

```

%Circuito Superamortecido
close; clear;
t=linspace(0,4.3e-6,1000);%tempo de 0 a 4.3 micro segundos
t_n=[0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0]*1e-6;%tempo que foram observados os valores de tensão
%Entre com a equação da tensão de saída teórica na próxima linha depois remova "%"
%v0_t= xxx;
%Entre com a equação da tensão de saída observada na próxima linha depois remova "%"
%v0_n=[xx xx xx xx xx xx xx xx xx];
figure;
plot(t,v0_t,'linewidth', 2); grid off; hold on;
plot(xlim,[10 10], '--k', 'LineWidth',1)
plot(t_n,v0_n, '*');
ax = gca;
ax.Box = 'off';
ax.YAxisLocation = 'origin';
ax.XAxisLocation = 'origin';
ax.FontName='times';
ax.FontSize=12;
ylim([0 12])
xlim([0 4.3e-6])
xticks([t_n 4.3e-6])
xticklabels([strsplit(num2str(t_n/1e-6)), '\it{t(\mus)}'])
yticks([0 10 12])
yticklabels({'0', '10', '\it{v_o(t)},V'});
arrowsPlot(ax,1,1);

```

```

%Circuito Criticamente Amortecido
close; clear;
t=linspace(0,2.5e-6,1000);%tempo de 0 a 2.5 micro segundos
t_n=linspace(0,2.25,10)*1e-6;%tempo que foram observados os valores de tensão
%Entre com a equação da tensão de saída teórica na próxima linha depois remova "%"
%v0_t= xxx;
%Entre com a equação da tensão de saída observada na próxima linha depois remova "%"
%v0_n=[xx xx xx xx xx xx xx xx xx];
figure;
plot(t,v0_t,'linewidth', 2); grid off; hold on;
plot(t_n,v0_n, '*');
ax = gca;
ax.Box = 'off';
ax.YAxisLocation = 'origin';
ax.XAxisLocation = 'origin';
ax.FontName='times';
ax.FontSize=12;
ylim([0 12])
xlim([0 2.5e-6])
plot(xlim,[10 10], '--k', 'LineWidth',1)
xticks([t_n 2.5e-6])
xticklabels([strsplit(num2str(t_n/1e-6)), '\it{t(\mus)}'])
yticks([0 10 12])
yticklabels({'0', '10', '\it{v_o(t)},V'});

```

```
arrowsPlot(ax,1,1);
```

```
%Circuito Sub-Amortecido
close; clear;
t=linspace(0,4.3e-6,1000);;%tempo de 0 a 4.3 micro segundos
t_n=[0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0]*1e-6;%tempo que foram observados os valores de tensão
%Entre com a equação da tensão de saída teórica na próxima linha depois remova "%"
%v0_t= xxx;
%Entre com a equação da tensão de saída observada na próxima linha depois remova "%"
%v0_n=[xx xx xx xx xx xx xx xx xx];
figure;
plot(t,v0_t,'linewidth', 2); grid off; hold on;
plot(t_n,v0_n,'*');
ax = gca;
ax.Box = 'off';
ax.YAxisLocation = 'origin';
ax.XAxisLocation = 'origin';
ax.FontName='times';
ax.FontSize=12;
ylim([0 15])
xlim([0 4.3e-6])
plot(xlim,[10 10], '--k', 'LineWidth',1)
xticks([t_n 4.3e-6])
xticklabels([strsplit(num2str(t_n/1e-6)), '\it{t(\mus)}'])
yticks([0 10 15])
yticklabels({'0', '10', '\it{v_o(t)},V'});
arrowsPlot(ax,1,1);
```

FUNÇÕES SUPLEMENTARES NECESSÁRIAS PARA A EXECUÇÃO DOS SCRIPTS ACIMA

```
function twn=Ts_super(alpha,w0,epsilon)
    qsi=alpha/w0;
    delta=1e-6;
    An=1/(2*sqrt(qsi^2-1));
    Bn=1/(qsi+sqrt(qsi^2-1));
    Cn=1/(qsi-sqrt(qsi^2-1));
    tn=1.5e-6*w0;
    while 1
        fn=An*(Bn*exp(-tn/Bn)-Cn*exp(-tn/Cn))+epsilon;
        if abs(fn)>delta
            dfn=-An*(exp(-tn/Bn)-exp(-tn/Cn));
            tn=tn-fn/dfn;
        else
            break;
        end
    end
    twn=tn/w0;
end
```

```

function twn=Ts_cri(w0,epsilon)
    delta=1e-6;
    tn=1e-6*w0;
    while 1
        fn=(1+tn)*exp(-tn)-epsilon;
        if abs(fn)>delta
            dfn=-tn*exp(-tn);
            tn=tn-fn/dfn;
        else
            break;
        end
    end
    twn=tn/w0;
end

function t0=T0(alpha,w0)
    a=angle(roots([1 -2*alpha w0^2]));
    a=abs(a(1));
    t0=([1 2 3].*pi-a)/(sqrt(w0^2-alpha^2));
end

function [tp,vp]=TP(alpha,w0)
    qsi=alpha/w0;
    wd=sqrt(w0^2-alpha^2);
    tp=([1 3 5].*pi)/wd;
    vp=1-(exp(-alpha.*tp)/sqrt(1-qsi^2)).*(sin(wd.*tp+atan(sqrt(1-qsi^2)/qsi)));
end

function arrowsPlot(ax,dirX,dirY)
    xl=ax.XLim;
    yl=ax.YLim;
    y0=yl(1);
    if string(ax.XAxisLocation)=='top'
        y0=yl(2);
    elseif string(ax.XAxisLocation)=='bottom'
        y0=yl(1);
    else
        if yl(2)*yl(1)<=0
            y0=0;
        elseif yl(1)<=0
            y0=yl(2);
        end
    end
    end
    x0=xl(1);
    if string(ax.YAxisLocation)=='right'
        x0=xl(2);
    elseif string(ax.YAxisLocation)=='left'
        x0=xl(1);
    else
        if xl(2)*xl(1)<=0
            x0=0;
        elseif xl(1)<=0
            x0=xl(2);
        end
    end
end

```

```

    end
end
% Fix the axes sizes
axis([xl yl]);
if dirX<0; xf=xl(1); else; xf=xl(2); end
if dirY<0; yf=yl(1); else; yf=yl(2); end
Vx=dirX*[0 0 0.05]*(xl(2)-xl(1))+xf;
Vy=dirY*[-0.01 0.01 0]*(yl(2)-yl(1))+yf;
patch(Vx,Vy, 'k', 'clipping', 'off');
Vy=dirY*[0 0 0.05]*(yl(2)-yl(1))+yf;
Vx=dirX*[-0.01 0.01 0]*(xl(2)-xl(1))+xf;
patch(Vx,Vy, 'k', 'clipping', 'off');
if (string(ax.Title.String)~= '')
    if string(ax.XAxisLocation)=='top'
        pp=ax.Title.Position;
        pp(2)=yl(1);
        ax.Title.VerticalAlignment='top';
        ax.Title.Position=pp;
    elseif (yl(2)==0)&& (string(ax.XAxisLocation)=='origin')
        pp=ax.Title.Position;
        pp(2)=yl(1);
        ax.Title.VerticalAlignment='top';
        ax.Title.Position=pp;
    end
end
end
end

```