
Table of Contents

| | |
|--|---|
| | 1 |
| Projeto de Módulos Fotovoltaicos | 1 |
| Simulação 01 | 1 |
| Simulação 02 | 5 |
| Simulação 03 | 8 |

```
clc
clear all
```

Projeto de Módulos Fotovoltaicos

Nome: Edilberto Elias Xavier Junior Matricula: 120210134 Geração de Energia

Simulação 01

```
Resistencia = [5, 10, 20, 40, 60, 80];

% Carregar o modelo do Simulink uma vez
load_system('Simulacao1.slx');

% Loop de 5 iterações
for i = 1:6
    R = Resistencia(i);

    % Executar a simulação
    out = sim('Simulacao1.slx');

    % Obter o tempo e os dados de saída
    t = out.simout.time;
    [~, idx] = min(abs(t - 0.1));
    potencia_data = out.simout.Data(:,1); % Potencia
    corrente_data = out.simout.Data(:,2); % Corrente
    tensao_data = out.simout.Data(:,3);   % Tensão

    % Criar uma nova figura a cada iteração
    figure(i)
    subplot(2,1,1)
    plot(t, round(potencia_data,2), LineStyle="-", LineWidth=1.5)
    ylabel(' [kW] ')
    ylim([min(potencia_data(idx:end))*0.8,max(potencia_data(idx:end))*1.1])
    title(['Simulação Carga = ' num2str(R) ' ' char(937)],FontSize=20)
    xlabel(' [s] ')
    xlim([0.1,1])
    grid on
    ax = gca;
    ax.XAxis.FontSize = 13;
    ax.YAxis.FontSize = 13;
```

```

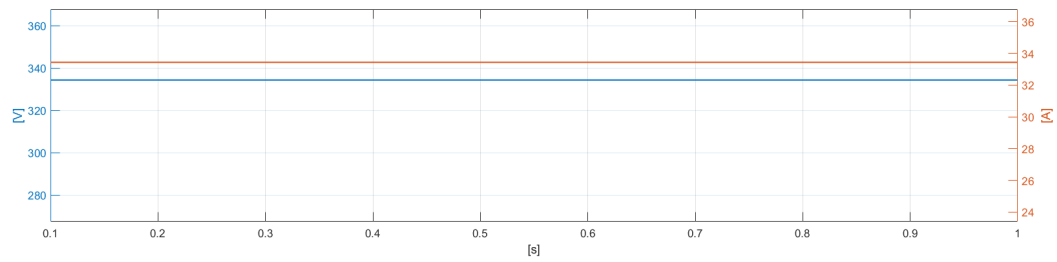
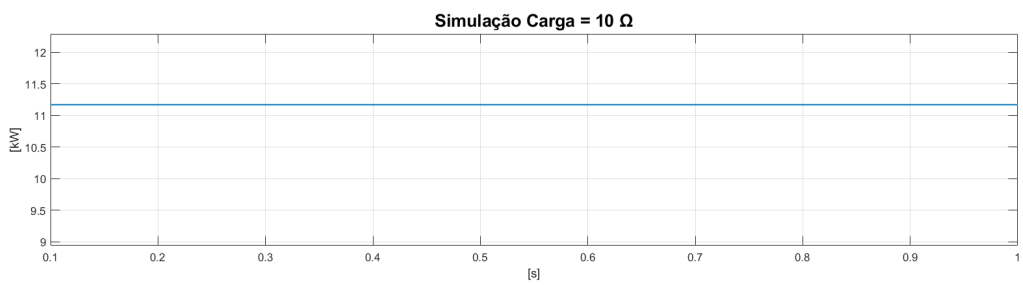
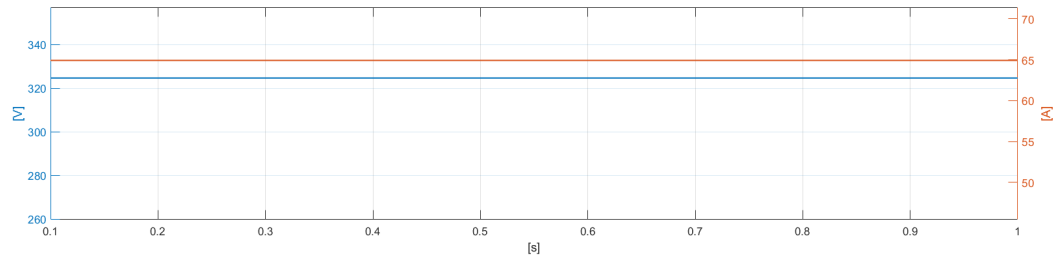
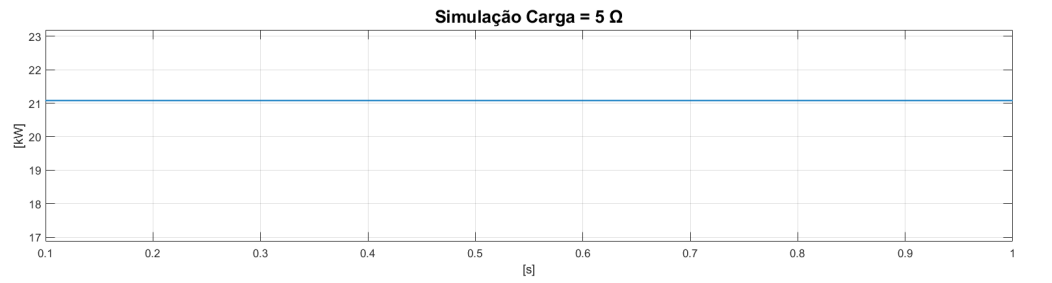
subplot(2,1,2)
yyaxis left
plot(t, round(tensao_data,2), LineStyle="-", LineWidth=1.5)
ylabel(' [V] ')
ylim([min(tensao_data(idx:end))*0.8,max(tensao_data(idx:end))*1.1])
yyaxis right
plot(t, round(corrente_data,2), LineStyle="-", LineWidth=1.5)
ylabel(' [A] ')
ylim([min(corrente_data(idx:end))*0.7,max(corrente_data(idx:end))*1.1])
xlabel(' [s] ')
xlim([0.1,1])
grid on
ax = gca;
