FH Joanneum

Graz

Model Based Design

---

02 Solar Cell

Training Unit 2 Solar Cell

Autor
David B. Heer

Lecturer
Alfred Steinhuber

Graz, January 8, 2019

Contents

Preparation

In the subject Model Based Design, a solar cell was modulated. The necessary equations are shown in Equation 1 - 4. The associated scheme is shown in Figure 1

$$
\begin{aligned}
I &= P_{ph} - I_D - I_{sh} && (1)\\
I_{ph} &= J_{sc} * A * \frac{G}{G_n} && (2)\\
I_D &= I_0 * \left(e^{\frac{V+I*R_S}{m*V_T}} - 1 \right) && (3)\\
I_{sh} &= \frac{V + I * R_S}{R_{sh}} && (4)
\end{aligned}
$$

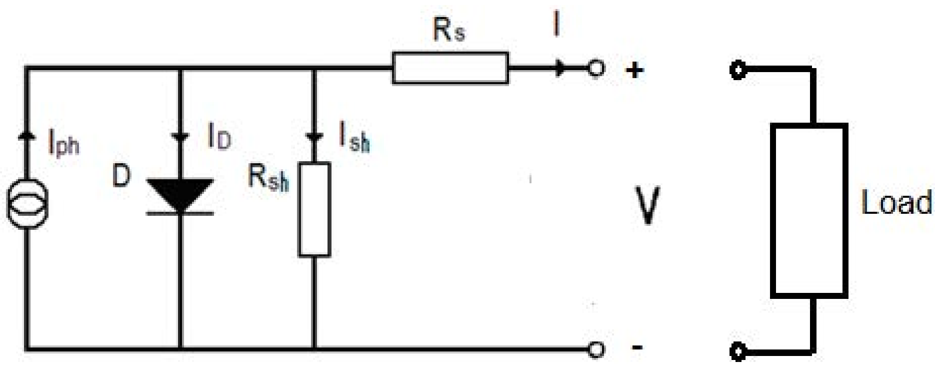

Figure 1: *Scheme of solar cell with current source, perfect diode, internal and load resistors.*

1 Model Development

1.1 Parameters

A .mat file has the advantage that it saves just variables. If you want to call variables in a .m file the whole skrip has to be calculated. This can be noticeable with many variables. See code in Listing 1 Line 5-19.

1.2 Interface

The irradiance can 't be influenced and is clearly an input. The solar cell delivers a power. A part of this power is losses in the cell itself. Depending on the irradiance [G] and the load, a different output voltage [V] and current [I] is established. Charging a battery requires a constant voltage [V]. Therefore, this is handled as an input and the current [I] as output.

⇐ [G] Irradiance

⇐ [V] Voltage

⇒ [I] Current

1.3 Model Design

With the equations from Equation 1 - 4 was built a model of the system with subsystems. In Figure 2 is an overview of the system. This topsystem represents Equation 1. Each of the subsytems symbolizes one of the Equations 2 - 4. Details of the subsystems are visible in Figure 3 - 5.

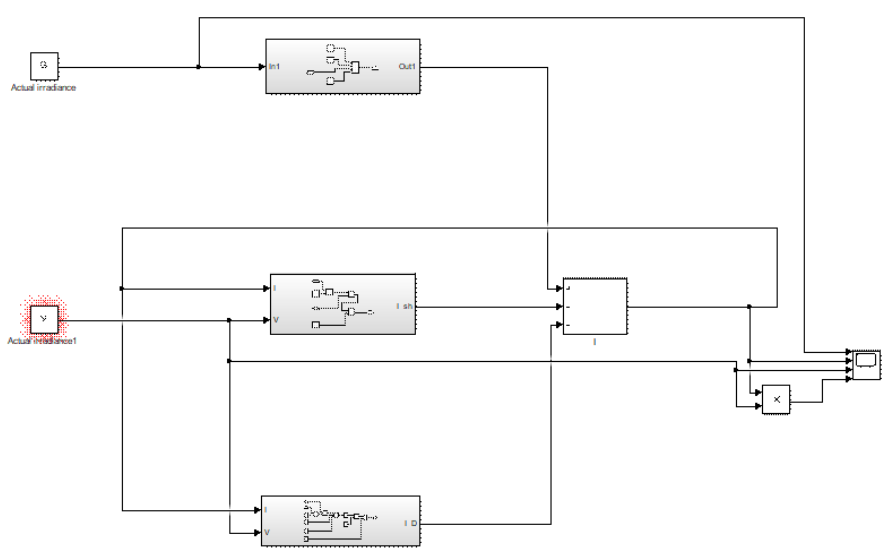

Figure 2: *Mathematical overview scheme based on the Equations 1 - 4*

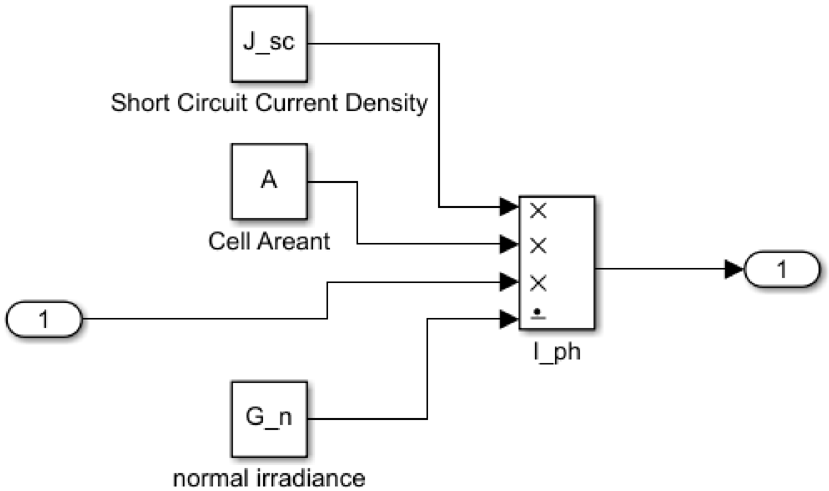

Figure 3: *The total current of the irradiance without losses. The top block on Figure 2.*

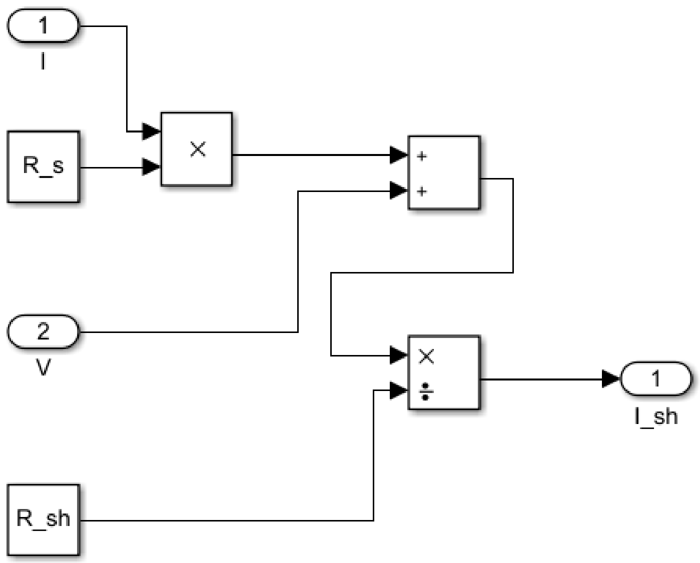

Figure 4: *Loss current from the internal resistance. The block in the middle on Figure 2.*

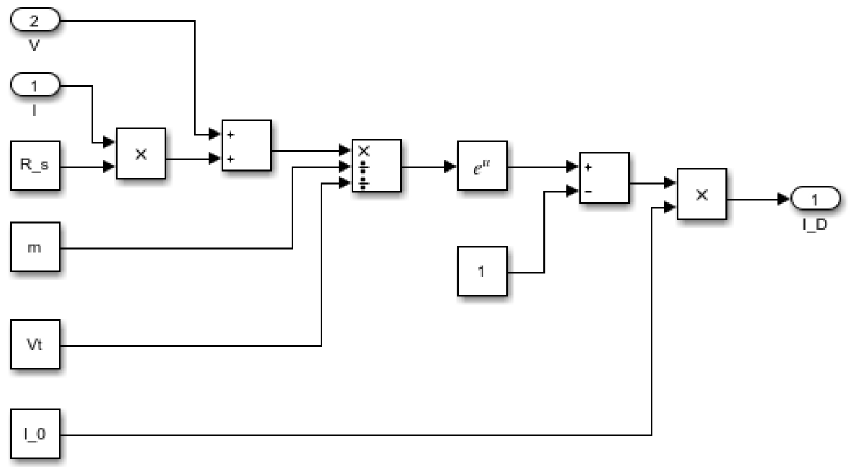

Figure 5: *Loss current of the diode. The bottom block on Figure 2*

1.4 Simulation

There where no warnings. Everything is fine.

