

# Projekt ćwiczeniowy 1

Do przedmiotu

## Zasady Budowy Robotów

Instrukcja użytkownika

# 1. Cel

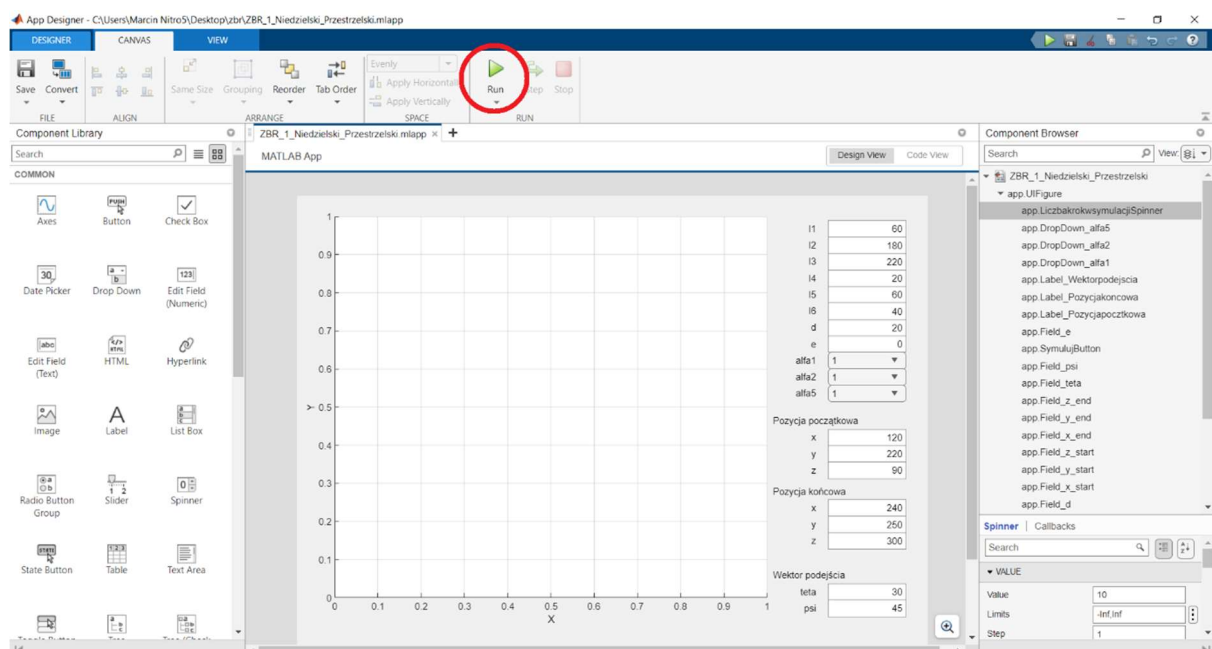
Celem projektu było stworzenie aplikacji, której zadaniem jest symulacja ruchu robota sześciokościowego o strukturze kinematycznej  $\{C_R, B_{R1}, B_{R2}, B_L, C_L, A_L\}$  i określonej budowie oraz wizualizacja ruchu punktu TCP. Użytkownik ma możliwość zadania wartości parametrów geometrycznych części regionalnej i lokalnej robota. Może on także wybrać wektor podejścia członów części lokalnej robota oraz wyboru punktu startowego i końcowego trajektorii ruchu zadaną liczbą kroków przejścia.

# 2. Wymagania

Poprawne działanie programu wymaga posiadania programu Matlab R2023a i bibliotekę AppDesigner.

# 3. Obsługa aplikacji

Aby uruchomić aplikację należy kliknąć dwukrotnie na plik „ZBR\_1\_Niedzielski\_Przestrzelski” lub wgrać do Matlab-a. Następnie uruchomić program prze wciśnięcie zielonej strzałki.



