****

**Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie**

WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ

**Sprawozdanie Robotyka Kosmiczna**

Autor:  
 **Dawid Lisek**

Kierunek studiów: **Automatyka i Robotyka**

Specjalizacja: **Komputerowe Systemy Sterowania**Grupa: pn 13:15 - 15:45

Korzystając z warunków brzegowych należało wyznaczyć 10 równań, które posłużyły do wyliczenia współczynników: 𝑎0, 𝑎1, 𝑎2, 𝑎3, 𝑏0, 𝑏1, 𝑐0, 𝑐1, 𝑐2, 𝑐3.

Równania zostały wyznaczone na podstawie poniższych warunków brzegowych, które muszą spełniać równania poszczególnych faz ruchu.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Równania opisujące poszczególne fazy ruchu:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Następnie należało zapisać te równania w programie Matlab przy użyciu przybornika Symbolic Math Toolbox oraz wyznaczyć współczynniki przy pomocy funkcji solve().

Zapisane równania w języku Matlab:

clear all

syms a\_0 a\_1 a\_2 a\_3 b\_0 b\_1 c\_0 c\_1 c\_2 c\_3

syms q\_p q\_k v

syms t\_a t\_c t\_k t\_b

syms d\_qk d\_qp

t\_c = t\_k-t\_b;

A = a\_0 - q\_p

A = 

B = a\_1 - d\_qp

B = 

C = a\_0 + a\_1\*t\_a + a\_2\*(t\_a)^2 + a\_3\*(t\_a)^3 - b\_0 - b\_1\*t\_a

C = 

D = a\_1 + 2\*a\_2\*t\_a + 3\*a\_3\*(t\_a)^2 - b\_1

D = 

E = a\_1 + 2\*a\_2\*t\_a + 3\*a\_3\*(t\_a)^2 - c\_1 - 2\*c\_2\*t\_c - 3\*c\_3\*(t\_c)^2

E = 

F = b\_1 - v

F = 

G = b\_0 + b\_1\*t\_c - c\_0 - c\_1\*t\_c - c\_2\*(t\_c)^2 - c\_3\*(t\_c)^3

G = 

H = b\_1 - c\_1 - 2\*c\_2\*t\_c - 3\*c\_3\*(t\_c)^2

H = 

I = c\_0 + c\_1\*t\_k + c\_2\*(t\_k)^2 + c\_3\*(t\_k)^3 -q\_k

I = 

J = c\_1 + 2\*c\_2\*t\_k + 3\*c\_3\*(t\_k)^2 - d\_qk

J = 

Rozwiązane równania przy pomocy funkcji solve:

[Z] = solve(A==0, B==0, C==0, D==0, E==0, F==0, G==0, H==0, I==0, J==0 ,a\_0, a\_1, a\_2, a\_3, b\_0, b\_1, c\_0, c\_1, c\_2, c\_3)

Z = *struct with fields:*

a\_0: q\_p

a\_1: d\_qp

a\_2: -(6\*q\_p - 6\*q\_k + 4\*d\_qp\*t\_a + 3\*d\_qk\*t\_b - 4\*t\_a\*v - 3\*t\_b\*v + 6\*t\_k\*v)/(2\*t\_a^2)

a\_3: (2\*q\_p - 2\*q\_k + d\_qp\*t\_a + d\_qk\*t\_b - t\_a\*v - t\_b\*v + 2\*t\_k\*v)/t\_a^3

b\_0: q\_k - (d\_qk\*t\_b)/2 + (t\_b\*v)/2 - t\_k\*v

b\_1: v

c\_0: (2\*q\_k\*t\_b + d\_qk\*t\_k^2 - t\_k^2\*v - 2\*d\_qk\*t\_b\*t\_k)/(2\*t\_b)

c\_1: (d\_qk\*t\_b - d\_qk\*t\_k + t\_k\*v)/t\_b

c\_2: (d\_qk - v)/(2\*t\_b)

c\_3: 0

Kolejnym krokiem było napisanie funkcji Trajectory\_Generation do generowania trajektorii przegub według poniższego schematu:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Utworzona funkcja Trajectory Generation:

function [q,dq,ddq] = Trajectory\_Generation(solved, q\_p\_in,dq\_p\_in,q\_k\_in,dq\_k\_in,Tk\_in,Ta\_in,Tb\_in,dt\_in,V\_in)