2 Model Testing

For the following simulations the irradiance was fixed at $1000^{W}/_{m^2}$. On all scopes the x-axes is the time in seconds. The y-axes are declarated in each window.

2.1 Short Circuit

In the event of a short circuit, the output voltage is set to 0V. In Figure 6 are the graphes with the values visible. The total power is increased over the resistances R_s and R_{sh}. The current is visible on Figure 6 and is approximately 4.35A. A higher irradiance delievers a higher short circuit current. More about the influence of irradiance in Section 3.

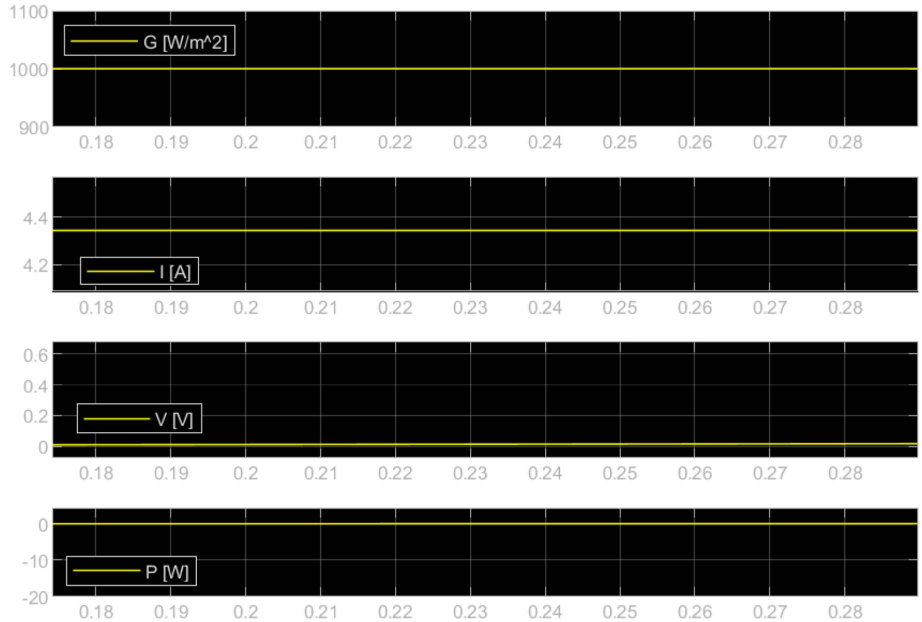

Figure 6: *The scope arround the point of a short circuit with irradiance of* $1000^W/_{m^2}$*. On all scopes the x-axes is the time in seconds. The y-axes is declarated in each window.*

2.2 Open Circuit

When no load is connected, the whole power is increased over R_{sh}. The current I is 0A. The values can be read in Figure 7 at the place where the current crosses the 0A line. The voltage at this point is about 0.455V.

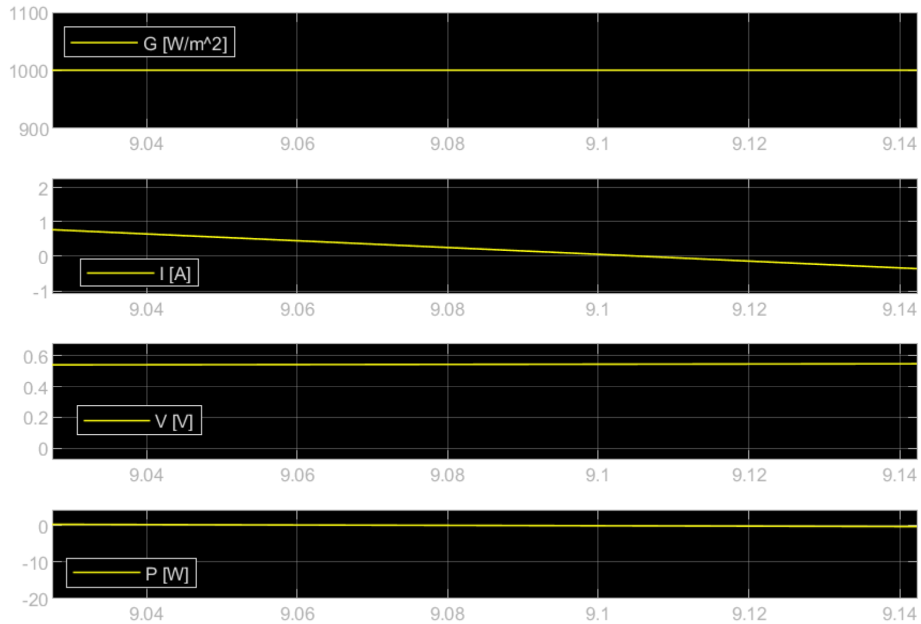

Figure 7: *The scpoe arraound the point of an open circuit with irradiance of* $1000^{W}/_{m^2}$.

2.3 Open Circuit with a higher Voltage

If the voltage raises further the current get negative and the power too. That means the solar cell can 't deliver more then this approximately 0.455V by an irradiance of $1000^{W}/_{m^2}$. This is visible in Figure 18.

2.4 Open Circuit with a higher Irradiance

If the irradiance is increased to $1500^{W}/_{m^2}$, a higher output voltage will occur. This is visible on Figure 8 where the voltage is about 0.55V. More about this topis in Section 3.

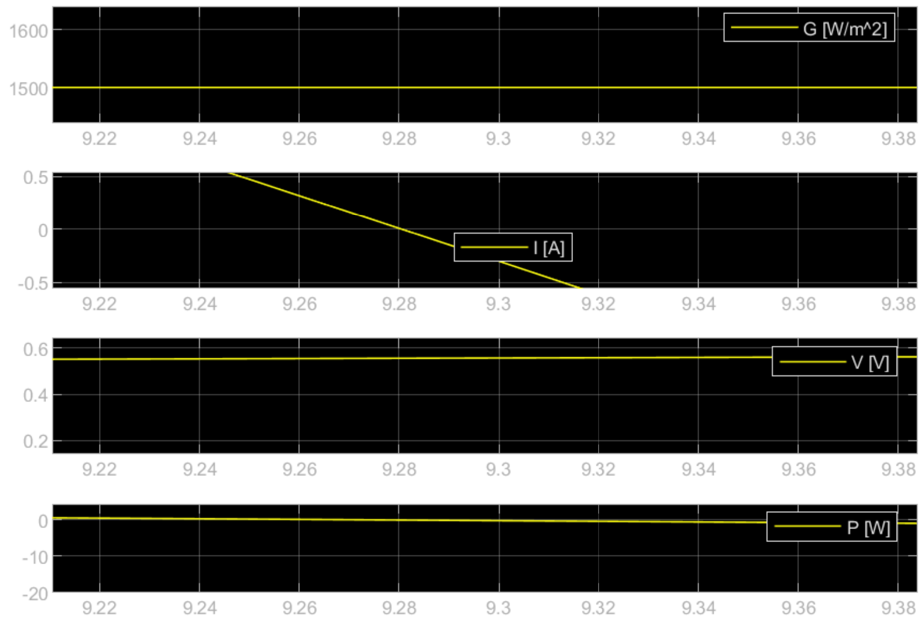

Figure 8: *The scpoe arraound the point of an open circuit with irradiance of* $1500^{W}/_{m^2}$.