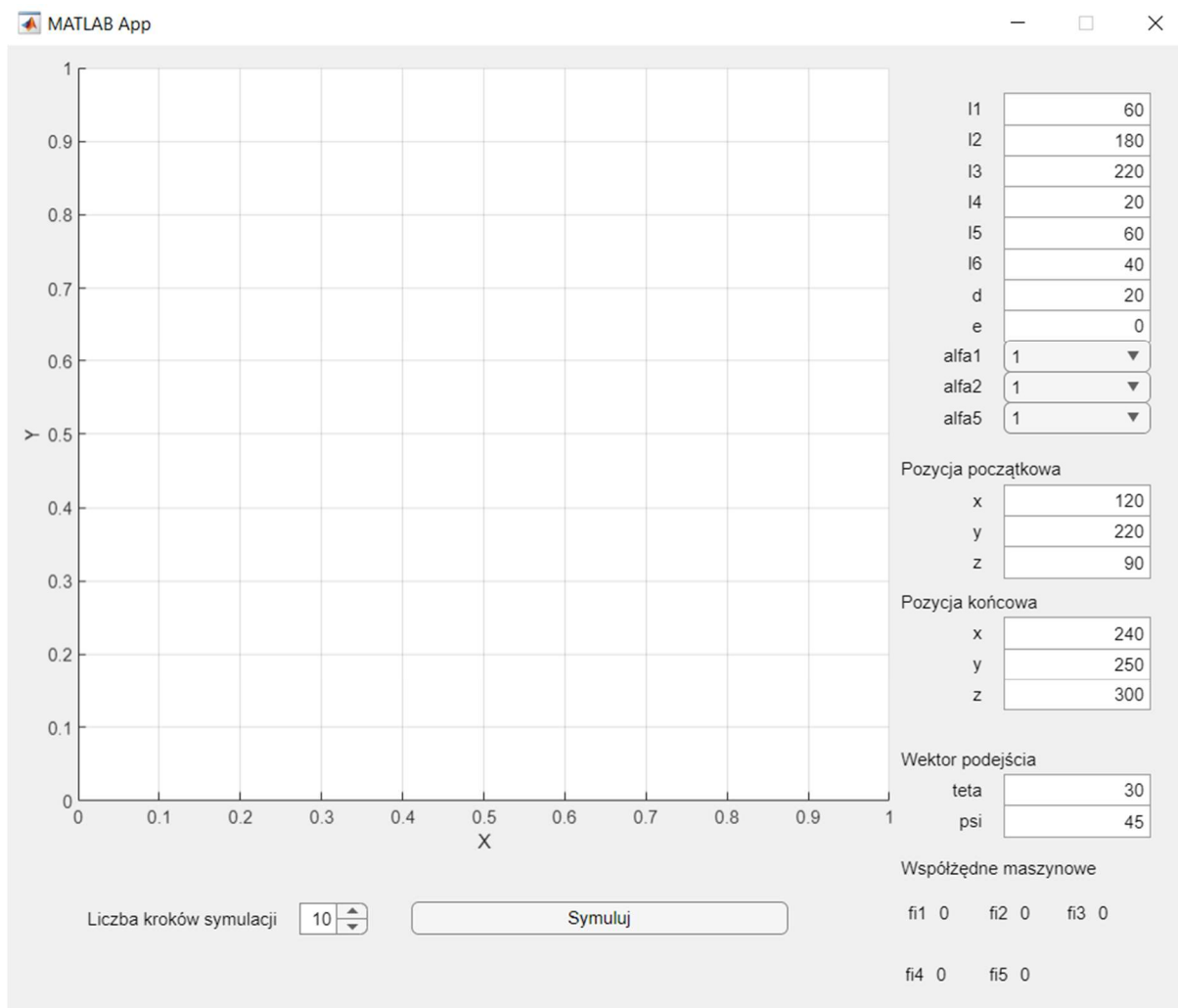
Rys.1 Uruchomienie aplikacji

Użytkownik może zmieniać długość poszczególnych członów robota. Może on także wybrać wektor podejścia członów części lokalnej robota oraz wyboru punktu startowego i końcowego trajektorii ruchu zadaną liczbą kroków przejścia.