ax.XAxis.FontSize = 13;
ax.YAxis(1).FontSize = 13;
ax.YAxis(2).FontSize = 13;
saveas(gcf, ['Simulacao_' num2str(R) '.png']);

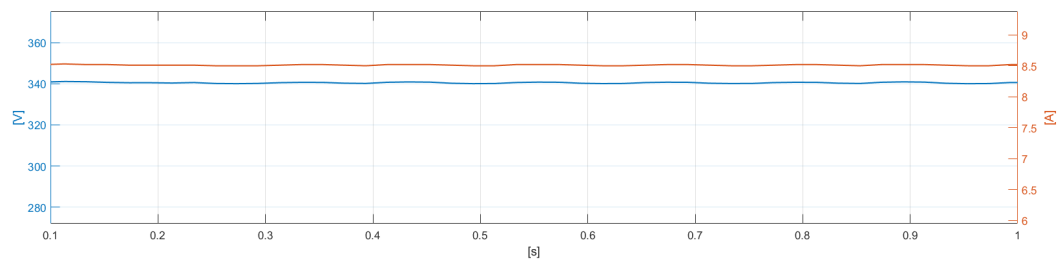
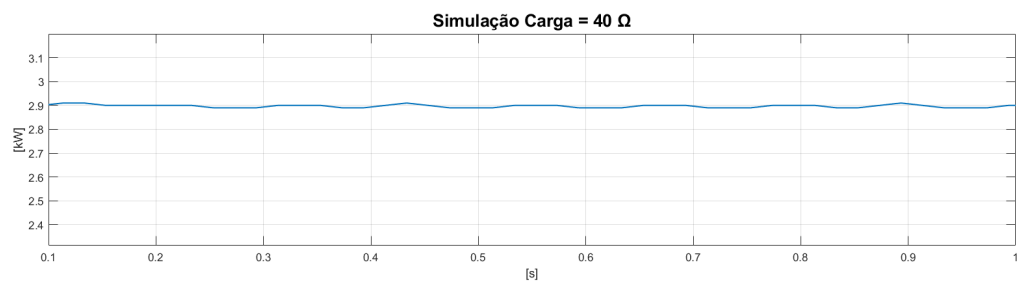
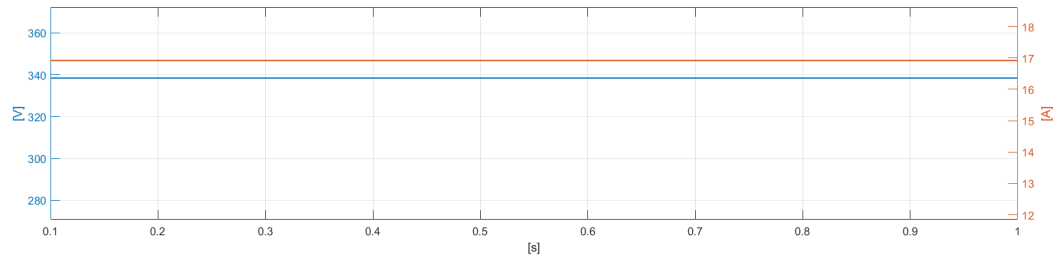
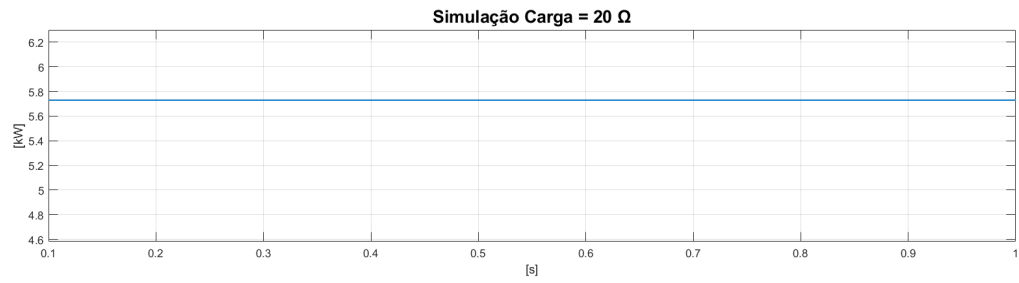
disp(['Simulação Carga = ' num2str(R) ' ' char(937)]);
disp(['Potencia = ' num2str(round(potencia_data(end),2)) ' kW']);
disp(['Tensão = ' num2str(round(tensao_data(end),2)) ' V']);
disp(['Corrente = ' num2str(round(corrente_data(end),2)) ' A']);
end

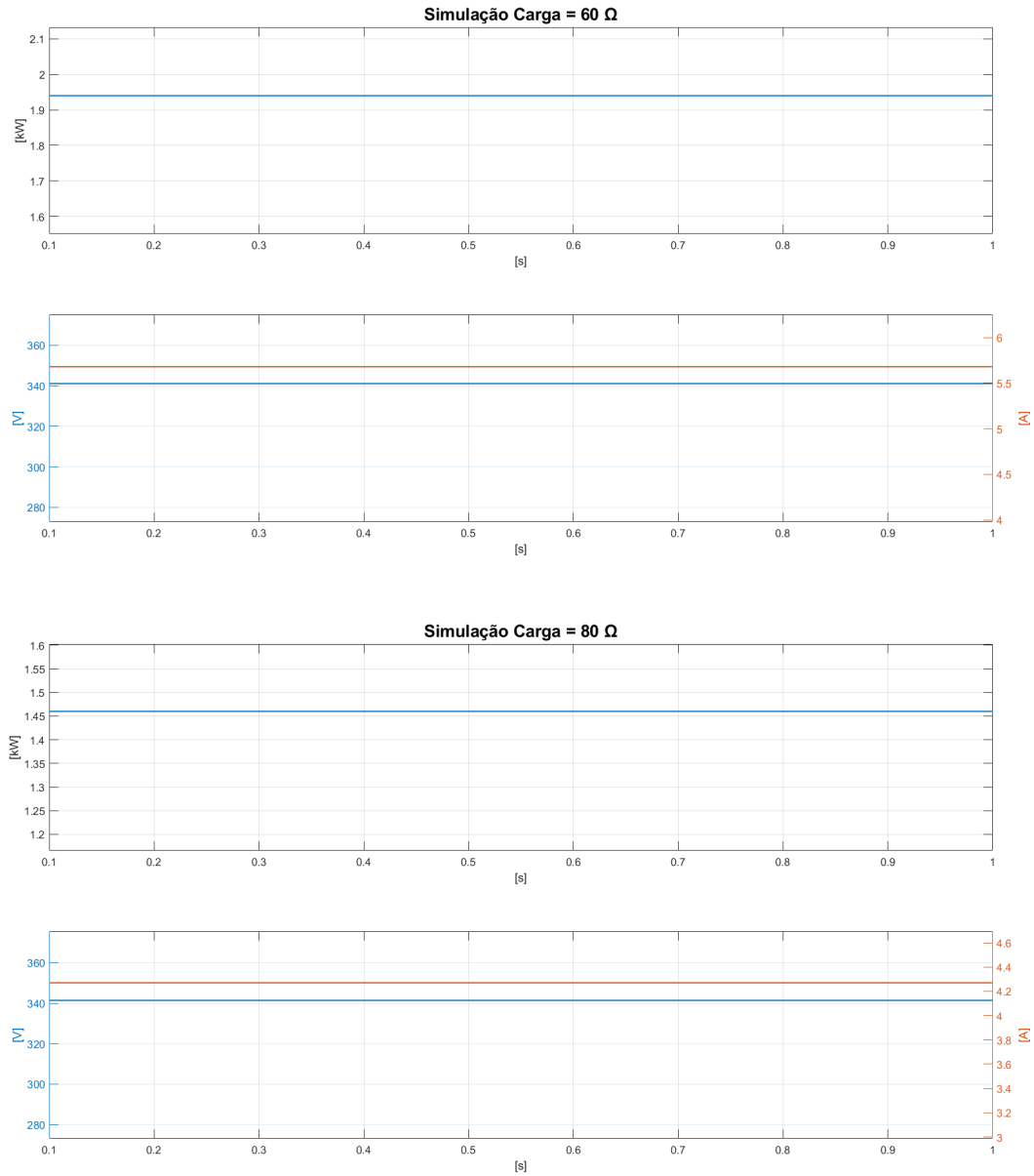
% Fechar o modelo após a simulação
save_system('Simulacao1.slx');
close_system('Simulacao1.slx');

Simulação Carga = 5  $\Omega$ 
Potencia = 21.08 kW
Tensão = 324.67 V
Corrente = 64.93 A
Simulação Carga = 10  $\Omega$ 
Potencia = 11.17 kW
Tensão = 334.27 V
Corrente = 33.43 A
Simulação Carga = 20  $\Omega$ 