2.5 Storing of simulation results

The loged datas will be stored in a Struct. Each port has the properties:

- PortType
- PortIndex
- PropagatedName
- Blockpath
- Values
- Name

Each value is logged every 0.2 seconds. The Figure 9 is made with the loged datas after the resistance ramp was removed. The resistance and the power graphs are calculated with voltage and current. See Listing 1 Line 25-56.

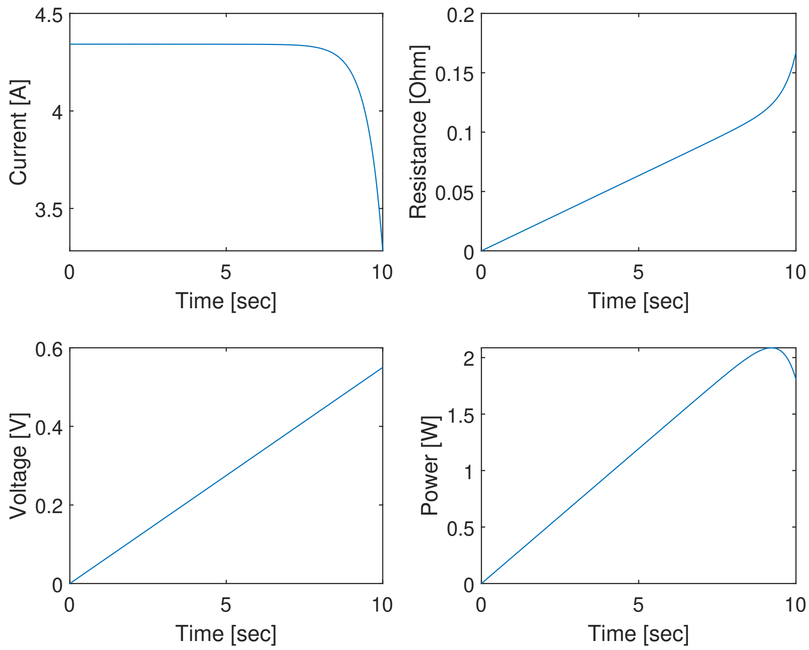

Figure 9: *The ploted graph is made with the loged datas.*

2.6 Characteristic curves of the solar cell

The simulink model (Figure 10) was extended with a variable R_{Load} as a ramp with a slope of $0.1^{\Omega}/_{s}$. The voltage V was defined as product of R_{Load} and I. The $R_{Load}$was added to the scoap in Figure 11.
It is clear visible that the current becomes smaller after about one second with greater resistance and the voltage first rises quickly and goes to saturation at about 5.5V from about 1.25 seconds. The power has its peak of almost 2W at about 1.25 seconds.

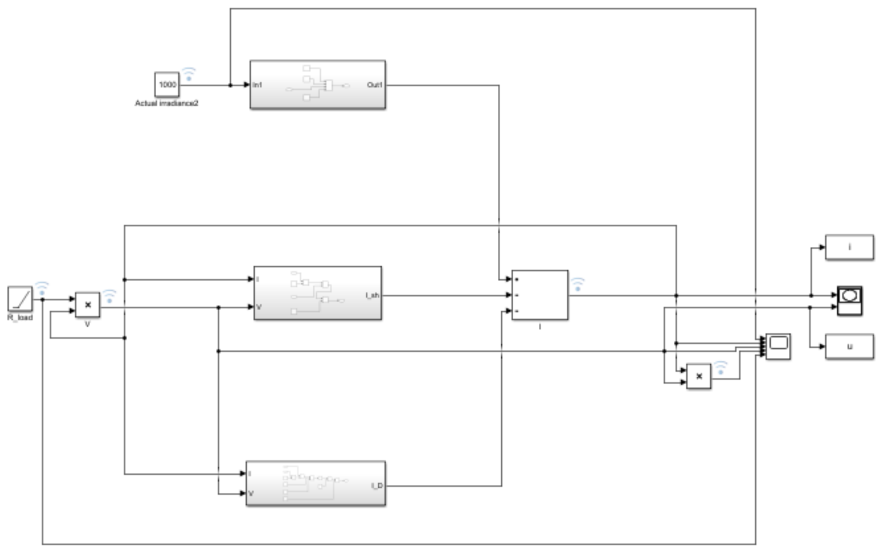

Figure 10: *The voltage defined as product of I and R_{Load}. The variables V and I where recorded as array in Matlab. Visible on the right side of the figure, next to an XY-Scope.*

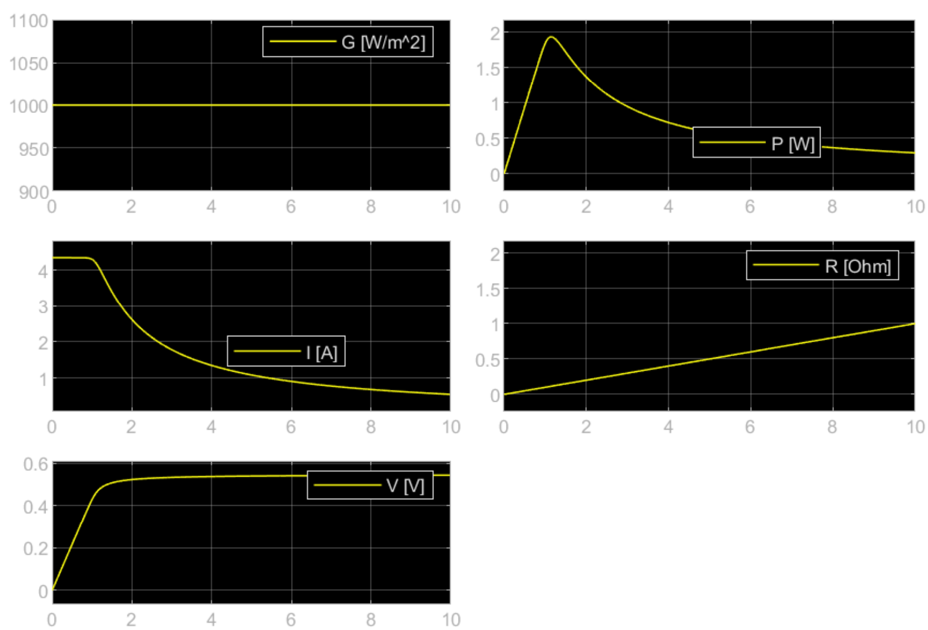

Figure 11: *The extended scope with R_{Load}.*

In Figure 12 is the dependence between current and voltage. The goal is to find the point with the largest area with the product of voltage and current. This can be seen more easily in Figure 11 as peak of the power in the graph top right.