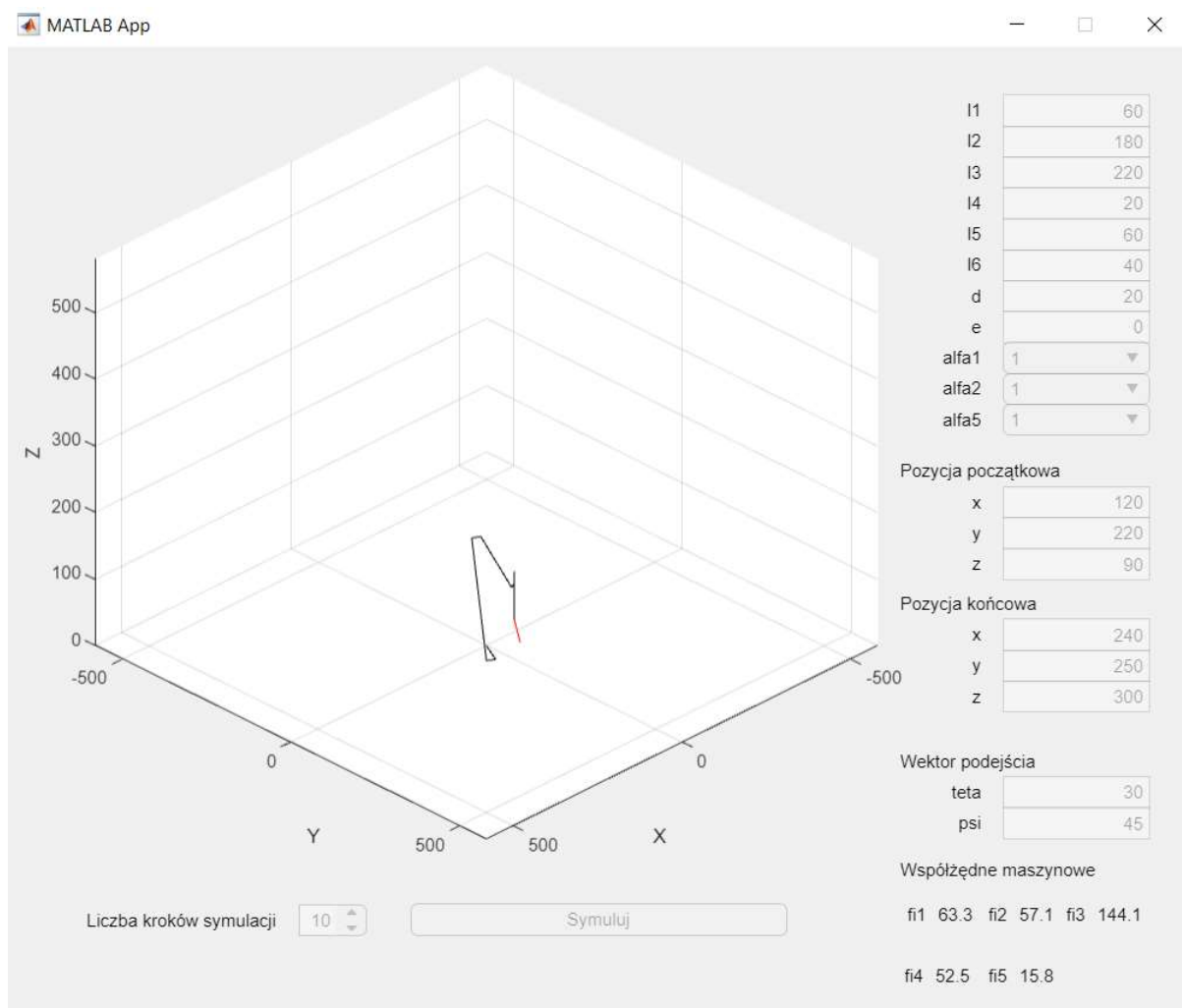
Należy wybrać liczbę kroków symulacji. Każdy krok symulacji wykonuje się 0,2 sekundy.

Wciśnięcie przycisku symuluj zainicjuje działanie programu i pokaże się obraz robota w 3D, poruszający się po trasie ruchu TCP, przy zastosowaniu interpolacji liniowej.

Po najechaniu na gotowy wykres można za pomocą myszki manipulować kątem patrzenia na wykres, a także przybliżyć lub oddalić go.

