

Kurzanleitung zum erstellen und verwenden von PV-Look-up Tabellen in Plecs

Um ein beliebiges PV-Modul in Plecs zu simulieren muss zuvor ein Look-up Tabelle (LUT) generiert und in plecs eingebunden werden. Diese Kurzanleitung soll auf die dafür notwendigen Schritte eingehen. Für die Modellierung der Kennfelder wird das mathematische Model von U.Boeke [1] verwendet. Eine Look-up Tabelle wird mittels Skript generiert und in einem 3D-LUT-Block in Plecs geladen.

1 Vorraussetzungen

Für die Modellierung werden zunächst folgende Daten über das PV-Modul benötigt:

- Leerlaufspannung: V_{OC}
- Kurzschlussstrom: I_{SC}
- Spannung im Maximum Power Point(MPP): V_{MPP}
- Strom im MPP: I_{MPP}
- Leerlaufspannung bei geringer Sonneneinstrahlung (ca. $200\text{W}/\text{m}^2$): V_{MIN}
- Spannungstemperaturkoeffizient: T_{CV}
- Strom-Temperaturkoeffizient: T_{CI}

Weiterhin werden die Programme Matlab[2] oder GNU Octave benötigt. Alle beigefügten Skripte wurden mit Octave Version 7.3.0 unter Linux getestet. Für die Verwendung mit Matlab müssen evtl. Änderungen vorgenommen werden.

2 Verwendung

2.1 LUT generieren

In der Datei „pv_lut.m“ müssen zunächst die oben genannten Parameter des PV-Moduls eingegeben werden. Auch können hier z.B. Variablen für die Stützstellenanzahl der Kennlinien festgelegt werden. Für das Ausführen des Skripts sollten die Hinweise in den Kommentaren beachtet werden.

2.2 Ausgabe überprüfen

Die Binäre „pv_lut.mat“ Datei, welche im selben Ordner wie das Skript generiert wird, kann nun in Plecs eingebunden werden. Es ist allerdings ratsam das Ergebnis vorher zu überprüfen. Da es sich um eine Binärdatei handelt, kann diese nicht lesbar im Texteditor angezeigt werden. Mit dem beigefügten Skript „look_into_binary_lut.m“ können die Daten im Befehlsfenster von Octave angezeigt werden. Mit dem Skript „plot_Isurf.m“ wird das Kennlinienfeld als 3D-Plot für eine vom Nutzer eingegebene Temperatur angezeigt. Wichtig ist, dass das Skript im selben Ordner wie das LUT legt. Andernfalls müssen die Pfade im Code angepasst werden.

2.3 PV-Modul in Plecs

In Plecs wird das PV Modul durch ein 3D-Table Block und einer Idealen Stromquelle nachgebildet. Der Ausgang des 3D-Table wird dabei mit dem Eingang der „Current Source (controlles)“ verbunden. Dieses System kann wiederum als ein PV-Block-Subsystem in einer Schaltung verwendet werden. Ein Beispiel dafür ist in der Datei „pv_lut.plecs“ angefügt.

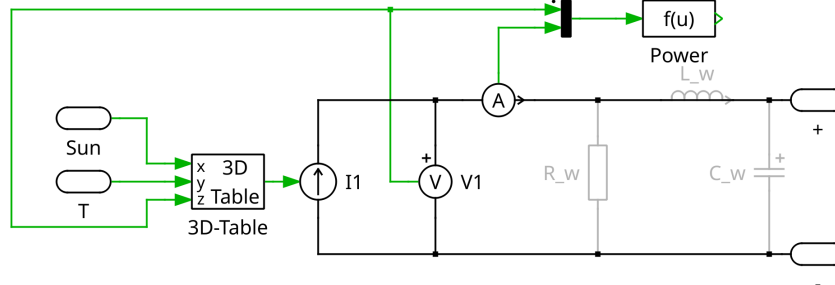


Abbildung 1: PV Modul als Subsystem in Plecs

2.4 Einbindung des LUT in Plecs

Um ein LUT in Plecs zu verwenden, muss es bei der Initialisierung des PV-Blocks geladen werden. Der Befehl „load('pv_lut');“ wird dafür im Mask Editor unter dem Reiter „initialization“ eingefügt. Der Mask Editor kann STRG + M aufgerufen werden.

Als nächstes werden die Blockparameter des 3D-Table angepasst. Zunächst muss der Array Name angegeben werden. Hierbei ist zu beachten, dass damit nicht der Dateiname des LUTs gemeint ist. Es handelt sich um den Namen des Arrays innerhalb von „pv_lut.mat“. Dieser wurde bereits der Arraydeklaration im Octave-Skript Zeile 61 festgelegt.

Bleibt noch die Bestimmung der Vektoren für die x,y,z Parameter. In diesem Fall sind die Daten im LUT folgendermaßen formatiert:

Der Parameter x entspricht der Sonneneinstrahlung, y ist die Temperatur und z die Spannung. Der Vektor muss entsprechend der Menge, sowie Wertebereich eines Parameters angegeben werden:

$$[\text{Minimum:Schrittweite:Maximum}]$$

Beispielhaft für die Sonneneinstrahlung, welche im LUT von 0 bis 1000W/m² hinterlegt ist und insgesamt 10 Stützstellen hat:

$$[0:100:1000]$$

Damit wird die Ideale Stromquelle entsprechend der Look-up Tabelle gesteuert und alle Schritte zu einer erfolgreichen Simulation eines PV-Moduls durchgeführt.

Literatur

- [1] Ulrich Boke. „A simple model of photovoltaic module electric characteristics“. In: *2007 European Conference on Power Electronics and Applications*. 2007, S. 1–8. DOI: 10.1109/EPE.2007.4417572.
- [2] The MathWorks Inc. *MATLAB*. Natick, Massachusetts, United States, 2022. URL: <https://www.mathworks.com>.