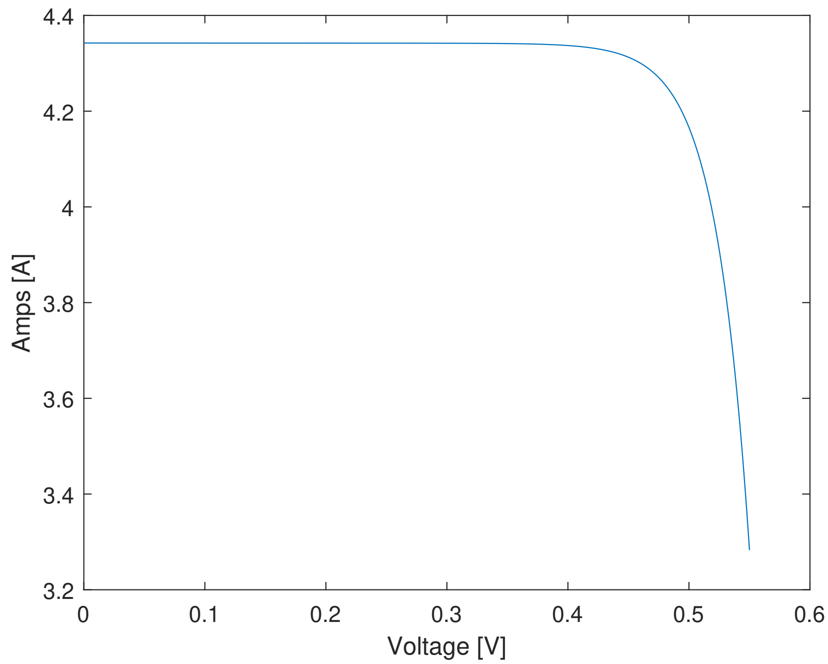

Figure 12: *XY-Plot with voltage and current. The product of current and voltage from each point on the line is the resulting power in watt.*

3 Irradiance effect on solar cell performance

3.1 Temperature

The thermal voltage was already set at the start of the laboratory unit to 25.85V at 300K. The difference of a ΔT of 1.85K is unknown. More on this topic in Section 5.

3.2 Variable Irradiance

On the following Figures 13 - 17 the irradiance was varied. All other settings have been retained. The scaling of the plots is also the same in all figures. On Figure 18 is an current voltage plot overview with all irradiances. On Figure 21 is an overview with power and voltage with different irradiances. The Matlab code for Figure 18 and Figure 21 is in Listing 1 Line 66 - 126.

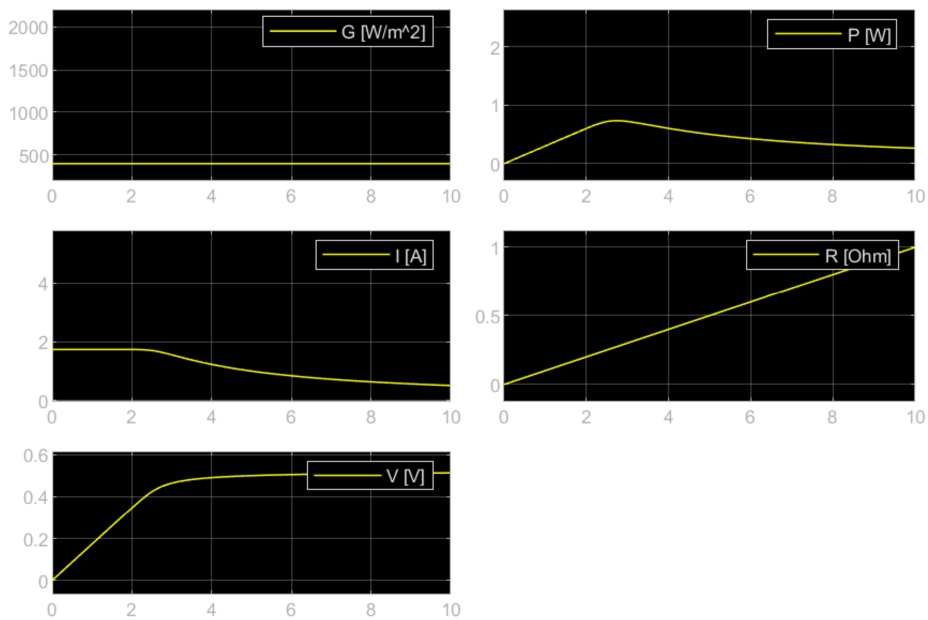

Figure 13: *Fixed irradiance at 400* $^{W}/_{m^2}$ *with a variable resistance as ramp from 0-1Ω*

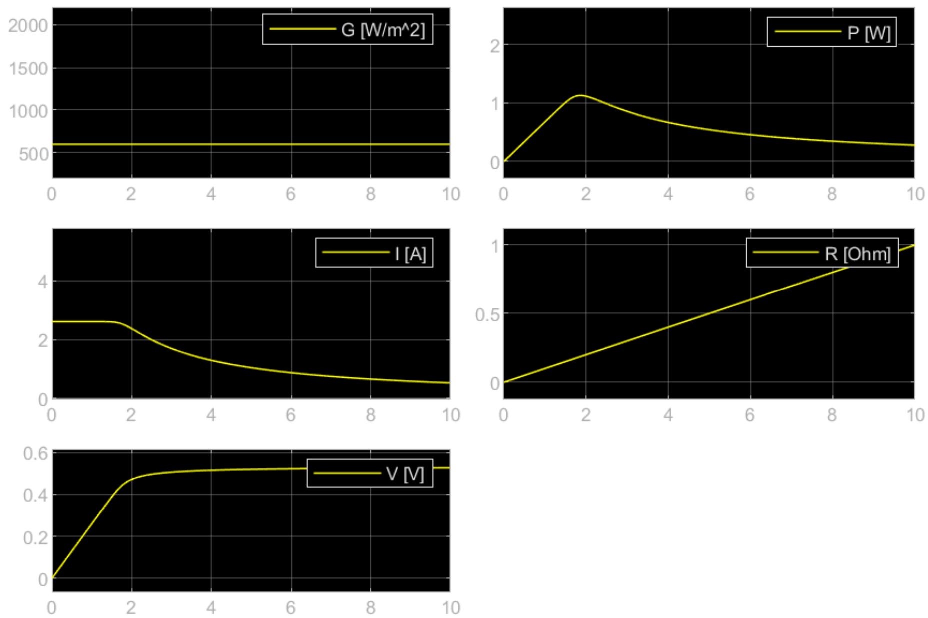

Figure 14: *Fixed irradiance at 600* $^{W}/_{m^2}$ *with a variable resistance as ramp from 0-1*Ω

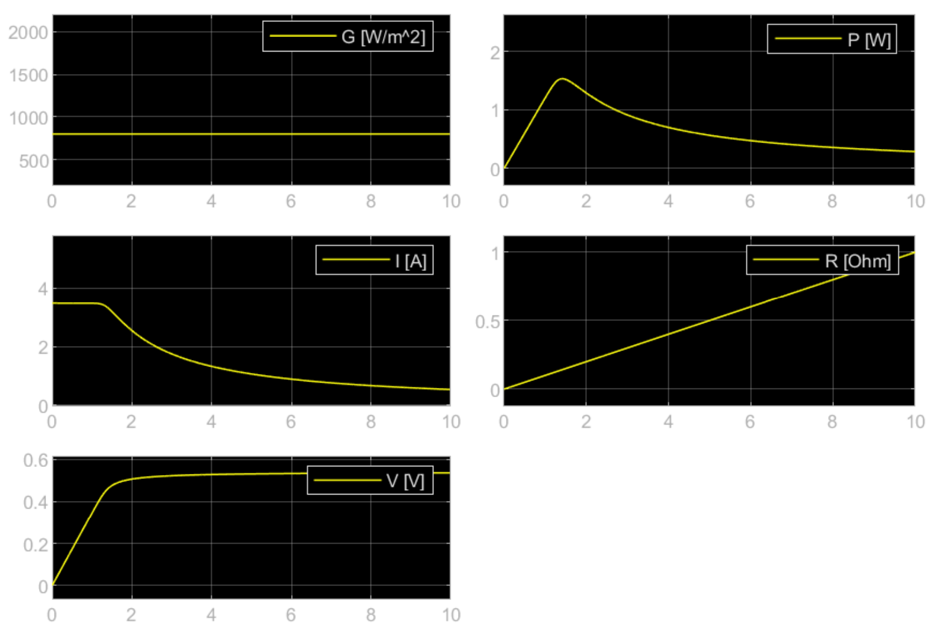

Figure 15: *Fixed irradiance at 800* $^{W}/_{m^2}$ *with a variable resistance as ramp from 0-1*Ω