Potencia = 5.73 kW
Tensão = 338.45 V
Corrente = 16.92 A
Simulação Carga = 40  $\Omega$ 
Potencia = 2.9 kW
Tensão = 340.62 V
Corrente = 8.52 A
Simulação Carga = 60  $\Omega$ 
Potencia = 1.94 kW
Tensão = 341.08 V
Corrente = 5.68 A
Simulação Carga = 80  $\Omega$ 
Potencia = 1.46 kW
Tensão = 341.4 V
Corrente = 4.27 A

```







Simulação 02

Carregar o modelo do Simulink uma vez

```
load_system('Simulacao2.slx');
```

```
% Executar a simulação
```

```
out = sim('Simulacao2.slx');
```

```
% Obter o tempo e os dados de saída sem filtro
```

```
t = out.simout.time;
```

```
y = out.simout.data;
```

```

figure(7)
plot(t, y, LineStyle="-", LineWidth=1.5)
ylabel('[V]')
title(['Simulação Saída da Célula Fotovoltaica'],FontSize=20)
xlabel('[s]')
xlim([0,1])
grid on
ax = gca;
ax.XAxis.FontSize = 13;
ax.YAxis.FontSize = 13;
saveas(gcf, ['Simulacao_Saida_da_Celula_Fotovoltaica.png']);

