POLITECHNIKA RZESZOWSKA im. IGNACEGO ŁUKASIEWICZA

KKM

WYDZIAŁ BUDOWY MASZYN I LOTNICTWA POLITECHNIKI RZESZOWSKIEJ

# Geometria i kinematyka zazębień

# Laboratorium

# Ćwiczenie 5: Analiza styku ewolwentowego zewnętrznego zazębienia płaskiego

Opracował: dr inż. Michał Batsch

# 1 Wprowadzenie

W ramach ćwiczenia, na podstawie wiadomości z wykładu należy przeprowadzić analizę styku zębów w płaskim ewolwentowym zazębieniu zewnętrznym. W szczególności należy wyznaczyć: linię zazębienia (przyporu), błąd ruchu i rozkład prędkości poślizgu. W tym celu zostaną utworzone trzy m-pliki: *dane.m*, *tooth_contact.m*, i jeden plik funkcji *TCA.m*.

## 1.1 Plik „dane.m”

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

```
%% Podstawowe dane kół zębatych
m=3;
alfa=20*pi/180;
z1=17;
z2=35;

a0=0.5*m*(z1+z2);
ar=80;

invalfa=tan(alfa)-alfa;
alfaw=acos(a0/ar*cos(alfa));
invalfaw=tan(alfaw)-alfaw;

x1px2=(z1+z2)*(invalfaw-invalfa)/(2*tan(alfa));
x1=x1px2/2;
x2=x1px2/2;
ap=a0+x1px2*m;
km=ap-ar;
```

Rys. 1 Podstawowe parametrów pary zębatej

W następnej kolejności definiowane są parametry zębnika (Rys. 2) i koła (Rys. 3).

```
%% Koło 1
ha1=m-km+x1*m;
hf1=1.25*m-x1*m;
h1=ha1+hf1;

d1=m*z1;
r1=d1/2;
da1=d1+2*ha1;
ra1=da1/2;
df1=d1-2*hf1;
rf1=df1/2;

db1=d1*cos(alfa);
rb1=db1/2;
s1=pi*m/2+2*x1*m*tan(alfa);
e1=pi*m/2-2*x1*m*tan(alfa);
fis1=s1/r1;
alfaa1=acos(db1/da1);
invalfaa1=tan(alfaa1)-alfaa1;
rw1=rb1/cos(alfaw);
dw1=2*rw1;
invalfaat1=tan(alfaa1)-alfaa1;
sa1=da1*(s1/d1+invalfa-invalfaat1);
ta1=invalfaa1+alfaa1;
fisym1=invalfa+s1/r1/2;
```

Rys. 2 Definicja parametrów zębnika

```
%% Koło 2
ha2=m-km+x2*m;
hf2=1.25*m-x2*m;
h2=ha2+hf2;

d2=m*z2;
r2=d2/2;
da2=d2+2*ha2;
ra2=da2/2;
df2=d2-2*hf2;
rf2=df2/2;
db2=d2*cos(alfa);
rb2=db2/2;
s2=pi*m/2+2*x2*m*tan(alfa);
e2=pi*m/2-2*x2*m*tan(alfa);
fis2=s2/r2;
alfaa2=acos(db2/da2);
invalfaa2=tan(alfaa2)-alfaa2;
tetak2=sqrt(ra2^2/rb2^2-1);
rw2=rb2/cos(alfaw);
dw2=2*rw2;
invalfaat2=tan(alfaa2)-alfaa2;
sa2=da2*(s2/d2+invalfa-invalfaat2);
alfaF2=tan(alfa)-4*(1-x2)/(z2*sin(alfa));
ta2=invalfaa2+alfaa2;
fisym2=invalfa-e2/r2/2;
```