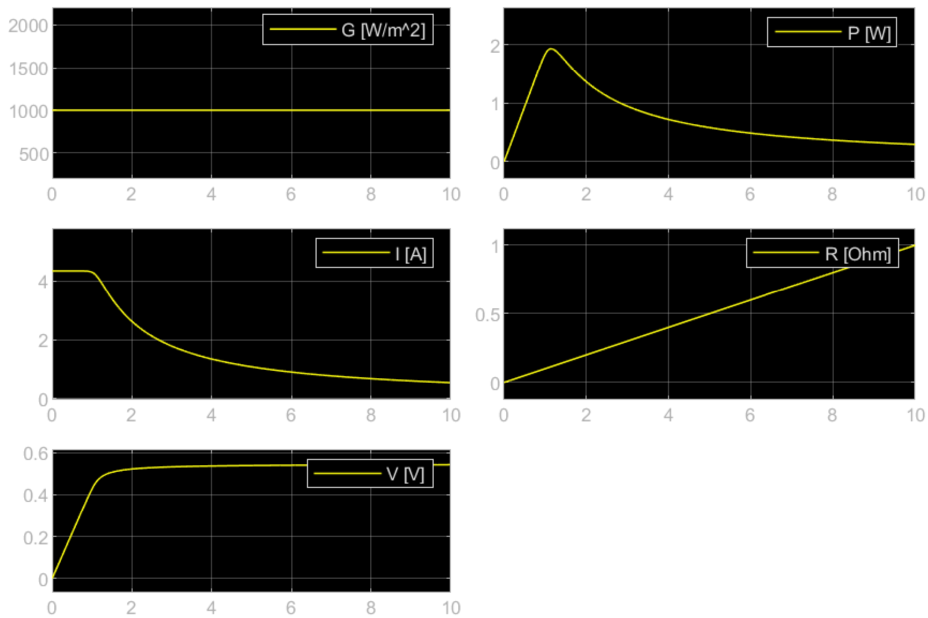

Figure 16: *Fixed irradiance at 1000* $^{W}/_{m^2}$ *with a variable resistance as ramp from 0-1*Ω

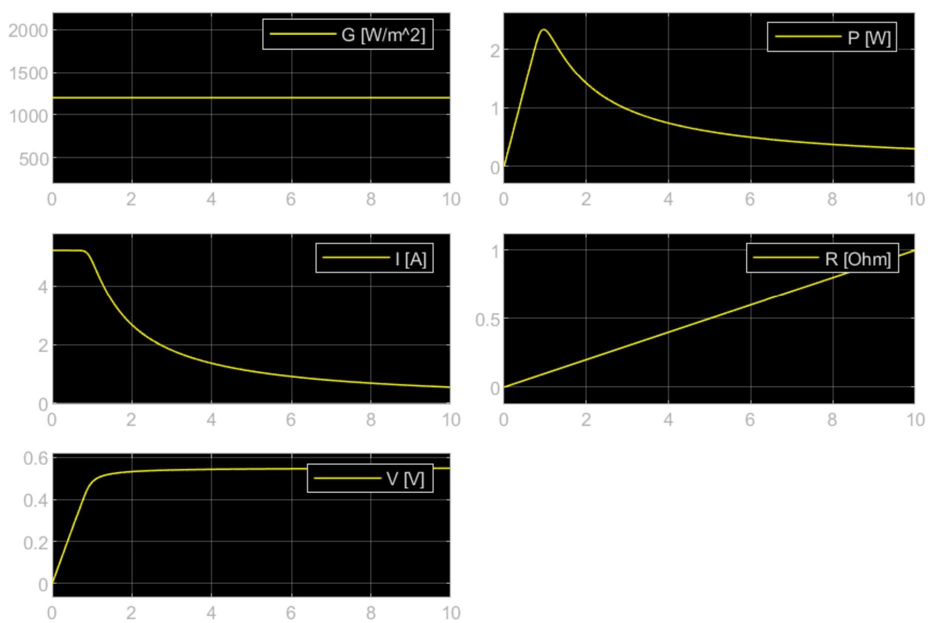

Figure 17: *Fixed irradiance at 1200* $^{W}/_{m^2}$ *with a variable resistance as ramp from 0-1*Ω

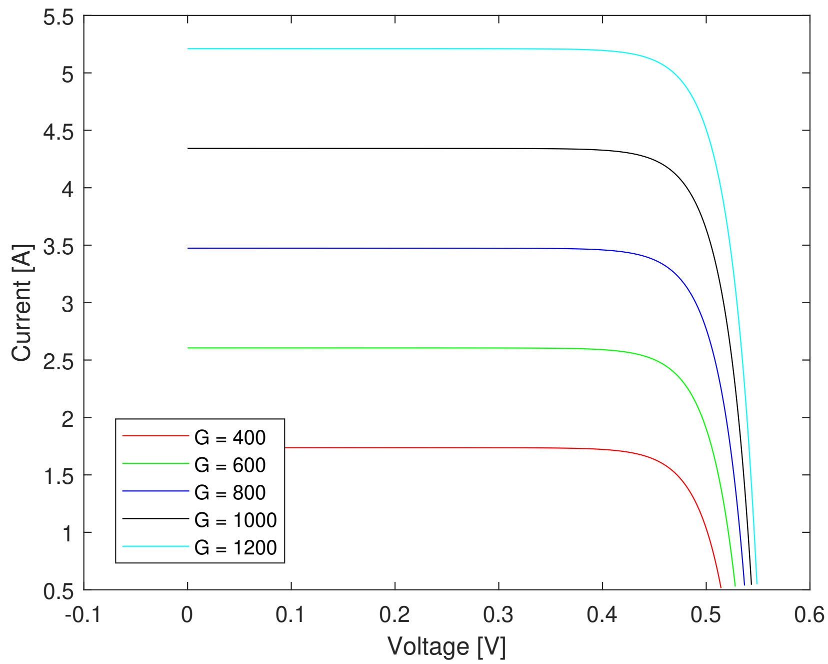

Figure 18: *Current dipence on voltage. The different lines symbolizes different irradiances G.*

3.3 Maximum Power Point

On the following Figures 19 and 20 are an XY-plot of current and voltage next to a XY-plot of voltage and power. The scale is different on each plot to make the Maximum Power Point better visible. On the last Figure 21 is an overview of all Maximum Power Points from the different irradiances. The Matlab code is in Listing 1 Line131 - 145.