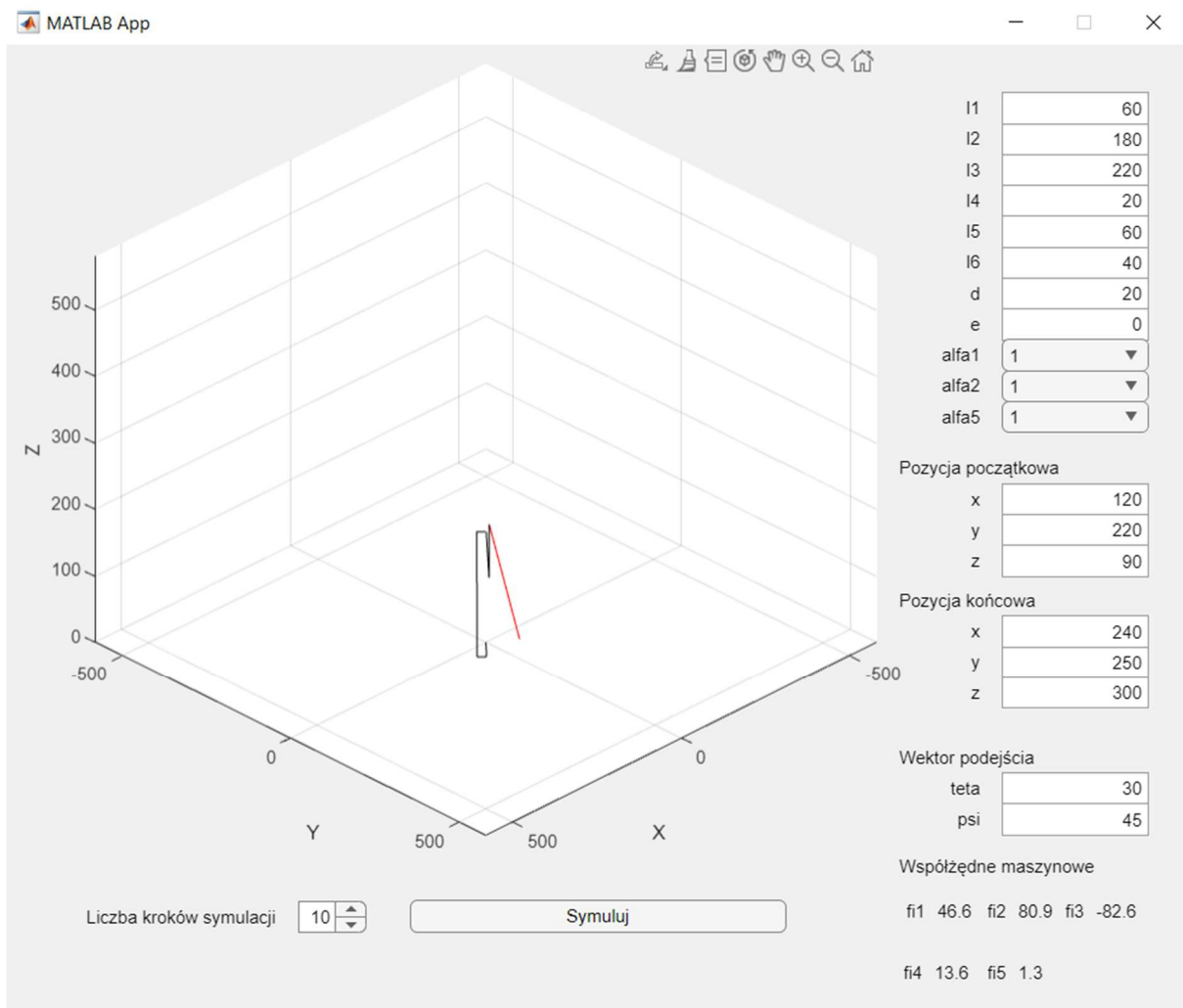


Rys.2 Okno symulacji po włączeniu programu

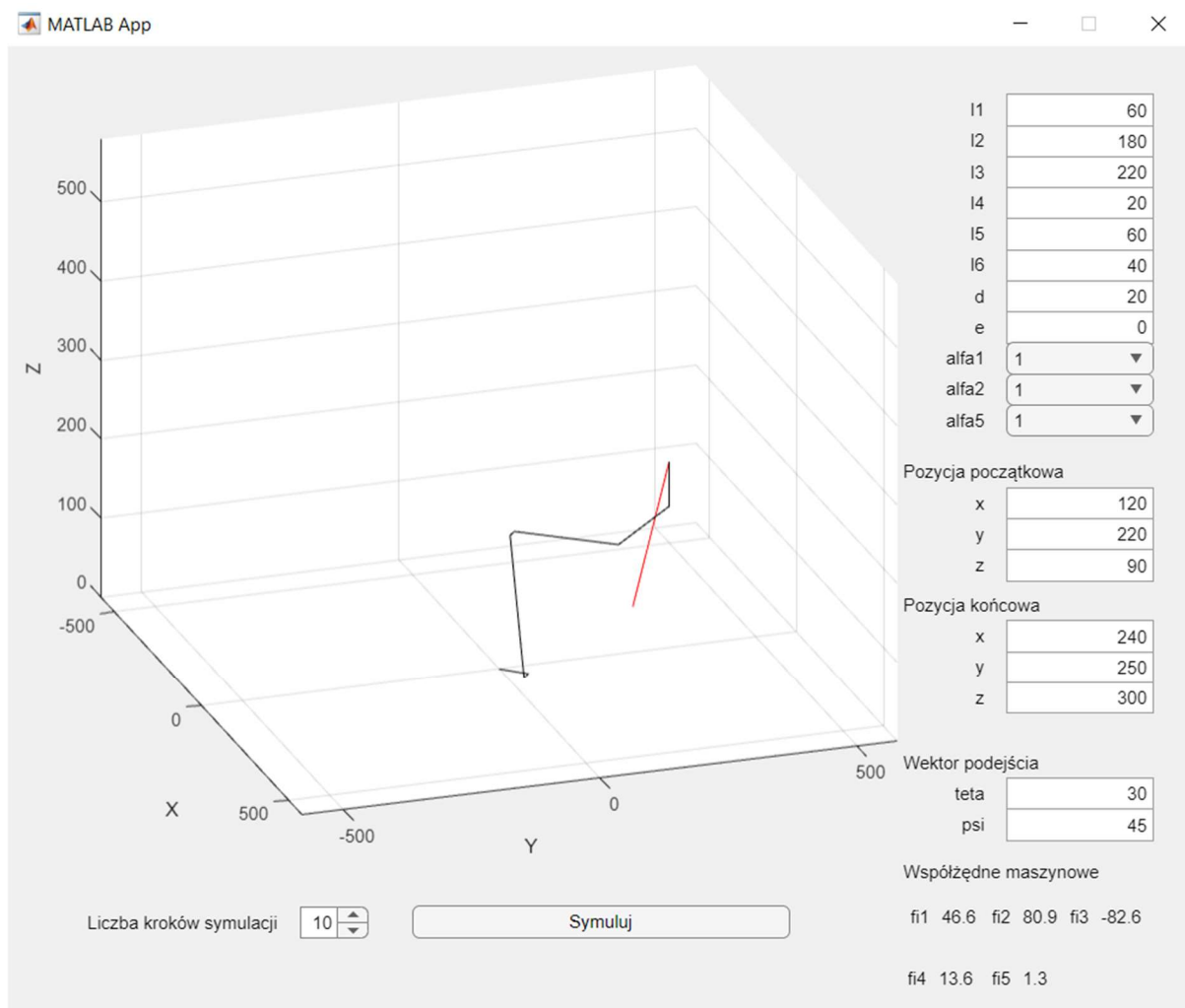


Rys.3 Początek ruchu robota

Ostatnia prosta(czerwona) obrazuje trajektorię ruchu TCP.



Rys.4 Pozycja końcowa robota



Rys.5 Pozycja końcowa ukazana z innego kąta

Listing kodu:

```
properties (Access = public)
    l1;
    l2;
    l3;
    l4;
    l5;
    l6;
    e;
    d;

    l; %l=l5+l6

    alfa1;
    alfa2;
    alfa3;
    alfa5;

    %wartosci wpisywane przez uzytkownika
    %poczatkowa pozycja tcp
    X_p;
    Y_p;
    Z_p;

    %na potrzeby programu
    xp;
    yp;
    zp;
    %wartosci wpisywane przez uzytkownika
    %koncowa pozycja tcp
    xk;
    yk;
    zk;

    xr;
    yr;
    zr;

    psi;
    teta;

    point1=[0,0,0];
    point1prim=[0,0,0];
    point2prim=[0,0,0];
    point2=[0,0,0];
    point3=[0,0,0];
    point4=[0,0,0];
    point5=[0,0,0];
    point6=[0,0,0];

    line1;
    line2;
    line3;
    line4;
    line5;
    line6;
    line7;
    line8;
```

```

%linia pokazująca ruch ostatniego punktu
line9;

S1=0;
C1=0;
a=0;
b=0;
S2=0;
C2=0;
S3=0;
C3=0;
S23=0;
C23=0;
S5=0;
C5=0;
S234=0;
C234=0;
S4=0;
C4=0;
S6=0;
C6=0;

%współrzędne maszynowe
fi1;
fi2;
fi3;
fi4;
fi5;
fi23;
fi234;
fi6;

kroki=0;
czas=0;

sprawdz=0;
wieksze_od_zera;
end

methods (Access = public)
function oblicz(app)
    step = app.kroki;

    X_k = app.xk;
    Y_k = app.yk;
    Z_k = app.zk;

    app.point6=[app.xk,app.yk,app.zk];
    app.l = app.l5 + app.l6;

    %zmiana z char na double
    app.alfa1 = str2double(app.alfa1);
    app.alfa2 =str2double(app.alfa2);
    app.alfa5 =str2double(app.alfa5);

    for i = 0:1:step