Rys. 3 Definicja parametrów koła

W dalszej kolejności obliczane są parametry zazębienia, między innymi kątowe położenie początkowych i końcowych punktów odcinka przyporu czy wskaźnik zazębienia oraz następuje zapis danych w postaci pliku *dane.mat* (Rys. 4).

```
%% Zazębienie
C1=1/(2*pi)*sqrt((1+2*ha1/d1)^2*(1+tan(alfa)^2)-1);
C2=1/(2*pi)*sqrt((1+2*ha2/d2)^2*(1+tan(alfa)^2)-1);
C3=ar/(pi*m)*sin(alfaw)/cos(alfa);
ea=z1*C1+z2*C2-C3;

E2C=sqrt(ra1^2-rb1^2)-rw1*sin(alfaw);
E1C=sqrt(ra2^2-rb2^2)-rw2*sin(alfaw);

filE2=acos((ra1^2+rw1^2-E2C^2)/(2*ra1*rw1));
fi2E1=acos((ra2^2+rw2^2-E1C^2)/(2*ra2*rw2));

rfo2=sqrt(ar^2+ra1^2-2*ar*ra1*cos(filE2));
rfo1=sqrt(ar^2+ra2^2-2*ar*ra2*cos(fi2E1));

fi2E2=asin(E2C/rfo2*sin(pi/2-alfaw));
filE1=asin(E1C/rfo1*sin(pi/2-alfaw));

alfalE1=acos(rw1/rfo1*cos(alfaw));
invalfalE1=tan(alfalE1)-alfalE1;

filp=-(filE1+(invalfa-invalfalE1)+fis1/2);
filk=filE2+invalfaat1-invalfa-fis1/2;

dfo1=2*rfo1;
alfafo1=acos(db1/dfo1);
invalfatfo1=tan(alfafo1)-alfafo1;
tfo1=invalfatfo1+alfafo1;

dfo2=2*rfo2;
alfafo2=acos(db2/dfo2);
invalfatfo2=tan(alfafo2)-alfafo2;
tfo2=invalfatfo2+alfafo2;

save dane.mat
```

Rys. 4 Definicja parametrów zazębienia i zapis danych

## 1.2 Plik „tooth_contact.m”

W tym pliku zostanie rozwiązany układ równań zazębienia. W pierwszej kolejności wczytywane są dane geometryczne zdefiniowane i zapisane uprzednio w pliku *dane.mat* (Rys. 5).

```
close all
clear all
clc

load dane.mat

dfil=(filk-filp)/(800-1);
da=0;

global rb1 rb2 fisym1 fisym2 fil da ar
```

Rys. 5 Wczytywanie danych geometrycznych

Ponadto obliczany jest krok dyskretyzacji kąta obrotu zębnika, definiowany błąd rozstawienia osi oraz deklarowane są zmienne globalne – wykorzystywane w dalszej części przez funkcję z układem równań. Następnie (Rys. 6) w pętli rozwiązywany jest układ równań zazębienia.

```
%% Rozwiązanie równania zazębienia
i=1;
for fil=filp:dfil:filk

x0=[(tal-tfol)/(filk-filp)*(fil-filp)+tfol; (tfo2-ta2)/(filk-filp)*(fil-filp)+ta2; fil*z1/z2];
options=optimset('Display','iter','MaxFunEvals',2000,'MaxIter',2000);
[x,fval] = fsolve(@TCA,x0,options);
t1=x(1);
t2=x(2);
fi2=x(3);

if t1<=tal && t2<=ta2

T1(i)=t1;
T2(i)=t2;

FI1(i)=fil;
FI2(i)=fi2;

dr1fdfilx(i) = rb1*(cos(fil + fisym1 - t1) - t1*sin(fil + fisym1 - t1));
dr1fdfily(i) = -rb1*(sin(fil + fisym1 - t1) + t1*cos(fil + fisym1 - t1));

dr2fdfi2x(i)=rb2*(cos(fi2 - fisym2 + t2) + t2*sin(fi2 - fisym2 + t2));
dr2fdfi2y(i)=rb2*sin(fi2 - fisym2 + t2) - rb2*t2*cos(fi2 - fisym2 + t2);

LS1x(i)=rb1*(sin(fil + fisym1 - t1) + t1*cos(fil + fisym1 - t1));
LS1y(i)=rb1*(cos(fil + fisym1 - t1) - t1*sin(fil + fisym1 - t1));

LS2x(i)=rb2*(sin(fi2 - fisym2 + t2) - t2*cos(fi2 - fisym2 + t2));
LS2y(i)=ar+da - rb2*cos(fi2 - fisym2 + t2) - rb2*t2*sin(fi2 - fisym2 + t2);

DR1x(i)=rb1*cos(fisym1 - pi/2)*(cos(t1) + t1*sin(t1)) + rb1*sin(fisym1 - pi/2)*(sin(t1) - t1*cos(t1));
DR1y(i)=rb1*cos(fisym1 - pi/2)*(sin(t1) - t1*cos(t1)) - rb1*sin(fisym1 - pi/2)*(cos(t1) + t1*sin(t1));

i=i+1;
end
end
```