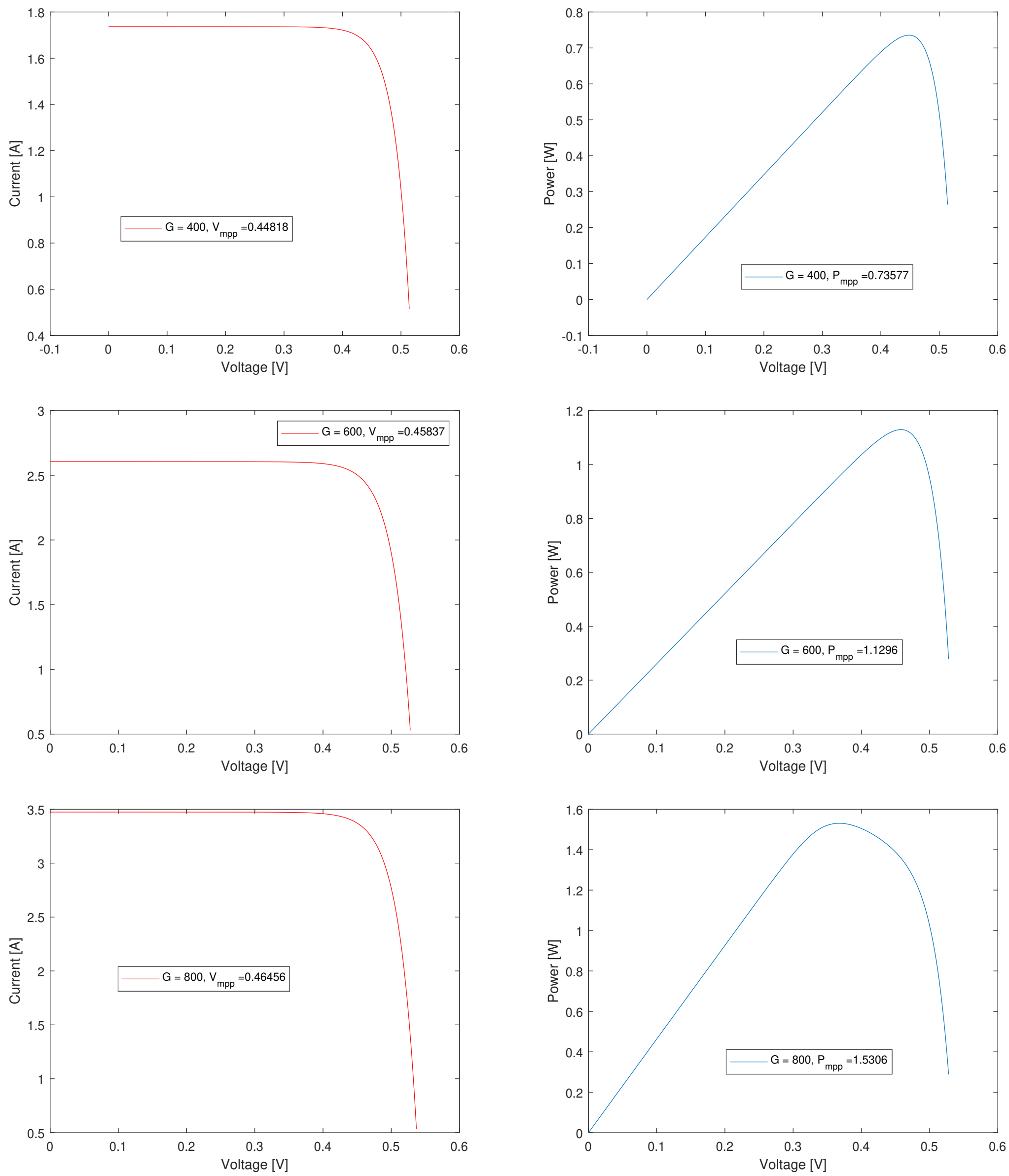

Figure 19: *Maximum Power Point, G = 400, 600, 800*
1 of 2

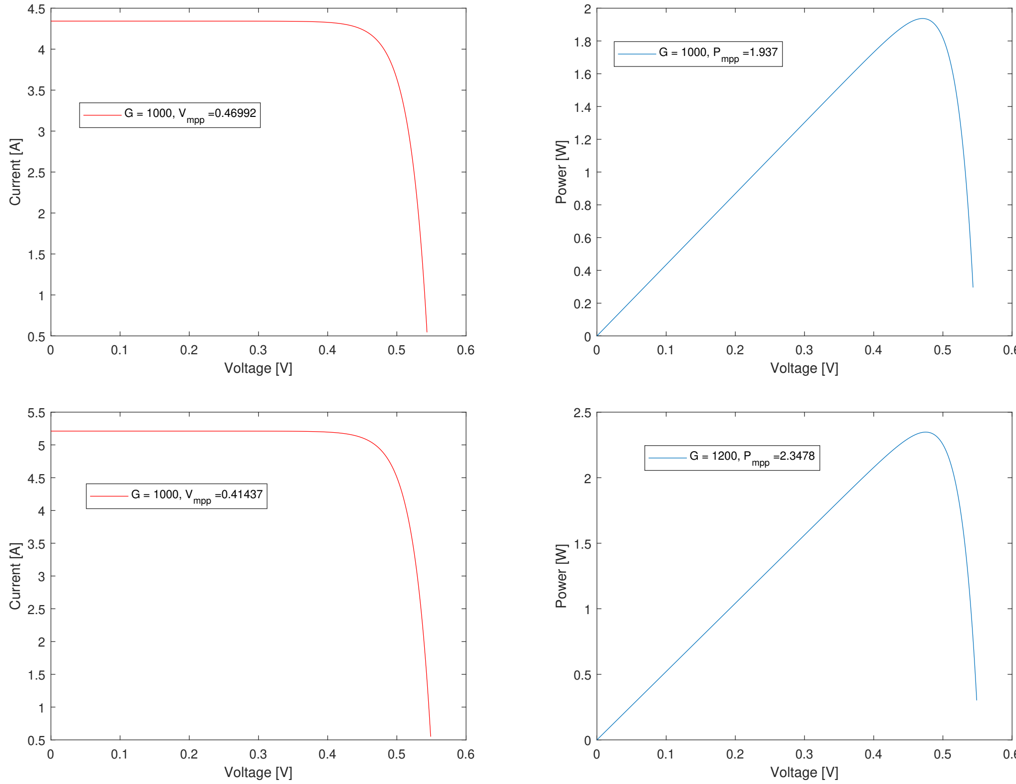

Figure 20: *Maximum Power Point, G = 1000, 1200*
2 of 2

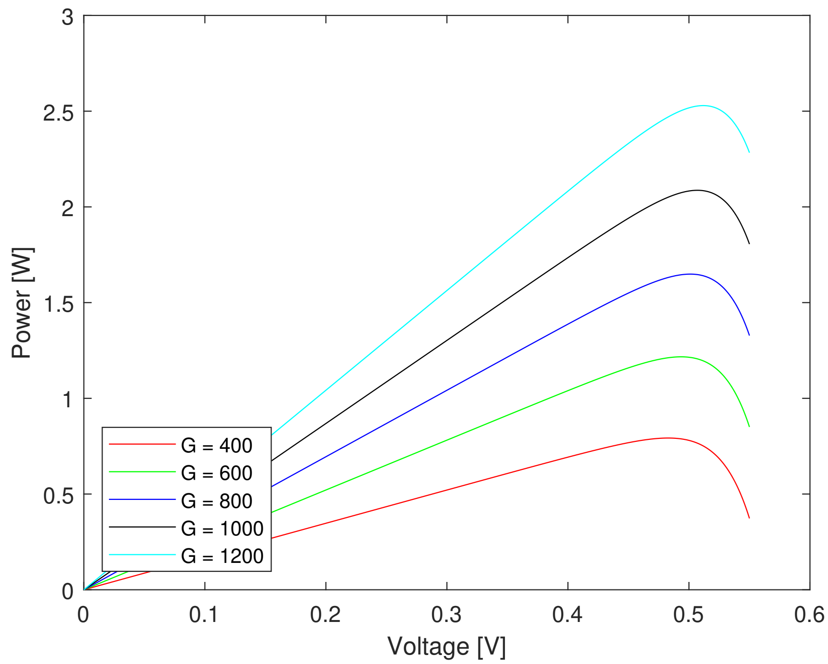

Figure 21: *Power depence on voltage. The different lines symbolizes different irradiances G.*

4 Lookup table creation

In the Tables 1 and 2 are two examples of lookup tables. Table 1 is two dimensional with two inputs. Table 2 is like in the task with voltage as input and current as output. This lookup tables are just for visualisation. So there are less values in it. The code to built these two Arrays is in Listing 1 Line 148 - 180.
With the commad tic and toc would be stoped the simulation time from the code in Listing 1 Line 64 - 97. The stoped time was 6.9 seconds. The model was simulated five times with different irradiances. The averrage for each simulation is so $\frac{6.9s}{5} = 1.4s$.
The next step was to comment out the saves to the arrays and measure the time again. This total time was 6.6s. Thats 1.3s per simulation. Not a big difference to the preview time.
Finally, the time was measured to read the array. The result was 0.03s and much smaller then one simulation run. The measurements are not very meaningful and are difficult to reproduce.
The advantage is that if the data is often needed, the simulation does not always have to be restarted. Thats saves time and energy. This is helpful in simulations which often have to be repeated.