syms q\_p q\_k v

syms t\_a t\_c t\_k t\_b

syms d\_qk d\_qp

val = subs(solved,[q\_p, d\_qp, q\_k, d\_qk, t\_k, t\_a, t\_b, v],[q\_p\_in dq\_p\_in q\_k\_in dq\_k\_in Tk\_in Ta\_in Tb\_in V\_in]);

t1 = 0:dt\_in:Ta\_in;

q\_1 = val.a\_0 + val.a\_1 \* t1 + val.a\_2 \* (t1).^2 + val.a\_3 \* (t1).^3;

d\_q1 = val.a\_1 + 2\*val.a\_2\*t1 + 3\*val.a\_3 \* (t1).^2;

dd\_q1 = 2\*val.a\_2 + 6\*val.a\_3\*t1;

t2 = Ta\_in:dt\_in:Tb\_in;

q\_2 = val.b\_0 + val.b\_1 \* t2;

d\_q2 = val.b\_1 + zeros(1, length(t2));

dd\_q2 = zeros(1, length(t2));

t3 = Tb\_in:dt\_in:Tk\_in;

q\_3 = val.c\_0 + val.c\_1 \* t3 + val.c\_2 \* (t3).^2 + val.c\_3 \* (t3).^3;

d\_q3 = val.c\_1 + 2\*val.c\_2\*t3 + 3\*val.c\_3 \* (t3).^2;

dd\_q3 = 2\*val.c\_2 + 6\*val.c\_3\*t3;

t = [0:dt\_in:Tk\_in];

q = [q\_1(1:end-1), q\_2(1:end-1), q\_3];

dq = [d\_q1(1:end-1), d\_q2(1:end-1), d\_q3];

ddq = [dd\_q1(1:end-1), dd\_q2(1:end-1), dd\_q3];

figure;

subplot(1, 3, 1);

plot(t1, q\_1, t2, q\_2, t3, q\_3, t1(1), q\_p\_in, 'o', Tk\_in, q\_k\_in, 'o')

legend('Faza I', 'Faza II', 'Faza III', 'Pozycja początkowa', 'Pozycja końcowa')

title('Pozycja')

grid on

axis auto

subplot(1, 3, 2);

plot(t1, d\_q1, t2, d\_q2, t3, d\_q3, t1(1), dq\_p\_in, 'o', Tk\_in, dq\_k\_in, 'o')

legend('Faza I', 'Faza II', 'Faza III', 'Prędkość początkowa', 'Prędkość końcowa')

title('Prędkość')

grid on

axis auto

subplot(1, 3, 3);

plot(t1, dd\_q1, t2, dd\_q2, t3, dd\_q3)

legend('Faza I', 'Faza II', 'Faza III')

title('Przyspieszenie')

grid on

axis auto

end

Następnie należało wywołać poniższą funkcję dla manipulatora o dwóch przegubach obrotowych oraz narysować przebiegi pozycji kątowej, prędkości kątowej oraz przyspieszenia kątowego.

Funkcja została wywołana z poniższymi argumentami:

* Przegub 1

solved = Z;

q\_p\_in = 0;

dq\_p\_in = 20;

q\_k\_in = 100;

dq\_k\_in = 2;

Tk\_in = 12;

Ta\_in = 4;

Tb\_in = 6;

dt\_in = 1;

V\_in = 10;

out = Trajectory\_Generation(solved, q\_p\_in, dq\_p\_in, q\_k\_in, dq\_k\_in, Tk\_in, Ta\_in, Tb\_in, dt\_in, V\_in);

* Przegub 2

solved = Z;

q\_p\_in = 100;

dq\_p\_in = 2;

q\_k\_in = 200;

dq\_k\_in = 4;

Tk\_in = 12;

Ta\_in = 4;

Tb\_in = 6;

dt\_in = 1;

V\_in = 10;

out\_1 = Trajectory\_Generation(solved, q\_p\_in, dq\_p\_in, q\_k\_in, dq\_k\_in, Tk\_in, Ta\_in, Tb\_in, dt\_in, V\_in);

Obraz zawierający wykres

Opis wygenerowany automatycznie