Rys. 6 Rozwiązanie układu równań

W tym celu wykorzystana została funkcja fsolve. Jeżeli obliczone parametry zarysów (kąty odtaczania) t1 i t2 są mniejsze od wartości granicznych (od kątów odtaczania na średnicach wierzchołkowych) obliczane są współrzędne: linii zazębienia, drogi punktu styku na zarysie zęba zębnika oraz pochodne wektorów wodzących względem kątów obrotu. Po zakończeniu pracy w pętli obliczane są prędkości poślizgu (Rys. 7) na podstawie zdefiniowanej prędkości kątowej zębnika om.

```
%% Obliczanie prędkości poślizgu
n=i-1;
oml=100*2*pi/60;
dfi1=diff(FI1);
dfi2=diff(FI2);
dt=dfi1/oml;

dfildt=dfi1./dt;
vtr1x=dr1fdfi1x(1:n-1).*dfildt;
vtr1y=dr1fdfi1y(1:n-1).*dfildt;

dfi2dt=dfi2./dt;
vtr2x=dr2fdfi2x(1:n-1).*dfi2dt;
vtr2y=dr2fdfi2y(1:n-1).*dfi2dt;

v12x=vtr1x-vtr2x;
v12y=vtr1y-vtr2y;
v12=sqrt(v12x.^2+v12y.^2);
[v12min,imin]=min(v12);
```

Rys. 7 Obliczanie prędkości poślizgu

Wizualizacji uzyskanych wyników dokonuje się w kolejnym kroku podzielonym na części dotyczące zębnika (Rys. 8), koła (Rys.9) i linii styku (Rys. 10).

```
%% Wykres zębnik
t1=tfo1:(ta1-tfo1)/100:ta1;
figure
fi1=FI1(1);
x1f = rb1*(sin(fi1 + fisym1 - t1) + t1.*cos(fi1 + fisym1 - t1));
y1f = rb1*(cos(fi1 + fisym1 - t1) - t1.*sin(fi1 + fisym1 - t1));
plot(x1f,y1f,'-r','Linewidth',1.5)
hold on
fi1=FI1(imin);
x1f = rb1*(sin(fi1 + fisym1 - t1) + t1.*cos(fi1 + fisym1 - t1));
y1f = rb1*(cos(fi1 + fisym1 - t1) - t1.*sin(fi1 + fisym1 - t1));
plot(x1f,y1f,'-r','Linewidth',1.5)
hold on
fi1=FI1(end);
x1f = rb1*(sin(fi1 + fisym1 - t1) + t1.*cos(fi1 + fisym1 - t1));
y1f = rb1*(cos(fi1 + fisym1 - t1) - t1.*sin(fi1 + fisym1 - t1));
plot(x1f,y1f,'-r','Linewidth',1.5)
hold on
fi=50*pi/180:0.001:(180-50)*pi/180;
plot(dfo1/2*cos(fi),dfo1/2*sin(fi),'--r')
hold on
plot(dw1/2*cos(fi),dw1/2*sin(fi),'-.r')
hold on
plot(da1/2*cos(fi),da1/2*sin(fi),'--r')
grid on
axis equal
hold on
```