	0Ω	$40m\Omega$	$80m\Omega$	$120m\Omega$	$160m\Omega$	$200m\Omega$	$240m\Omega$	$280m\Omega$
$400^{W}/_{m^2}$	0	0.1207	0.2413	0.3619	0.4825	0.6018	0.7049	0.7347
$600^{W}/_{m^2}$	0	0.2715	0.543	0.8141	1.0657	1.1161	1.015	0.9055
$800^{W}/_{m^2}$	0	0.4827	0.9652	1.427	1.4834	1.2872	1.1107	0.971
$1000^{W}/_{m^2}$	0	0.7542	1.5069	1.9265	1.6412	1.3686	1.1648	1.0114
$1200^{W}/_{m^2}$	0	1.086	2.1503	2.1538	1.7291	1.4216	1.2026	1.0407

Table 1: *Lookup Table with G and R_{Load} as input and P as output*

$Voltage\,[V]$	0	0.022	0.044	0.066	0.088	0.11	0.132	0.154
$Current\,[A]$	4.3424	4.3424	4.3423	4.3423	4.3423	4.3423	4.3422	4.3422

Table 2: *Lookup Table with G and R_{Load} as input and P as output*

5 Temperature Effect on PV Performance

In this section, the influence of temperature was included. For this purpose, the simulink model was adjusted as shown in Figure 22. The model was simulated with the temperatures 15°C, 20°C, 30°C, 40°C and 50°C. The values were converted from Celsius to Kelvin for the Input. The code is in Listing 1 Line 186 - 241.

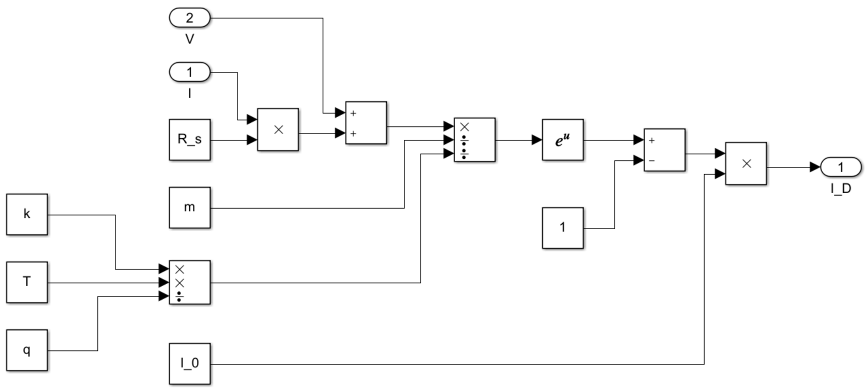

Figure 22: *Update of the Figure 5. Now with the influence of temperature.*

In Figure 23 and 24 are the plots with current and power depending on the voltage at different temperatures. Mathematically, it is true that the output power increases with higher temperature. Because V_t gets bigger with increasing temperature T.

$$T_t = \frac{k * T}{q} \tag{5}$$

The larger V_t effects that the loss current become smaller. This can be seen in Equation 3. The plots prove this calculation, seen in Figure 23 and 24. On this figures is also visible that that the influence is only visible with a large R_{load}. However, this is not what can be expected physicaly. Solar cells produces more energy by lower temperatures [1] [2].

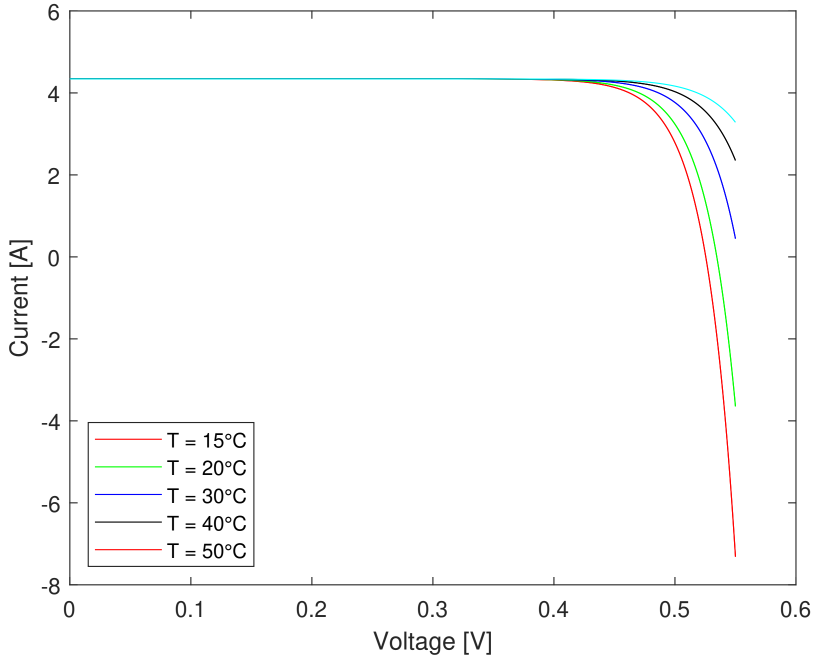

Figure 23: *Current dipence on Voltage with a constant irradiance. The different lines are the different temperatures 15°C, 20°C, 30°C, 40°C and 50°C.*

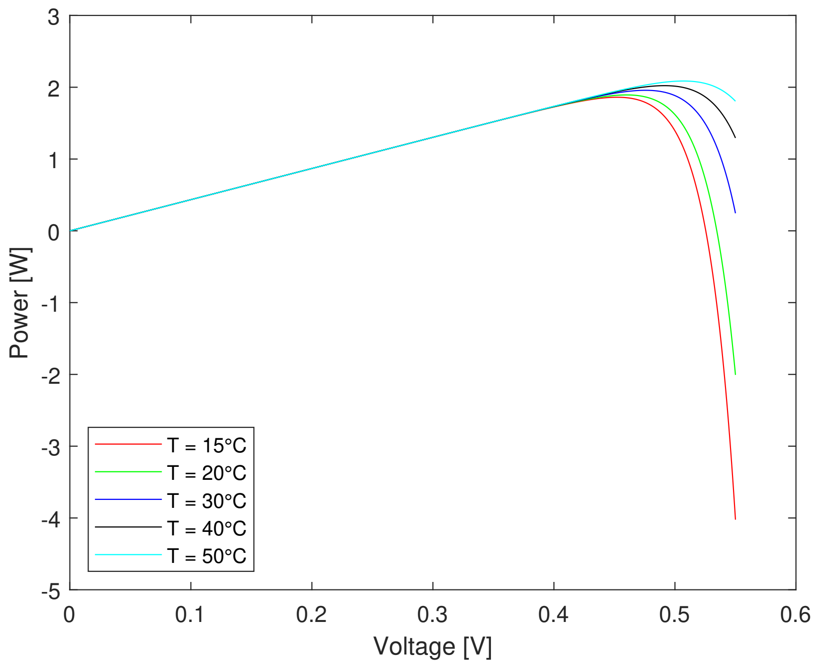

Figure 24: *Power dipence on Voltage with a constant irradiance. The different lines are the different temperatures 15°C, 20°C, 30°C, 40°C and 50°C.*

References

[1] U. Tietze, C. Schenk, and E. Gamm, *Halbleiter-Schaltungstechnik (German Edition)*. Springer, 2012.

[2] A. Niederl, "Leistungssteigerung von Photovoltaikanlagen durch Modulkühlung." Website. Online available on: `https://www.tugraz.at/fileadmin/user_upload/Events/Eninnov2014/files/lf/LF_Niederl.pdf` ; called on 30. October 2018.