% Obter o tempo e os dados de saída sem filtro
t = out.simout1.time;
y = out.simout1.data;

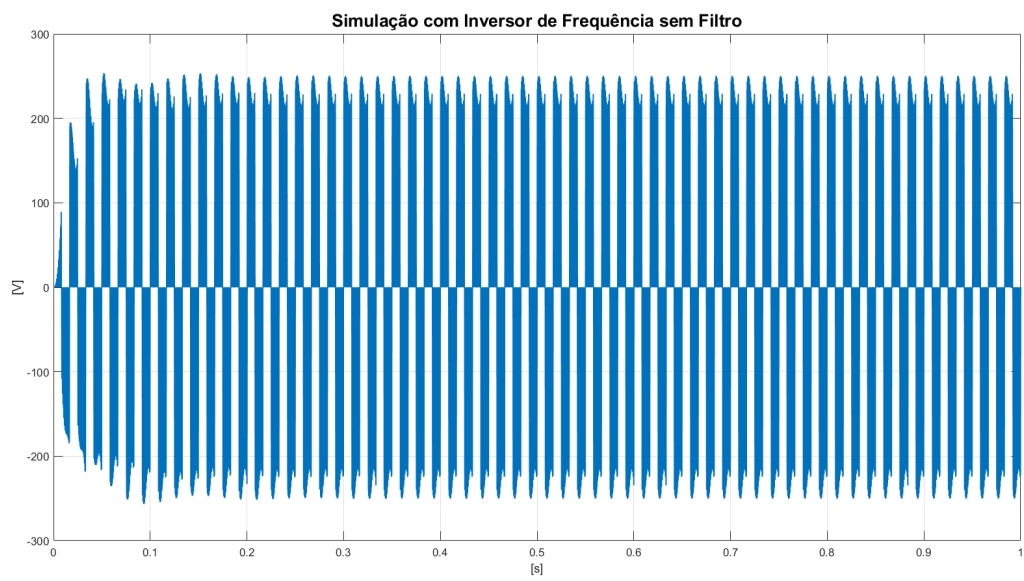
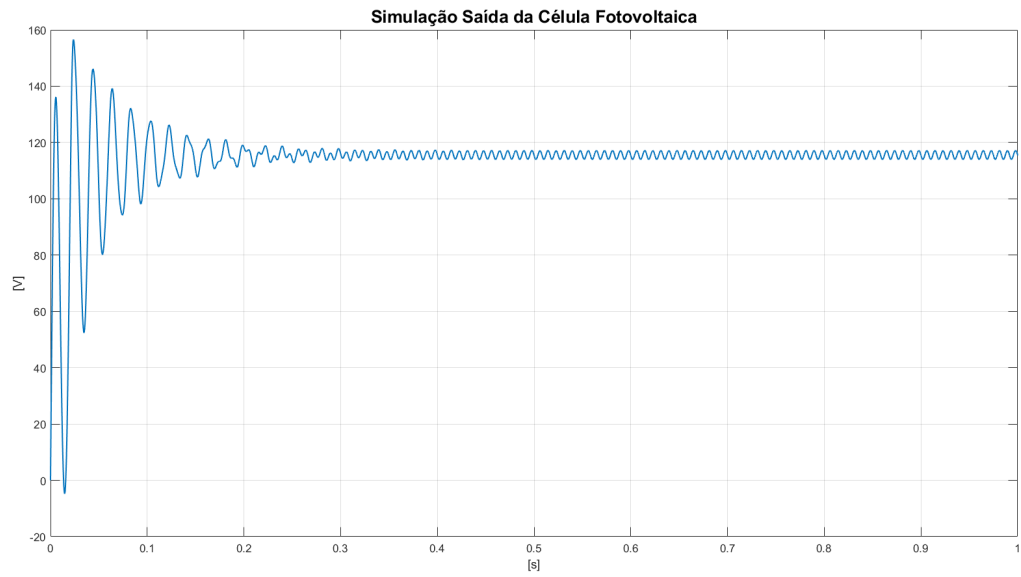
figure(8)
plot(t, y, LineStyle="-", LineWidth=1.5)
ylabel('[V]')
title(['Simulação com Inversor de Frequência sem Filtro'],FontSize=20)
xlabel('[s]')
xlim([0,1])
grid on
ax = gca;
ax.XAxis.FontSize = 13;
ax.YAxis.FontSize = 13;
saveas(gcf, ['Simulacao_com_Inversor_de_Frequência_sem_Filtro.png']);

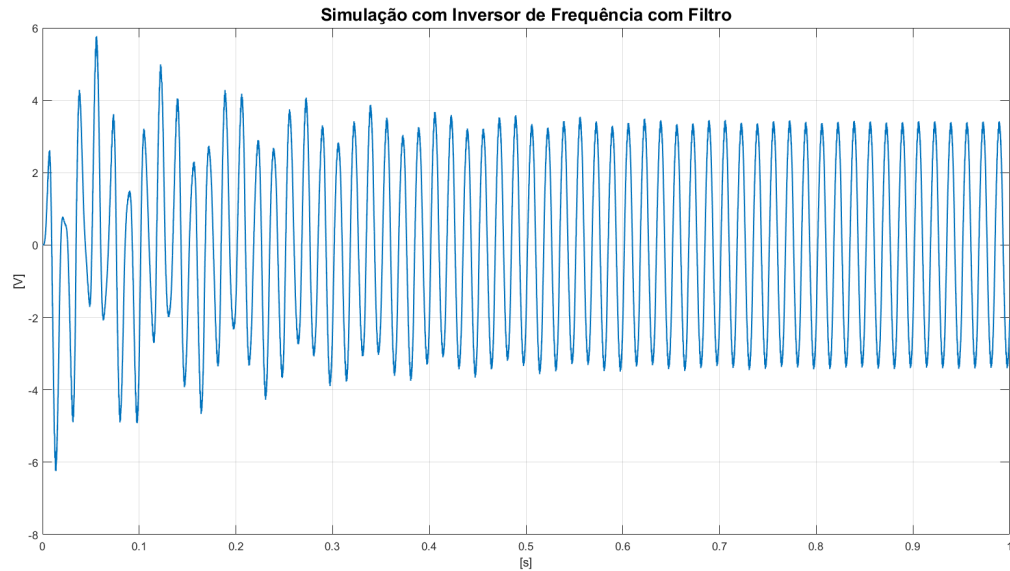
% Obter o tempo e os dados de saída sem filtro
t = out.simout2.time;
y = out.simout2.data;

figure(9)
plot(t, y, LineStyle="-", LineWidth=1.5)
ylabel('[V]')
title(['Simulação com Inversor de Frequência com Filtro'],FontSize=20)
xlabel('[s]')
xlim([0,1])
grid on
ax = gca;
ax.XAxis.FontSize = 13;
ax.YAxis.FontSize = 13;
saveas(gcf, ['Simulacao_com_Inversor_de_Frequência_com_Filtro.png']);