Rys. 8 Wizualizacja wyników - zębnik

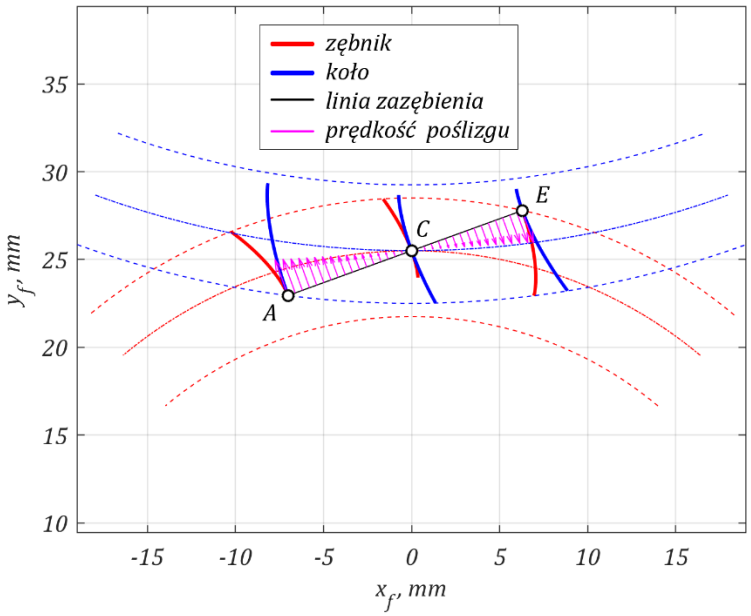
```
%% Wykres koło
t2=tfo2:(ta2-tfo2)/100:ta2;
fi2=FI2(1);
x2f = rb2*(sin(fi2 - fisym2 + t2) - t2.*cos(fi2 - fisym2 + t2));
y2f = ar+da - rb2*cos(fi2 - fisym2 + t2) - rb2*t2.*sin(fi2 - fisym2 + t2);
plot(x2f,y2f,'-b','Linewidth',1.5)
hold on
fi2=FI2(imin);
x2f = rb2*(sin(fi2 - fisym2 + t2) - t2.*cos(fi2 - fisym2 + t2));
y2f = ar+da - rb2*cos(fi2 - fisym2 + t2) - rb2*t2.*sin(fi2 - fisym2 + t2);
plot(x2f,y2f,'-b','Linewidth',1.5)
hold on
fi2=FI2(end);
x2f = rb2*(sin(fi2 - fisym2 + t2) - t2.*cos(fi2 - fisym2 + t2));
y2f = ar+da - rb2*cos(fi2 - fisym2 + t2) - rb2*t2.*sin(fi2 - fisym2 + t2);
plot(x2f,y2f,'-b','Linewidth',1.5)
hold on
fi=(180+70)*pi/180:0.001:(360-70)*pi/180;
plot(dfo2/2*cos(fi),dfo2/2*sin(fi)+ar+da,'--b')
hold on
plot(dw2/2*cos(fi),dw2/2*sin(fi)+ar+da,'-.b')
hold on
plot(da2/2*cos(fi),da2/2*sin(fi)+ar+da,'--b')
grid on
axis equal
```

Rys. 9 Wizualizacja wyników – koło

```
%% Wykres linia zazębienia
hold on
plot(LS1x,LS1y,'-k')
hold on
quiver([LS1x(1:18:n-1),LS1x(n-1)],[LS1y(1:18:n-1),LS1y(n-1)],[vl2x(1:18:end),vl2x(end)],[vl2y(1:18:end),vl2y(end)],'-m','filled')
hold on
plot([LS1x(1),LS1x(imin),LS1x(end)],[LS1y(1),LS1y(imin),LS1y(end)],'ok',...
    'LineWidth',1,...
    'MarkerEdgeColor','k',...
    'MarkerFaceColor','white',...
    'MarkerSize',5)
axis equal
xlabel 'x_f , mm'
ylabel 'y_f , mm'
```

Rys. 10 Wizualizacja wyników – linia zazębienia

Przykładowe wyniki działania programu przedstawiono na rys. 11.




