



**POLITECHNIKA  
RZESZOWSKA**  
im. IGNACEGO ŁUKASIEWICZA



**WYDZIAŁ  
BUDOWY MASZYN  
I LOTNICTWA**  
POLITECHNIKI RZESZOWSKIEJ

## **Geometria i kinematyka ząbów**

### **Laboratorium**

## **Ćwiczenie 4: Obróbka kształtowa narzędziem krążkowym cz. 1**

Opracował: dr inż. Michał Batsch

# 1 Wprowadzenie

W ramach ćwiczenia, na podstawie wiadomości z wykładu należy wygenerować powierzchnię działania narzędzia krążkowego do obróbki asymetrycznego uzębienia śrubowego. W pierwszej kolejności należy wygenerować zarys czołowy obrabianego uzębienia. W tym celu zostaną utworzone cztery m-pliki: *dane.m*, *definicja\_narzedzia.m*, *wrab.m*, *ewolwenta.m*, *powierzchnia.m* i jeden plik funkcji *row\_zaz.m*.

## 1.1 Plik „dane.m”

W tym pliku wprowadza się podstawowe dane uzębienia obrabianego. Można to zrealizować jak przedstawiono na rysunku 1.

```
%% Dane koła obrabianego
x=0.5;
xt=x;
z=19;
mn=3;
Beta=20*pi/180;
xn=xt/cos(Beta);
b=30;
mt=mn/cos(Beta);
m=mt;
d=mt*z;
r=d/2;
ha=mn;
hf=1.25*mn;
h=ha+hf;
da=d+2*ha+2*x*m;
ra=da/2;
df=d-2*hf+2*x*m;
rf=df/2;
jo=0.04*mn;
rt=r+x*m;
H=pi*2*r/tan(Beta);
```

Rys. 1 Definicja parametrów koła zębatego

W następnej kolejności definiowane są parametry strony biernej (rys. 2) i strony czynnej zęba (rys. 3)

```

%% --- Strona bierna
alfanc=20*pi/180;
alfatc=atan(tan(alfanc)/cos(Beta));
invalfatc=tan(alfatc)-alfatc;
gammatc=invalfatc;
dbc=d*cos(alfatc);
rbc=dbc/2;
sc=(pi*mt-j0)/2+2*x*mn*tan(alfatc);
ec=(pi*mt+j0)/2-2*x*mn*tan(alfatc);
fisc=sc/r;
alfaadc=acos(dbc/da);
gammaadc=tan(alfaadc)-alfaadc;
tetakc=sqrt(ra^2/rbc^2-1);

```

Rys. 2 Definicja parametrów strony biernej zęba

```

%% --- Strona czynna
alfand=30*pi/180;%30*pi/180;
alfatd=atan(tan(alfand)/cos(Beta));
invalfatd=tan(alfatd)-alfatd;
gammatd=invalfatd;
dbd=d*cos(alfatd);
rbd=dbd/2;
sd=(pi*mt-j0)/2+2*x*mn*tan(alfatd);
ed=(pi*mt+j0)/2-2*x*mn*tan(alfatd);
fisd=sd/r;
alfaad=acos(dbd/da);
gammaad=tan(alfaad)-alfaad;
tetakd=sqrt(ra^2/rbd^2-1);

```

Rys. 3 Definicja parametrów strony czynnej zęba

W dalszej kolejności obliczane są parametry zarysu odniesienia (początkowe i końcowe punkty odcinków narzędzia oraz promienie zaokrąglenia naroży) i zapis danych w postaci pliku *dane.mat* (Rys. 4).

```

%% Zarys odniesienia

roNc=0.2*mn;
roNd=0.2*mn;

xA1=-pi*m/4-j0/4-ha*tan(alfatc);
yA1=ha;

yA2=-hf+roNc-roNc*sin(alfatc);
xA2=-pi*m/4-j0/4+abs(yA2)*tan(alfatc);

xA3=xA2+roNc*cos(alfatc);
yA3=-hf;

yA5=-hf+roNd-roNd*sin(alfatd);
xA5=pi*m/4+j0/4-abs(yA5)*tan(alfatd);

xA4=xA5-roNd*cos(alfatd);
yA4=-hf;

yA6=ha;
xA6=xA5+(abs(yA5)+ha)*tan(alfatd);

%% Zapis danych

save dane.mat

