
Table of Contents

.....	1
Atrad#sim atvasin#jumu	2
sinuso#da	2
nulles sign#ls	3
Line#ri main#gs sign#ls	3
Trokš#a sign#ls	3
nulles sign#ls	3
Vektoru apvienošana	3

```
function lab5
global E i0 a R
R = 1; i0=4e-6; Ud=0.32; id=2;
% solve (id == i0*(exp(a*Ud)-1),a)

%ans =

%log(id/i0 + 1)/Ud

a=log(id/i0 + 1)/Ud;
t = 0:0.01:5.5;
Et = 10*lab3_gabrans(t);
%Et = psigfrm(t);
Urt = [];
for E = Et
    Ur = newmet5 ;
    Urt=[Urt,Ur];
end
plot(t,Et,t,Urt)
legend('E_t(t)','U_R(t)')

function fx = funx(UR)
% Vienadojumam risinasana ar minejumu metodi
%funx.m
%i0=1; a=1; E=1; R=1;
global E i0 a R
fx = i0*(exp(a*(E-UR))-1)-UR/R;

function fx = fund(UR)
% Atrad#sim atvasin#jumu
%fund.m
%a=1; i0=1; E=1; R=1;
global E i0 a R
fx = - 1/R - a*i0*exp(a*(E - UR));

function x0 = newmet5
% Mekl#sim risin#jumu ar Nutona metodi
%newmet5.m
if nargin ==0
x0=0;
```

```

end
if nargin < 2
epsilon = 1e-3;
end
delta = funx(x0)/fund(x0);
while abs(delta) > epsilon
delta = funx(x0)/fund(x0);
x0=x0-delta;
end

function fx = funx(UR)
% Vienadojumam risinasana ar minejumu metodi
%funx.m
%i0=1; a=1; E=1; R=1;
global E i0 a R
fx = i0*(exp(a*(E-UR))-1)-UR/R;

function fx = fund(UR)

```

Atrad#sim atvasin#jumu

```

%fund.m
%a=1; i0=1; E=1; R=1;
global E i0 a R
fx = - 1/R - a*i0*exp(a*(E - UR));

function x0 = newmet5
% Mekl#sim risin#jumu ar Nutona metodi
%newmet5.m
if nargin ==0
x0=0;
end
if nargin < 2
epsilon = 1e-3;
end
%delta = funx(x0)/fund(x0);
%while abs(delta) > epsilon
%delta = funx(x0)/fund(x0);
%x0=x0-delta;

function y= lab3_gabrans(t)

if nargin == 0
t=0:0.01:5.5;
end

```

sinuso#da

```

A0 = 0;

```

```
A = 3;
T = (2.5-1)/3.5;
f = 1/T;
delay = 1;
%t_sin = 0:0.01:1.43;
t_sinf = (t>=0)&(t<1.45);
t_sin = t(t_sinf);
y_sin = A0+A*sin(2*pi*f*(t_sin-delay));
%plot(t_sin,y_sin)
```

nulles sign#ls

```
%t1_zero = 1.45:0.01:2;
t1_zerof = (t>=1.45)&(t<2);
t1_zero = t(t1_zerof);
y1_zero = 0*ones(size(t1_zero));
%plot(t1_zero,y1_zero)
```

Line#ri main#gs sign#ls

```
k = (0.5-0)/(2-3); %sl#puma koeficients
%t_saw = 2:0.01:3;
t_sawf = (t>=2)&(t<3);
t_saw = t(t_sawf);
delay = 3;
y_saw = k*(t_saw-delay);
%plot(t_saw,y_saw)
```

Trokš#a sign#ls

```
%t_noise = 3:0.01:4.5;
t_noisef = (t>=3)&(t<4.5);
t_noise = t(t_noisef);
y_noise = 3*rand(size(t_noise))-1.5;
% plot(t_noise,y_noise)
```

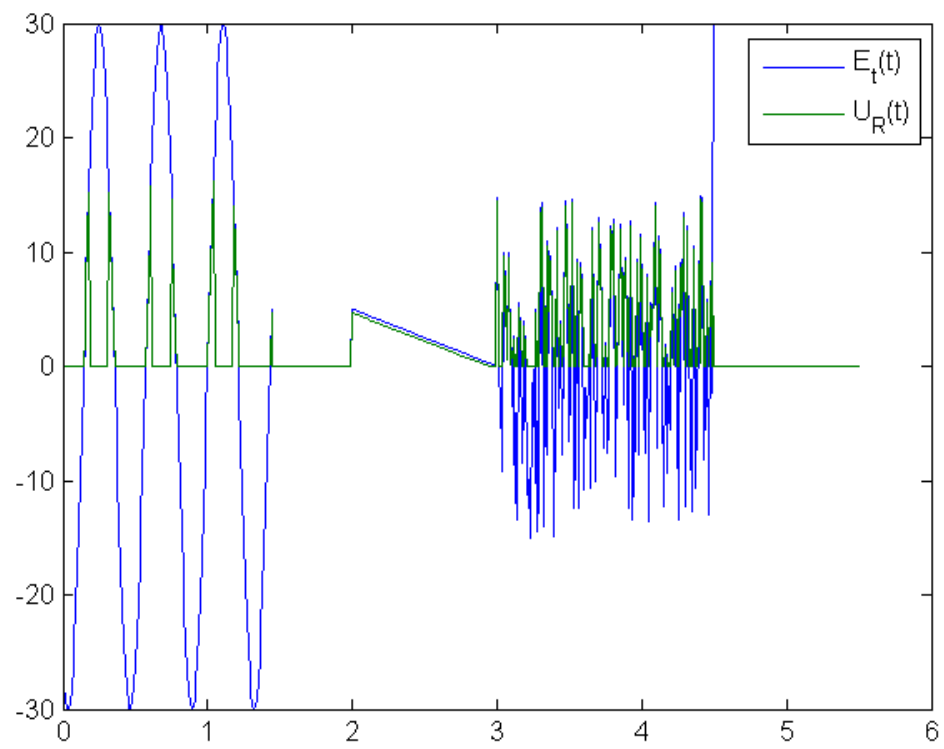
nulles sign#ls

```
%t2_zero = 4.5:0.01:5.5;
t2_zerof = (t>=4.5)&(t<=5.5);
t2_zero = t(t2_zerof);
y2_zero = 3*ones(size(t2_zero));
%plot(t2_zero,y2_zero)
```

Vektoru apvienošana

```
t = [t_sin,t1_zero,t_saw,t_noise,t2_zero];
y = [y_sin,y1_zero,y_saw,y_noise,y2_zero];
if nargin == 0
plot(t,y);
```

```
y = [];  
end
```



Published with MATLAB® R2013a