% Fechar o modelo após a simulação
save_system('Simulacao2.slx');
close_system('Simulacao2.slx');

```





Simulação 03

Carregar o modelo do Simulink uma vez

```
load_system('Simulacao3.slx');

% Executar a simulação
out = sim('Simulacao3.slx');

% Obter o tempo e os dados de saída sem filtro
t = out.simout.time;
tensao = out.simout.data(:,3);
corrente = out.simout.data(:,2);
potencia = out.simout.data(:,1);

figure(10)
subplot(2,1,1)
plot(t, round(potencia,2), LineStyle="--", LineWidth=1.5)
ylabel('[kW]')
xlabel('[s]')
grid on
ax = gca;
ax.XAxis.FontSize = 13;
ax.YAxis.FontSize = 13;
title(['Simulação controle do MPPT'],FontSize=20)
subplot(2,1,2)
yyaxis left
plot(t, round(tensao,2), LineStyle="--", LineWidth=1.5)
ylabel('[V]')
yyaxis right
plot(t, round(corrente,2), LineStyle="--", LineWidth=1.5)
ylabel('[A]')
xlabel('[s]')
```

```

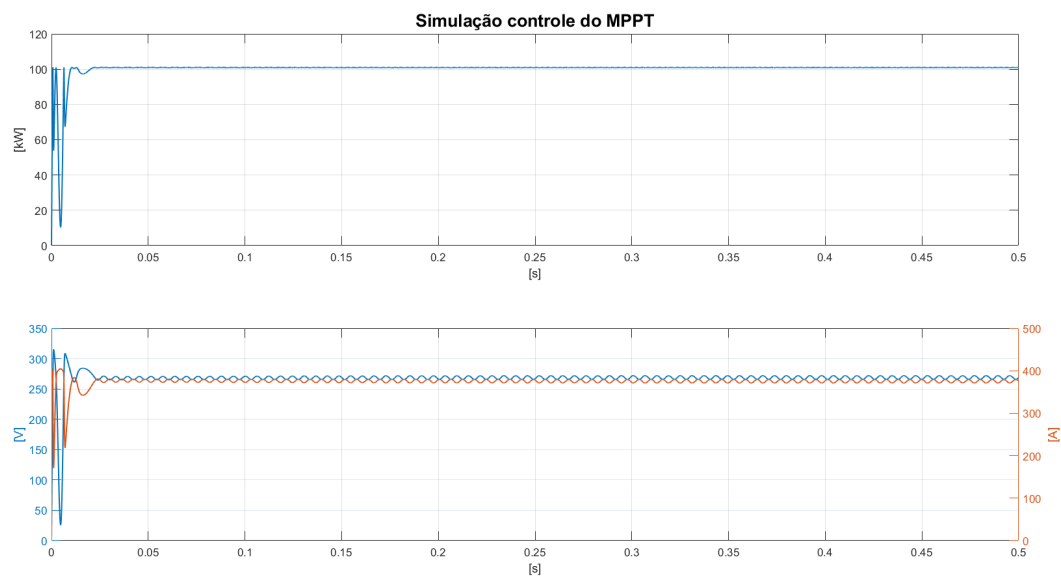