```

Rys. 4 Definicja parametrów zarysu odniesienia i zapis danych

## 1.2 Plik „definicja\_narzedzia.m”

W tym pliku zostanie wygenerowany i wyeksportowany zarys narzędzia zębatkowego. W pierwszej kolejności ładowane są dane geometryczne zdefiniowane i zapisane uprzednio w pliku *dane.mat*. Następnie definiowane są w sposób parametryczny współrzędne punktów poszczególnych odcinków zarysu narzędzia (Rys. 5).

```
clc
clear
close all

load dane.mat

%% 1 odcinek narzedzia
t1p=0;
t1k=(abs(yA1)+abs(yA2))/cos(alfatc);
t1=t1p:0.01:t1k;
x1=xA1+t1*sin(alfatc);
y1=yA1-t1*cos(alfatc);
%% 2 odcinek narzedzia
t2p=pi+alfatc;
t2k=3/2*pi;
t2=t2p:0.01:t2k;
x2=roNc*cos(t2)+xA2+roNc*cos(alfatc);
y2=roNc*sin(t2)+yA3+roNc;
%% 3 odcinek narzedzia
t3p=0;
t3k=xA4-xA3;
t3=t3p:0.01:t3k;
x3=xA3+t3;
y3=yA3.*ones(1,length(t3));
%% 4 odcinek narzedzia
t4p=3/2*pi;
t4k=3/2*pi+pi/2-alfatd;
t4=t4p:0.01:t4k;
x4=roNd*cos(t4)+xA4;
y4=roNd*sin(t4)+yA4+roNd;
%% 5 odcinek narzedzia
t5p=0;
t5k=(abs(yA5)+abs(yA6))/cos(alfatd);
t5=0:0.01:t5k;
x5=xA5+t5*sin(alfatd);
y5=yA5+t5*cos(alfatd);
%% zarys całego narzedzia

ZAR_N_X=[x1 x2 x3 x4 x5];
ZAR_N_Y=[y1 y2 y3 y4 y5];

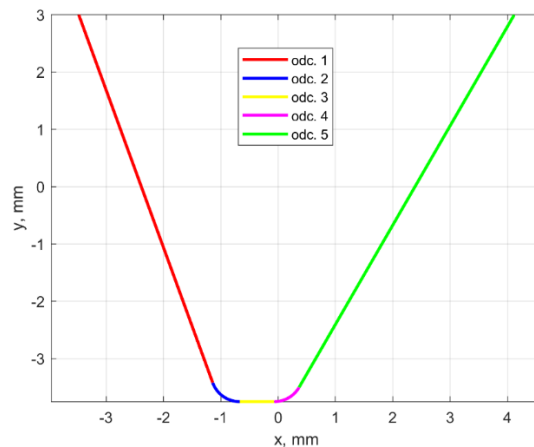
save('ZAR_N_X.mat','ZAR_N_X');
save('ZAR_N_Y.mat','ZAR_N_Y');
```

Rys. 5 Definicja zarysu narzędzia

Narzędzie składa się z pięciu fragmentów:

- 1 – prosta (strona bierna),
- 2 – zaokrąglone naroże (strona bierna),
- 3 – wierzchołek narzędzia zębatki,
- 4 – zaokrąglone naroże (strona czynna),
- 5 – prosta (strona czynna)

Narzędzie powinno wyglądać w sposób zbliżony do tego przedstawionego na rys. 6.



Rys. 6 Zdefiniowany zarys narzędzia zębatkowego

### 1.3 Plik „wrab.mat”

W tym pliku zgodnie z informacjami z wykładu zostanie wygenerowany wrąb. Całość sprowadza się do kodu przedstawionego na rysunku 7.

```
clc
clear
close all

load dane.mat
load ZAR_N_X.mat
load ZAR_N_Y.mat

X=ZAR_N_X;
Y=ZAR_N_Y;

dx=(X(length(X))-X(1))/2000;
xNN=X(1):dx:X(length(X));
yNN=interp(X,Y,xNN);

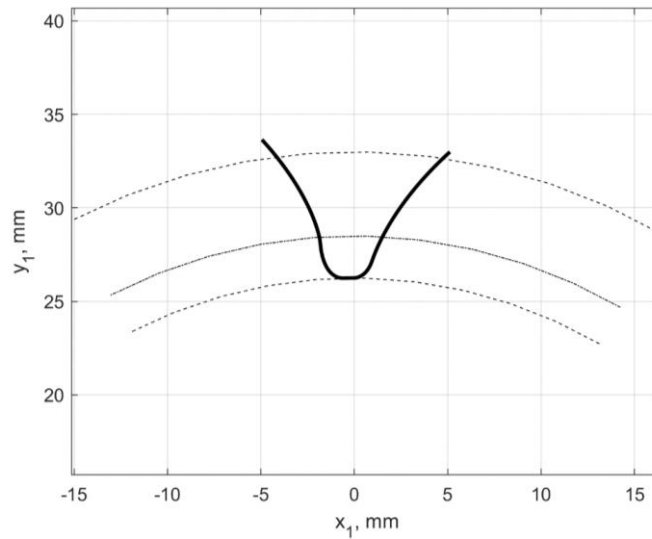
dxNN=diff(xNN);
dyNN=diff(yNN);
xNN=xNN(1:end-1);
yNN=yNN(1:end-1);

fi=(xNN+yNN.*dyNN./dxNN)/rt;
x11=cos(fi).*(xNN - fi*rt) + sin(fi).*(rt + yNN);
y11=cos(fi).*(rt + yNN) - sin(fi).*(xNN - fi*rt);

figure
plot(x11,y11,'-')
grid on;
axis equal;
```