```



```

xt=((X_k-app.X_p)/step)*i + app.X_p;
yt=((Y_k-app.Y_p)/step)*i + app.Y_p;
zt=((Z_k-app.Z_p)/step)*i + app.Z_p;
app.point6=[xt,yt,zt];
app.xp = xt - app.l * cosd(app.teta) * sind(app.psi);
app.yp = yt- app.l * cosd(app.teta) * sind(app.psi);
app.zp = zt - app.l * sind(app.teta);

app.wieksze_od_zera = app.xp^2+app.yp^2-app.e^2; %ta wartosc musi
byc wieksza od 0

app.S1=1/(app.xp^2+app.yp^2)*(app.e*app.xp+
app.alfa1*app.yp*sqrt(app.xp^2+app.yp^2-app.e^2));
app.C1=1/( app.xp^2+ app.yp^2)*(-1* app.e* app.yp+
app.alfa1* app.xp*sqrt( app.xp^2+ app.yp^2- app.e^2));
app.sprawdz = app.S1^2+app.C1^2; %Jedynka Trygonometryczna
if abs(app.S1)>=abs(app.C1)
    app.fi1=asin(app.S1);
else
    app.fi1=acos(app.C1);
end

app.S5=cosd( app.teta)*(sind( app.psi)* app.C1-cosd( app.psi)*
app.S1);
app.C5= app.alfa5*sqrt(1- app.S5.^2);
app.sprawdz = app.S5^2+app.C5^2; %Jedynka Trygonometryczna
if abs(app.S5)>=abs(app.C5)
    app.fi5=asin(app.S5);
else
    app.fi5=acos(app.C5);
end

app.S234=sind( app.teta)/ app.C5;
app.C234=(cosd( app.teta)/ app.C5)*(cosd( app.psi)* app.C1+sind(
app.psi)* app.S1);
app.sprawdz = app.S234^2+app.C234^2;%Jedynka Trygonometryczna
if abs(app.S234)>=abs(app.C234)
    app.fi234=asin(app.S234);
else
    app.fi234=acos(app.C234);
end

%Wyznaczenie końca części regionalnej R
app.xr= app.xp - app.l4* app.C1* app.C234;
app.yr= app.yp - app.l4* app.S1* app.C234;
app.zr= app.zp - app.l4* app.S234;
app.wieksze_od_zera = app.xr^2+app.yr^2-app.e^2; %ta wartosc musi
byc wieksza od 0

app.a=-app.l1+( app.alfa1*sqrt(app.xr^2+ app.yr^2- app.e^2));
app.b=1/(2* app.l2)*(app.a^2+ app.zr^2+app.l2^2-app.l3^2);
app.wieksze_od_zera = app.a^2+app.zr^2-app.b^2; %ta wartosc musi
byc wieksza od 0

app.S2=(1/(app.a^2+app.zr^2))*(app.zr*app.b+app.alfa2*app.a*(app.a^2+app.zr^2-
app.b^2)^(0.5));