List of Figures

List of Tables

Anhang

A Matlab Code

Listing 1: *Matlabcode*

```
% Variables for MBD Unit2
%1.1
clear all;

J_sc = 343;      %A/m2
G = 1000;        %W/m2
G_n = 1000;      %W/m2
A = 0.01266;     %m2
R_s = 1e-6;      %Ohm
R_sh = 1000;     %Ohm
I_0 = 2.8e-9;    %A
Vt = 25.85e-3;   %V
m = 1;           % 1= ideal diode
V = 0.5;           %V
k= 1.38064852*10^-23;
T = 26.85 +273.15;
q=1.602176565*10^-19;

save var.mat;

%%
% Logdaten auslesen
% logsout{2}.Values.Data
% 2.4
figure(99)
subplot(2,2,1);
plot(logsout{2}.Values.Time, logsout{2}.Values.Data)
xlabel('Time␣[sec]')
ylabel('Current␣[A]')

subplot(2,2,2);
plot(logsout{2}.Values.Time, logsout{4}.Values.Data ./logsout{2}.Values.
   Data)
xlabel('Time␣[sec]')
ylabel('Resistance␣[Ohm]')

subplot(2,2,3);
plot(logsout{2}.Values.Time, logsout{4}.Values.Data)
xlabel('Time␣[sec]')
ylabel('Voltage␣[V]')

subplot(2,2,4);
plot(logsout{2}.Values.Time, logsout{3}.Values.Data)
```

```
        xlabel('Time␣[sec]')
        ylabel('Power␣[W]')

        %saveas(gcf, '../../documentation/figures/subplots.eps', 'eps')

        %%
        % XY Plot with u & i
        % 2.4
        sim('unit2');
        x=u.get('Data');
        y=i.get('Data');
        plot(x,y);
        xlabel('Voltage␣[V]')
        ylabel('Amps␣[A]')


        %%
        %sim('unit2', 'G', [400, 600,800,1000,1200])
        % Simuliere mit verschiedenen G-Werten und speichere die Werte
        % Sim Zeit nehmen mit tic - toc
        % 3.1 & 4
        tic

        G = 400;
        sim('unit2');
        u400=u.get('Data');
        i400=i.get('Data');

        G = 600;
        sim('unit2');
        u600=u.get('Data');
        i600=i.get('Data');

        G = 800;
        sim('unit2');
        u800=u.get('Data');
        i800=i.get('Data');

        G = 1000;
        sim('unit2');
        u1000=u.get('Data');
        i1000=i.get('Data');

        G = 1200;
        sim('unit2');
        u1200=u.get('Data');
        i1200=i.get('Data');

        toc

        %%
```

```
% Zeit um Array zu lesen
tic
[P_max, max_ind] = max(u1200.*i1200)
toc

%%
%XY Plot Leistung über Spannung
% 3.2
plot(u400, i400.*u400, 'r')
hold on
xlabel('Voltage␣[V]');
ylabel('Power␣[W]');
plot(u600, i600.*u600, 'g')
plot(u800, i800.*u800, 'b')
plot(u1000, i1000.*u1000, 'k')
plot(u1200, i1200.*u1200, 'c')
legend({'G␣=␣400', 'G␣=␣600', 'G␣=␣800', 'G␣=␣1000', 'G␣=␣1200'},'Location','southwest')
hold off
%saveas(gcf, '../../documentation/figures/xy_vp.eps', 'eps')

%%
%XY Plot Strom über Spannung
% 3.2
plot(u400, i400, 'r')
hold on
xlabel('Voltage␣[V]');
ylabel('Current␣[A]');
plot(u600, i600, 'g')
plot(u800, i800, 'b')
plot(u1000, i1000, 'k')
plot(u1200, i1200, 'c')
legend({'G␣=␣400', 'G␣=␣600', 'G␣=␣800', 'G␣=␣1000', 'G␣=␣1200'},'Location', 'southwest')
hold off
%saveas(gcf, '../../documentation/figures/xy_vc.eps', 'eps')

%%
% XY Plot mit den max values in der Legende
[P_value, index_peak] =  max(u1000.*i1000);
figure(3);
plot(u1200, i1200, 'r')
xlabel('Voltage␣[V]');
ylabel('Current␣[A]');
legend(strcat('G␣=␣1000,␣V_{mpp}␣=␣', num2str(u1000(index_peak))));

%%
% XY Plot mit den max values in der Legende
% 3.3
figure(2);
plot(u1200, u1000.*i1000);
```

```
        xlabel('Voltage␣[V]');
        ylabel('Power␣[W]');
        legend(strcat('G␣=␣1200,␣P_{mpp}␣=␣', num2str(P_value)));

        %%
        %%Erstelle die Arrays zur Visualisierung für die Lookuptables mit
           undersampling
        % 4.0
        % i1arr = i1.get('Data');
        % u2arr = u1.get('Data');
        %
        % c2 = 1;
        % for c = 1:8
        %     p400(c) = u400(c2) * i400(c2);
        %     p600(c) = u600(c2) * i600(c2);
        %     p800(c) = u800(c2) * i800(c2);
        %     p1000(c) = u1000(c2) * i1000(c2);
        %     p1200(c) = u1200(c2) * i1200(c2);
        %     c2 = c2 + 40;
        % end

        % for c = 1:1000
        %     iv(c) = i1arr(c2);
        %     uv(c) = u2arr(c2);
        %     c2 = c2 + 1;
        % end
        %
        % %plt = [p400; p600; p800; p1000; p1200];
        % plt2 = [iv; uv];
        % toc
        % figure(4);
        % plot(uv, iv);
        % xlabel('Voltage [V]');
        % ylabel('Ampere [A]');
        % legend();
        % toc
        %%Erstelle den Sourcecode der Arrays für LaTex
        %latextable = latex(sym(vpa(round(plt2, 4))))
        %latextable = latex(sym(vpa(round(plt, 4))))

        %%
        % Simulation mit verschiedenen Temperaturen
        % 5.2

        T = 15 + 273.15;
        sim('unit2');
        u15=u.get('Data');
        i15=i.get('Data');

        T = 20 + 273.15;
        sim('unit2');
```

```
        u20=u.get(’Data’);
        i20=i.get(’Data’);

        T = 30 + 273.15;
        sim(’unit2’);
        u30=u.get(’Data’);
        i30=i.get(’Data’);

        T = 40 + 273.15;
        sim(’unit2’);
        u40=u.get(’Data’);
        i40=i.get(’Data’);

        T = 50 + 273.15;
        sim(’unit2’);
        u50=u.get(’Data’);
        i50=i.get(’Data’);


        %%
        %XY Plot Leistung über Voltage
        % 5.3
        plot(u15(1:end), i15(1:end).*u15(1:end), ’r’)
        hold on
        xlabel(’Voltage␣[V]’);
        ylabel(’Power␣[W]’);
        plot(u20(1:end), i20(1:end).*u20(1:end), ’g’)
        plot(u30(1:end), i30(1:end).*u30(1:end), ’b’)
        plot(u40(1:end), i40(1:end).*u40(1:end), ’k’)
        plot(u50(1:end), i50(1:end).*u50(1:end), ’c’)

        legend({’T␣=␣15C’, ’T␣=␣20C’, ’T␣=␣30C’, ’T␣=␣40C’, ’T␣=␣50C’},’Location’,’southwest’)
        hold off
        %saveas(gcf, ’../../documentation/figures/xy_vp_t.eps’, ’eps’)

        %%
        %XY Plot Strom über Spannung
        % 5.3
        plot(u15(1:end), i15(1:end), ’r’)
        hold on
        xlabel(’Voltage␣[V]’);
        ylabel(’Current␣[A]’);
        plot(u20(1:end), i20(1:end), ’g’)
        plot(u30(1:end), i30(1:end), ’b’)
        plot(u40(1:end), i40(1:end), ’k’)
        plot(u50(1:end), i50(1:end), ’c’)
        legend({’T␣=␣15C’, ’T␣=␣20C’, ’T␣=␣30C’, ’T␣=␣40C’, ’T␣=␣50C’},’Location’, ’southwest’)
        hold off
        %saveas(gcf, ’../../documentation/figures/xy_vc_t.eps’, ’eps’
```