Rys. 7 Generowanie wrąbu

Po wczytaniu punktów zarysu następuje interpolacja. W tym fragmencie kodu można zdecydować np. o zwiększeniu liczby punktów na zarysie oraz o rodzaju interpolacji (domyślnie - liniowa). W przykładzie przyjęto liczbę punktów 2000. Wygenerowany wrąb przedstawiono na rysunku 8.



Rys. 8 Wygenerowany wrąb

#### 1.4 Plik „evolwenta.m”

W tym pliku wygenerowany zarys czołowy zostanie interpolowany zadaną liczbą punktów w równych odległościach mierzonych wzdłuż krzywej. Można to zrealizować kodem przedstawionym na rysunku 9.

```

close all
clc
clear all

load dane.mat
load x11.mat
load y11.mat

x=x11;
y=y11;

rsymmmod=rot2na2(-pi/2)*[x;y];
xmod=rsymmmod(1,:);
ymod=rsymmmod(2,:);

pt = interparc(500,xmod,ymod,'spline');

xmod=pt(:,1);
ymod=pt(:,2);

dxmod=diff(xmod);
dymod=diff(ymod);

% dane kształtowego narzędzia krążkowego
delta=0;
rN=100-delta;
aN=rN+rN;
sigma=Beta;

save dane.mat

```

Rys. 9 Interpolacja ze stałymi odległościami

Wykorzystano istniejącą funkcję *interparc* dostępną na stronie:

<https://www.mathworks.com/matlabcentral/fileexchange/34874-interparc>

Ważne jest, aby plik funkcji znajdował się w katalogu roboczym MATLABA (tym, w którym tworzone są pliki).

## 1.5 Plik „powierzchnia.m”

W tym pliku zostanie wygenerowana powierzchnia działania i linia styku narzędzia z uzębieniem obrabianym. W pierwszej kolejności (Rys. 10) wczytywane są dane, definiowane są zmienne globalne (przekazywane do zewnętrznej funkcji równania zazębienia) oraz generowany jest zaimportowany zarys czołowy.

```

close all
clc
clear all
tic
load dane.mat

global Beta dXmod dYmod Xmod Ymod r aIN

figure
plot(xmod,ymod,'b')
hold on
fi=0:0.001:2*pi;
plot(df/2*cos(fi),df/2*sin(fi),'--')
grid on
plot(d/2*cos(fi),d/2*sin(fi),'-')
hold on
plot(da/2*cos(fi),da/2*sin(fi),'--')
grid on
axis equal
title 'Zarys czoiłowy koła obrabianego'

```

Rys. 10 Wczytywanie danych do obliczeń

W następnej kolejności generowany jest obrabiany wręb i walec podziałowy (Rys. 11).

```

imax=length(dxmod);
j=1;
dteta=0.001;
tetak=b/r*tan(Beta);

for teta=-tetak/2:dteta:tetak/2;
    for i=1:imax;
        rkk=[cos(teta) -sin(teta) 0 0; sin(teta) cos(teta) 0 0; 0 0 1 H/(2*pi)*teta; 0 0 0 1]*[xmod(i);ymod(i);0;1];
        xkk(i,j)=rkk(1);
        ykk(i,j)=rkk(2);
        zkk(i,j)=rkk(3);

        end
        i=i+1;
        j=j+1;
    end

    i=1;
    j=1;
    for par=0:0.01:2*pi
        for dl=-b/2:b/2
            okrkk=[r*cos(par);r*sin(par);dl;1];
            xokrkk(i,j)=okrkk(1);
            yokrkk(i,j)=okrkk(2);
            zokrkk(i,j)=okrkk(3);

            i=i+1;
            end
            i=1;
            j=j+1;
        end

        figure
        surf(xkk,ykk,zkk,'FaceColor','red')
        hold on
        surf(xokrkk,yokrkk,zokrkk,'FaceColor','red')
        hold on
        title 'Obrabiany wręb na tle walca podziałowego'

```

Rys. 11 Generowanie obrabianego wrębu i walca podziałowego

Następnie rozwiązywane jest równanie zazębienia dla wszystkich dyskretnej wartości parametru zarysu (zarys dany jest w sposób dyskretny, dlatego jego parametr można interpretować jako licznik  $i$ ), co pokazano na rys. 12.

```

i=1;
j=1;
for i=1:imax
    Xmod=xmod(i);
    Ymod=ymod(i);

    dXmod=dxmod(i);
    dYmod=dymod(i);
    x0 = 0;
    options=optimset('Display','iter','MaxFunEvals',2000,'MaxIter',2000);
    [x,fval] = fsolve(@row_zaz,x0,options);
    teta=x(1);
    TETA(i)=teta;
end

