiał	Masa
	Konstruował	Warchał K.	01.22		AGH	Wydział WIMiR	
	Sprawdził						
Podziałka	Nazwa			Materiał	Masa	Nr rysunku	
1:6	Robot lutowniczy			–	–	22.RU.Z1	

Źródła

- [1] <https://wobit.com.pl/produkt/10753/roboty-przemyslowe-tower/robot-tower-tr300/>
- [2] <https://kaisertech.co.uk/apollo-j-cat-stellar-soldering-robot>
- [3] <https://www.quiptech.com/europe/products-europe/capital-equipment/japan-unix-df-series-benchtop-automanted-robot-soldering/>
- [4] https://smtnet.com/company/index.cfm?fuseaction=view_company&company_id=54658&component=catalog&catalog_id=22551
- [5] <https://m.made-in-china.com/product/5-Axis-Soldering-Robot-943522915.html>
- [6] <https://www.jbctools.com/tr-soldering-iron-for-robot-product-1432.html>
- [7] <https://www.wmberg.com/products/more/electro-mechanical/1/sm23-401?pcSession=OK>
- [8] <https://b2b.partcommunity.com/3d-cad-models/sso/hzs-732-control-panel-7-sigmatek?info=sigmatek%2Fspecial20applications%2Fenergy20automation%2Fhzs2073220-20control20panel207.prj>
- [9] <https://b2b.partcommunity.com/3d-cad-models/sso/100002311-tx100-hmi-series-4-display-with-tx-visupro-runtime-high-quality-plastic-housi-turck?info=turck%2Fsteuerung20und20hmi%2Fhmi-bediengeraet%2F100002311.prj&cwid=5835>
- [10] https://b2b.partcommunity.com/3d-cad-models/sso/cwsb-neon-miniature-power-rockers-nkk-switches?info=nkkswitches%2Frockers%2Fcw Illumin%2Fcwsb Illumin_asmtab.prj&cwid=7735
- [11] https://b2b.partcommunity.com/3d-cad-models/sso/external-speed-potentiometer-orientalmotor-co-ltd?info=oriental%2Fselection_from_product_tree%2Fbrushless_dc_motors%2Foption%2Fset_machine.prj&cwid=8267
- [12] https://b2b.partcommunity.com/3d-cad-models/sso/din-6921-hexagon-bolts-with-flange-factory?info=factory%2F1_bolts_and_hex_cap_screws%2Fdin_6921_v2.prj&cwid=6839
- [13] <https://www.ebmia.pl/lozyska-z-aluminiowym-pierscieniem-do-stolow-obrotowych/201123-lozysko-talerzowe-prt-02-60-al.htm>