grid on
ax = gca;
ax.XAxis.FontSize = 13;
ax.YAxis(1).FontSize = 13;
ax.YAxis(2).FontSize = 13;
saveas(gcf, ['Simulacao_Controlo_do_MPPT.png']);

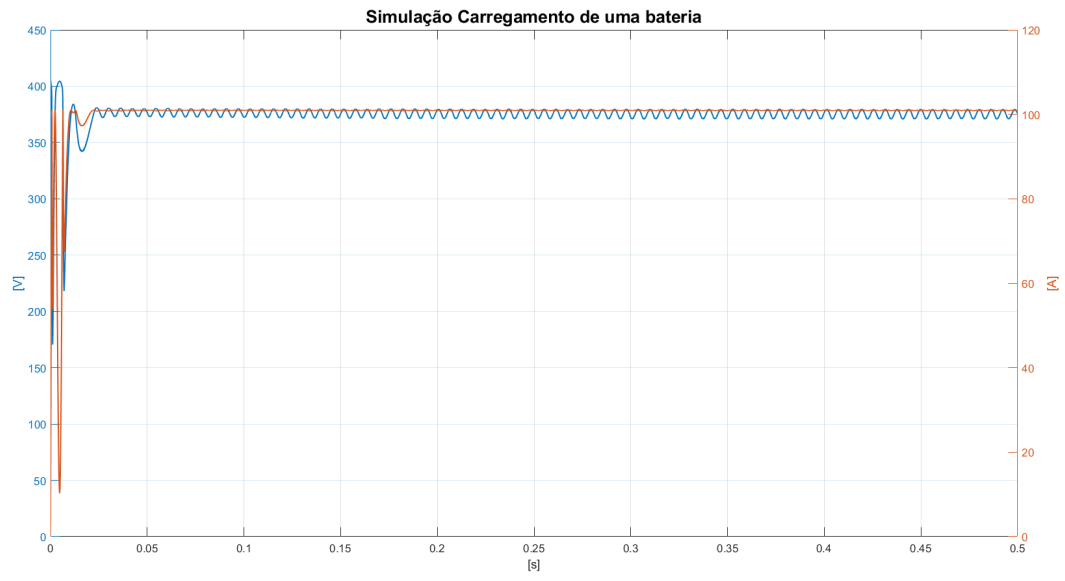
% Obter o tempo e os dados de saída sem filtro
t = out.simout1.time;
tensao_bt = out.simout.data(:,2);
corrente_bt = out.simout.data(:,1);

figure(11)
yyaxis left
plot(t, round(tensao_bt,2), LineStyle="-", LineWidth=1.5)
ylabel('[V]')
yyaxis right
plot(t, round(corrente_bt,2), LineStyle="-", LineWidth=1.5)
ylabel('[A]')
title(['Simulação Carregamento de uma bateria'],FontSize=20)
xlabel('[s]')
grid on
ax = gca;
ax.XAxis.FontSize = 13;
ax.YAxis(1).FontSize = 13;
ax.YAxis(2).FontSize = 13;
saveas(gcf, ['Simulacao_Carregamento_de_uma_Bateria.png']);

% Fechar o modelo após a simulação
save_system('Simulacao3.slx');
close_system('Simulacao3.slx');

```





Published with MATLAB® R2024a