Rys. 11 Przykładowe wyniki

Ostatnim etapem jest obliczenie błędu ruchu i wygenerowanie jego wykresu (Rys. 12).

```
%% Błąd ruchu i liczba przyporu
licznik=1;
for n=2:length(T1)
  FI1(licznik)=FI1(n);
  dfi2(licznik)=FI2(n)-FI2(n-1);
  dfi1(licznik)=FI1(n)-FI1(n-1);

  dlx(licznik)=LS1x(n)-LS1x(n-1);
  dly(licznik)=LS1y(n)-LS1y(n-1);
  dl(licznik)=sqrt(dlx(licznik).^2+dly(licznik).^2);
  licznik=licznik+1;
end
Ls=sum(dl);
eaTCA=Ls/(pi*m*cos(alfa));

figure
plot(FI1*180/pi,(dfi2-dfi2(1))*180/pi*3600,'-r')
hold on
plot((FI1+2*pi/z1)*180/pi,(dfi2-dfi2(1))*180/pi*3600,'-m')
hold on
plot((FI1+2*2*pi/z1)*180/pi,(dfi2-dfi2(1))*180/pi*3600,'-b')
xlabel('fi1 , deg');
ylabel('dfi2 , arcsek');
grid on
ylabel('$\Delta\varphi_2$ , arcsek','FontAngle','italic','FontSize',11,...
    'FontName','cambria',...
    'Interpreter','latex');
xlabel('$\varphi_1$ , deg','FontAngle','italic','FontSize',11,...
    'FontName','cambria',...
    'Interpreter','latex');
```

Rys. 12 Obliczanie błędu ruchu i liczby przyporu

### 1.3 Plik „TCA.mat”

W tym pliku zawarta jest funkcja z układem równań zazębienia (Rys. 13).

```
function F = TCA(x)
global rb1 rb2 fisym1 fisym2 fi1 da ar

t1=x(1);
t2=x(2);
fi2=x(3);

r1f=[rb1*(sin(fi1 + fisym1 - t1) + t1*cos(fi1 + fisym1 - t1));
     rb1*(cos(fi1 + fisym1 - t1) - t1*sin(fi1 + fisym1 - t1))];

r2f=[rb2*(sin(fi2 - fisym2 + t2) - t2*cos(fi2 - fisym2 + t2));
     ar+da - rb2*cos(fi2 - fisym2 + t2) - rb2*t2*sin(fi2 - fisym2 + t2)];

n1f=[cos(fi1 + fisym1 - t1);
    -sin(fi1 + fisym1 - t1)];

n2f=[-cos(fi2 - fisym2 + t2);
     -sin(fi2 - fisym2 + t2)];

ROW1=r1f(1)-r2f(1);
ROW2=r1f(2)-r2f(2);
ROW3=-n1f(2)-n2f(2);

F=[ROW1;ROW2;ROW3];
```

Rys. 13 Plik funkcji układu równań zazębienia

## 2 Zadanie

W ramach ćwiczenia, należy samodzielnie napisać program, który będzie wyznaczał linię zazębienia, błąd ruchu oraz rozkład prędkości poślizgu. Ponadto należy wyznaczyć wpływ błędu rozstawienia osi oraz współczynników przesunięcia zarysu na zmianę charakteru współpracy zębów – na błąd ruchu i wskaźnik zazębienia.

Podpowiedzi:

- pliki należy uruchamiać w takiej kolejności w jakiej były opisane w niniejszej instrukcji,
- plik funkcji *TCA* musi być w tym samym folderze roboczym,
- przeprowadzić symulację kilkukrotnie dla przekładni, w której ar=a0 (bez korekcji) zmieniając błąd rozstawienia np. da=0, da=0.1 itd. sprawdzając jaki jest wskaźnik zazębienia i błąd ruchu.
- przeprowadzić symulację kilkukrotnie dla przekładni, w której da=0, zmieniając współczynniki przesunięcia zarysu np. poprzez zmianę odległości osi ar=a0, ar=79, ar=80 lub poprzez zmianę ich rozdziału na poszczególne koła.