```

Rys. 12 Rozwiązywanie równania zazębienia

W funkcji *fsolve* przywołujemy zewnętrzną funkcję *row\_zaz*, w której zdefiniowana jest postać równania. Kod pliku *row\_zaz.m* przedstawiono na rys. 13.

```

function F = row_zaz(x)
global Beta dXmod dYmod Xmod Ymod r alN
teta=x(1);

F=-cos(Beta)*(dYmod*Ymod^2*sin(teta) - dXmod*Xmod^2*cos(teta)...
+ alN*dXmod*Xmod + alN*dYmod*Ymod + dXmod*r*Xmod + dYmod*r*Ymod...
- alN*dXmod*r*cos(teta) + alN*dYmod*r*sin(teta)...
- dYmod*Xmod*Ymod*cos(teta) + dXmod*Xmod*Ymod*sin(teta)...
+ dYmod*r^2*teta*cot(Beta)^2*cos(teta) + dXmod*r^2*teta*cot(Beta)^2*sin(teta));
end

```

Rys. 13 Funkcja równania zazębienia

Dalej generowana jest powierzchnia działania narzędzia (Rys. 14).

```

i=1;
j=1;
MN1=[1 0 0 -a1N; 0 cos(-pi/2+Beta) -sin(-pi/2+Beta) 0; 0 sin(-pi/2+Beta) cos(-pi/2+Beta) 0; 0 0 0 1];
for tetaN=-pi/10:0.01:pi/10;
MN0N=[cos(tetaN) sin(tetaN) 0 0; -sin(tetaN) cos(tetaN) 0 0; 0 0 1 0; 0 0 0 1];
MN01=MN0N*MN1;
for i=1:imax
Xmod=xmod(i);
Ymod=ymod(i);

teta=TETA(i);
rNN=MN01*[cos(teta) -sin(teta) 0 0; sin(teta) cos(teta) 0 0; 0 0 1 H/(2*pi)*teta; 0 0 0 1]*[Xmod;Ymod;0;1];
xNN(i,j)=rNN(1);
yNN(i,j)=rNN(2);
zNN(i,j)=rNN(3);

end
i=i+1;
j=j+1;
end

figure
surf(xNN,yNN,zNN,'FaceColor','red')
grid on
axis equal

```

Rys. 14 Generowanie powierzchni działania narzędzia

Ostatnim etapem jest wyznaczenie linii styku i zarysu narzędzia (Rys. 15)

```

% linia styku
tetaN=0;
MN0N=[cos(tetaN) sin(tetaN) 0 0; -sin(tetaN) cos(tetaN) 0 0; 0 0 1 0; 0 0 0 1];
MN01=MN0N*MN1;
MIN=[1 0 0 a1N; 0 cos(pi/2-Beta) -sin(pi/2-Beta) 0; 0 sin(pi/2-Beta) cos(pi/2-Beta) 0; 0 0 0 1];
for i=1:imax
Xmod=xmod(i);
Ymod=yymod(i);
teta=TETA(i);
rNLS=MN01*[cos(teta) -sin(teta) 0 0; sin(teta) cos(teta) 0 0; 0 0 1 H/(2*pi)*teta; 0 0 0 1]*[Xmod;Ymod;0;1];
xNLS(i)=rNLS(1);
yNLS(i)=rNLS(2);
zNLS(i)=rNLS(3);

r1LS=[cos(teta) -sin(teta) 0 0; sin(teta) cos(teta) 0 0; 0 0 1 H/(2*pi)*teta; 0 0 0 1]*[Xmod;Ymod;0;1];
x1LS(i)=r1LS(1);
y1LS(i)=r1LS(2);
z1LS(i)=r1LS(3);
end

figure
plot3(xNLS,yNLS,zNLS, '.')
grid on
axis equal

% zarys narzędzia
zN=zNLS;
xN=-sqrt(xNLS.^2+yNLS.^2);

figure
plot(zN,xN, '-')
grid on
axis equal
xlabel 'zNN'
ylabel 'xNN'

```

Rys. 15 Wyznaczenie linii styku i zarysu narzędzia

## 2 Zadanie

W ramach ćwiczenia, należy samodzielnie napisać program, który będzie generował linię styku narzędzia i jego zarys.

Podpowiedzi:

- pliki należy uruchamiać w takiej kolejności w jakiej były opisane w niniejszej instrukcji,
- pliki funkcji *interparc* i *row\_zaz* muszą być w tym samym folderze roboczym.