```

```

app.C2=(1/(app.a^2+app.zr^2))*(app.a*app.b-
app.alfa2*app.zr*(app.a^2+app.zr^2-app.b^2)^(0.5));
app.sprawdz = app.S2^2+app.C2^2 ;%Jedynka Trygonometryczna
if abs(app.S2)>=abs(app.C2)
    app.fi2=asin(app.S2);
else
    app.fi2=acos(app.C2);
end

app.S3=-1*( app.alfa2/ app.l3)*sqrt( app.a^2+ app.zr^2-
app.b^2);
app.C3=( app.b- app.l2)/ app.l3;
app.sprawdz = app.S3^2 + app.C3^2;%Jedynka Trygonometryczna
if abs(app.S3)>=abs(app.C3)
    app.fi3=asin(app.S3);
else
    app.fi3=acos(app.C3);
end

app.S23 = 1/ app.l3 * (app.zr - app.l2 * app.S2);
app.C23 = 1/ app.l3 * (app.a- app.l2 * app.C2);
app.sprawdz = app.S23^2 + app.C23^2;%Jedynka Trygonometryczna
if abs(app.S23)>=abs(app.C23)
    app.fi23=asin(app.S23);
else
    app.fi23=acos(app.C23);
end

app.S4= app.S234* app.C23- app.C234* app.S23;
app.C4= app.C234* app.C23+ app.S234* app.S23;
app.sprawdz = app.C4^2 + app.S4^2;%Jedynka Trygonometryczna
if abs(app.S4)>=abs(app.C4)
    app.fi4=asin(app.S4);
else
    app.fi4=acos(app.C4);
end

%Wszystkie jedynki trygonometryczne są poprawne więc wzory
%wyżej także są poprawne

% W matlabie najmniejszy mozliwy indeks to 1
app.point1(1)=app.l1*app.C1;
app.point1(2)=app.l1*app.S1;
app.point1(3)=0;

app.point1prim(1)=app.point1(1)+app.d*app.S1;
app.point1prim(2)=app.point1(2)-app.d*app.C1;
app.point1prim(3)=0;

app.point2prim(1)=app.point1prim(1)+app.l2*app.C2*app.C1;
app.point2prim(2)=app.point1prim(2)+app.l2*app.C2*app.S1;
app.point2prim(3)=app.l2*app.S2;

app.point2(1)=app.point2prim(1)-(app.d-app.e)*app.S1;
app.point2(2)=app.point2prim(2)+(app.d-app.e)*app.C1;
app.point2(3)=app.point2prim(3);

%inaczej Xr

```

```

app.point3(1)=app.point2(1)+app.l3*app.C1*app.C23;
app.point3(2)=app.point2(2)+app.l3*app.S1*app.C23;
app.point3(3)=app.point2(3)+app.l4*app.S23;

%inaczej Xp
app.point4(1)=app.point3(1)+app.l4*app.C1*app.C234;
app.point4(2)=app.point3(2)+app.l4*app.S1*app.C234;
app.point4(3)=app.point3(3)+app.l4*app.S234;

%inaczej Xt
%app.l = app.l5 + app.l6;
app.point5(1)=app.point4(1)+app.l*cosd(app.teta)*cosd(app.psi);
app.point5(2)=app.point4(2)+app.l*cosd(app.teta)*sind(app.psi);
app.point5(3)=app.point4(3)+app.l*sind(app.teta);

app.Axes.XLim=[-1*(app.l1+app.l2+app.l3+app.l4+app.l5+app.l6)
(app.l1+app.l2+app.l3+app.l4+app.l5+app.l6)];
app.Axes.YLim=[-1*(app.l1+app.l2+app.l3+app.l4+app.l5+app.l6)
(app.l1+app.l2+app.l3+app.l4+app.l5+app.l6)];
app.Axes.ZLim=[0 (app.l1+app.l2+app.l3+app.l4+app.l5+app.l6)];

view(app.Axes,[(app.l1+app.l2+app.l3+app.l4+app.l5+app.l6)/2
(app.l1+app.l2+app.l3+app.l4+app.l5+app.l6)/2
(app.l1+app.l2+app.l3+app.l4+app.l5+app.l6)/2]);

app.line1=[0,0,0];app.point1];
app.line2=[app.point1;app.point1prim];
app.line3=[app.point1prim;app.point2prim];
app.line4=[app.point2prim;app.point2];
app.line5=[app.point2;app.point3];
app.line6=[app.point3;app.point4];
app.line7=[app.point4;app.point5];
app.line8=[app.point5;app.point6];
app.line9=[app.X_p,app.Y_p,app.Z_p];app.point6];
cla(app.Axes);

plot3(app.Axes,app.line1(:,1),app.line1(:,2),app.line1(:,3),('black'));
plot3(app.Axes,app.line2(:,1),app.line2(:,2),app.line2(:,3),('black'));
plot3(app.Axes,app.line3(:,1),app.line3(:,2),app.line3(:,3),('black'));
plot3(app.Axes,app.line4(:,1),app.line4(:,2),app.line4(:,3),('black'));
plot3(app.Axes,app.line5(:,1),app.line5(:,2),app.line5(:,3),('black'));
plot3(app.Axes,app.line6(:,1),app.line6(:,2),app.line6(:,3),('black'));
plot3(app.Axes,app.line7(:,1),app.line7(:,2),app.line7(:,3),('black'));
plot3(app.Axes,app.line8(:,1),app.line8(:,2),app.line8(:,3),('black'));
plot3(app.Axes,app.line9(:,1),app.line9(:,2),app.line9(:,3),('red'));

app.fi1st.Text=string(round(app.fi1*180/pi(),1));
app.fi2st.Text=string(round(app.fi2*180/pi(),1));

