B priedas. Viršutinės galūnės 2D trijų laisvės laipsnių modelis

```
% Virsutines galunes 2D modelis su 3LL
clc
clear all
close all
% Matlab komandos kinematikai
syms t L1 L2 L3 m1 m2 m3 g
q1=sym('q1(t)');
q2=sym('q2(t)');
q3=sym('q3(t)');
c1=cos(q1); s1=sin(q1);
c2=cos(q2); s2=sin(q2);
c3=cos(q3); s3=sin(q3);
xA=L1*s1; yA=L1*c1; rA=[xA yA 0];
rC1=0.436*rA; vC1=diff(rC1,t);
xB=xA+L2*s2; yB=yA+L2*c2; rB=[xB yB 0];
rC2=rA+0.43*rB; vC2=diff(rC2,t);
xF=xA+xB+L3*s3; yF=yA+yB+L3*c3; rF=[xF yF 0];
rC3=rA+rB+0.506*rF; vC3=diff(rC3,t);
omega1=[0 0 diff(q1,t)];
omega2=[0 0 diff(q2,t)];
omega3=[0 0 diff(q3,t)];
% Matlab komandos kinetinei E
syms IC1 IC2 IC3
IC1=0.0131; IC2=0.0068; IC3=0.001; %kgm^2
IO=IC1+m1*0.436*L1^2; %rankos segmentu inercijos momentai
IA=IC2+m2*0.43*L2^2;
IB=IC3+m3*0.506*L3^2;
T1=IO*(omega1)*omega1.'; % .' masyvo transponavimas
T2=m2*(vC2)*vC2.'/2+IA*(omega2)*omega2.'/2;
T2=simple(T2);
T3=m3*(vC3)*vC3.'/2+IB*(omega3)*omega3.'/2;
T3=simple(T3);
T=expand(T1+T2+T3);% KE trims segmentams
% dalines T isvestines pagal qi(t)'
Tdq1=deriv(T,diff(q1,t));
Tdq2=deriv(T,diff(q2,t));
Tdq3=deriv(T,diff(q3,t));
% paskutinio atsakymo diferencijavimas pagal t
Tt1=diff(Tdq1,t);
Tt2=diff(Tdq2,t);
Tt3=diff(Tdq3,t);
 Tq1=deriv(T,q1); % dalines isvestines pagal qi(t)
Tq2=deriv(T,q2);
Tq3=deriv(T,q3);
% L lygties kairioji puse
LHS1=Tt1-Tq1;
LHS2=Tt2-Tq2;
LHS3=Tt3-Tq3;
% desiniosios puses formavimas
% apibendrintosios aktyviosios jegos
G1=[0 -m1*g 0];
G2=[0 -m2*g 0];
G3=[0 -m3*g 0];
syms T01z T12z T23z
T01=[0 0 T01z];
T12=[0 0 T12z];
T23=[0 0 T23z];
% padeties vektoriaus dalines isvestines
rC1_1=deriv(rC1,q1);
```

```
rC2_1=deriv(rC2,q1);
rC3_1=deriv(rC3,q1);
rC1_2=deriv(rC1,q2);
rC2_2=deriv(rC2,q2);
rC3_2=deriv(rC3,q2);
rC1_3=deriv(rC1,q3);
rC2_3=deriv(rC2,q3);
rC3_3=deriv(rC3,q3);
% kampiniu greiciu dalines isvestines
w1_1=deriv(omega1, diff(q1,t));
w2_1=deriv(omega2, diff(q1,t));
w3_1=deriv(omega3, diff(q1,t));
w1_2=deriv(omega1, diff(q2,t));
w2_2=deriv(omega2, diff(q2,t));
w3_2=deriv(omega3, diff(q2,t));
w1_3=deriv(omega1, diff(q3,t));
w2_3=deriv(omega2, diff(q3,t));
w3_3=deriv(omega3, diff(q3,t));
% apibendrintosios aktyviosios jegos Qi
Q1=rC1_1*G1.'+w1_1*T01.'+w1_1*(-T12.')+...
    rC2_1*G2.'+w2_1*T12.'+w2_1*(-T23.')+...
    rC3_1*G3.'+w3_1*T23.';
Q2=rC1_2*G1.'+w1_2*T01.'+w1_2*(-T12.')+...
    rC2_2*G2.'+w2_2*T12.'+w2_2*(-T23.')+...
    rC3_2*G3.'+w3_2*T23.';
Q3=rC1_3*G1.'+w1_3*T01.'+w1_3*(-T12.')+...
    rC2_3*G2.'+w2_3*T12.'+w2_3*(-T23.')+...
    rC3_3*G3.'+w3_3*T23.';
Lagrange1=LHS1-Q1;
Lagrange2=LHS2-Q2;
Lagrange3=LHS3-Q3;
% tolimesni veiksmai skirti inversinei dinamikai
syms T01z T12z T23z
T01=[0 0 T01z];
T12=[0 0 T12z];
T23=[0 0 T23z];
H=1.80; M=90;
data={L1, L2, L3, m1, m2, m3, g};
datan={0.186*H, 0.146*H, 0.108*H, 0.028*M, 0.016*M, 0.006*M, 9.81};
Lagr1=subs(Lagrange1, data, datan);
Lagr2=subs(Lagrange2, data, datan);
Lagr3=subs(Lagrange3, data, datan);
% Is L lygciu apskaiciuojami sukimo momentai
  sol=solve(Lagr1, Lagr2, Lagr3, 'T01z, T12z, T23z');
T01zc=sol.T01z;
T12zc=sol.T12z;
T23zc=sol.T23z;
ql={diff(q1,t,2),diff(q2,t,2),diff(q3,t,2),...
    diff(q1,t),diff(q2,t),diff(q3,t),q1,q2,q3};
qn={ddq1n, ddq2n, ddq3n,dq1n,dq2n,dq3n q1n, q2n, q3n};
%Pries rehabilit
T01zt=subs(T01zc, ql, qn);
T12zt=subs(T12zc, ql, qn);
T23zt=subs(T23zc, ql, qn);
t=laikas;
T01zt1=subs(T01zt,t,laikas);
T12zt1=subs(T12zt,t,laikas);
T23zt1=subs(T23zt,t,laikas);
% Normalizuoju pries
T01zt2=T01zt1./(max(abs(T01zt1)));
T12zt2=T12zt1./(max(abs(T12zt1)));
```

```
T23zt2=T23zt1./(max(abs(T23zt1)));
%Po rehabilit
T01ztpo=subs(T01zc, ql, qnpo);
T12ztpo=subs(T12zc, ql, qnpo);
T23ztpo=subs(T23zc, ql, qnpo);
T01ztpo1=subs(T01ztpo,t,laikas);
T12ztpo1=subs(T12ztpo,t,laikas);
T23ztpo1=subs(T23ztpo,t,laikas);
% Normalizuoju po
T01ztpo2=T01ztpo1./(max(abs(T01ztpo1)));
T12ztpo2=T12ztpo1./(max(abs(T12ztpo1)));
T23ztpo2=T23ztpo1./(max(abs(T23ztpo1)));
```