```

```

        app.fi3st.Text=string(round(app.fi3*180/pi(),1));
        app.fi4st.Text=string(round(app.fi4*180/pi(),1));
        app.fi5st.Text=string(round(app.fi5*180/pi(),1));
        pause(1/5);

    end

end

end

% Callbacks that handle component events
methods (Access = private)

    % Button pushed function: SymulujButton
    function SymulujButtonPushed(app, event)
        app.l1=app.Field_l1.Value;
        app.l2=app.Field_l2.Value;
        app.l3=app.Field_l3.Value;
        app.l4=app.Field_l4.Value;
        app.l5=app.Field_l5.Value;
        app.l6=app.Field_l6.Value;
        app.e=app.Field_e.Value;
        app.d=app.Field_d.Value;

        app.alfa1=app.DropDown_alfa1.Value;
        app.alfa2=app.DropDown_alfa2.Value;
        app.alfa5=app.DropDown_alfa5.Value;

        app.X_p=app.Field_x_start.Value;
        app.Y_p=app.Field_y_start.Value;
        app.Z_p=app.Field_z_start.Value;
        app.xk=app.Field_x_end.Value;
        app.yk=app.Field_y_end.Value;
        app.zk=app.Field_z_end.Value;

        app.psi=app.Field_psi.Value;
        app.teta=app.Field_teta.Value;

        app.kroki = app.LiczbakrokwsymulacjiSpinner.Value;

        set(app.LiczbakrokwsymulacjiSpinner,'Enable','off');
        set(app.Field_l1,'Enable','off');
        set(app.Field_l2,'Enable','off');
        set(app.Field_l3,'Enable','off');
        set(app.Field_l4,'Enable','off');
        set(app.Field_l5,'Enable','off');
        set(app.Field_l6,'Enable','off');
        set(app.Field_d,'Enable','off');
        set(app.Field_e,'Enable','off');
        set(app.Field_psi,'Enable','off');
        set(app.Field_teta,'Enable','off');
        set(app.Field_x_start,'Enable','off');
        set(app.Field_x_end,'Enable','off');
        set(app.Field_y_start,'Enable','off');
        set(app.Field_y_end,'Enable','off');
        set(app.Field_z_start,'Enable','off');
        set(app.Field_z_end,'Enable','off');
        set(app.Field_d,'Enable','off');

```

```
set(app.DropDown_alfa1, 'Enable', 'off');  
set(app.DropDown_alfa2, 'Enable', 'off');  
set(app.DropDown_alfa5, 'Enable', 'off');  
set(app.SymulujButton, 'Enable', 'off');
```

```
oblicz(app);
```

```
set(app.LiczbakrokwsymulacjiSpinner, 'Enable', 'on');  
set(app.Field_l1, 'Enable', 'on');  
set(app.Field_l2, 'Enable', 'on');  
set(app.Field_l3, 'Enable', 'on');  
set(app.Field_l4, 'Enable', 'on');  
set(app.Field_l5, 'Enable', 'on');  
set(app.Field_l6, 'Enable', 'on');  
set(app.Field_d, 'Enable', 'on');  
set(app.Field_e, 'Enable', 'on');  
set(app.Field_psi, 'Enable', 'on');  
set(app.Field_teta, 'Enable', 'on');  
set(app.Field_x_start, 'Enable', 'on');  
set(app.Field_x_end, 'Enable', 'on');  
set(app.Field_y_start, 'Enable', 'on');  
set(app.Field_y_end, 'Enable', 'on');  
set(app.Field_z_start, 'Enable', 'on');  
set(app.Field_z_end, 'Enable', 'on');  
set(app.Field_d, 'Enable', 'on');  
set(app.DropDown_alfa1, 'Enable', 'on');  
set(app.DropDown_alfa2, 'Enable', 'on');  
set(app.DropDown_alfa5, 'Enable', 'on');  
set(app.SymulujButton, 'Enable', 'on');
```

```
end  
end
```