

Zusammenfassung zum Capstoneprojekt "MPP-Tracker für einphasigen Wechselrichter auf STM32-Basis"

Jonathan Koeppen, Tobias Magh, Nils Meierhöfer, Moritz Prenzlów, Joshua-Jérôme Reichmann

Betreuung durch Prof. C. Dick

15.07.2024

Inhaltsverzeichnis

1	Einleitung	2
2	Genereller Aufbau	2
3	Parametrierung	3
4	Modellierung einer Lookup-Table	4
5	STM32 Einbindung in PLECS	5
6	Zusammenfassung und Ausblick	6

1 Einleitung

Dieses Projekt beschäftigt sich mit der Modellierung eines einphasigen PV-Wechselrichters mit Maximum Power Point Tracking (MPPT), der zum Testen von PV-Modulen eingesetzt wird. Die Implementierung erfolgt in der Software PLECS. Schwerpunkte sind zum einen die Modellierung des PV-Moduls in Abhängigkeit von Temperatur und Lichtintensität sowie die Implementierung eines Algorithmus für das MPPT Nachführung. Des Weiteren werden die Detektion eines Lastabwurfs, eine intelligente Abschaltung sowie eine Startroutine untersucht. Darauf aufbauend soll die implementierte Codegenerierung für einen STM32 aus PLECS heraus durchgeführt und getestet sowie die weitere technisch mögliche Umsetzung untersucht werden.

2 Genereller Aufbau

Die Schaltungstopologie basiert auf einem klassischen einphasigen Wechselrichter. Als Eingang dient ein PV-Modulstrang mit großem DC-Link-Kondensator und Entladewiderstand. Der Wechselrichter ist als H-Brücke ausgeführt. Das AC-Signal wird mit einem Tiefpassfilter geglättet. Die Last wird über einen Transformator als galvanisch getrennter Ausgang angeschlossen. Ein STM32-Mikrocontroller steuert und regelt den Wechselrichter, indem er Schaltsignale aus einer Sinus-PWM an die Leistungshalbleiter sendet. Ein MPP-Tracker stellt dabei den Modulationsgrad für die Sinus-PWM resultierend aus der Eingangsgröße des Tiefpasskondensatorstroms bereit.

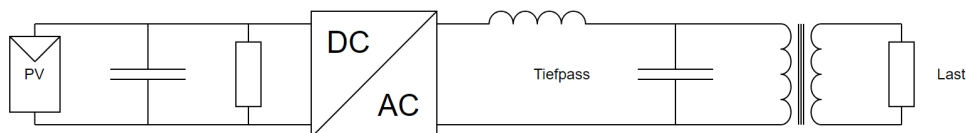


Abbildung 1: Übersicht der Schaltung

3 Parametrierung

Die Analyse und Visualisierung der Ergebnisse erfolgt durch die Generierung und Auswertung von CSV-Dateien, die eine strukturierte Darstellung der Daten ermöglichen. Ein PLECS-Simulationsskript wird verwendet, um einen mehrdimensionalen Parameter-Sweep durchzuführen, der verschiedene Kombinationen von Eingabeparametern systematisch untersucht. Parallel dazu wurde eine Parametrisierungsschnittstelle aufgebaut, die eine rudimentäre Eingabe und Anpassung der Parameter erlaubt. Das primäre Ziel dieser Methodik ist die Ermittlung eines optimalen Eingabevektors, der ein vorgegebenes Kriterium bestmöglich erfüllt. Verschiedene Kriterien konnte so getestet werden.

Hauptsächlich wurde jedoch getestet, in wie weit es zu Oszillationen innerhalb unseres LC-Tiefpassfiltergliedes kommt, wenn z.B. die Last plötzlich entfernt oder Kurzgeschlossen wird. Zudem interessierte uns, welche Auswirkungen eine Veränderung der PV-Leistung auf die Schaltung haben könnte.

Folgende Erkenntnis lässt sich festhalten: Durch Verwenden einer Schaltfrequenz von 48 kHz und einer Netzfrequenz von 500 Hz, erreichen wir eine ausreichende Distanz zur LC-Resonanzfrequenz von 6 kHz. Abschalteneinrichtungen für z.B. den Kurzschlussfall wurden erprobt, aufgrund der Komplexität welche die Schaltung dann erreichte, wurde für den Rahmen des Capstone-Projekts an dieser Stelle jedoch ein Schlussstrich gezogen.

Für weitere Informationen ist im Unterordner `Automatisiert_Testen` nachzuschauen.

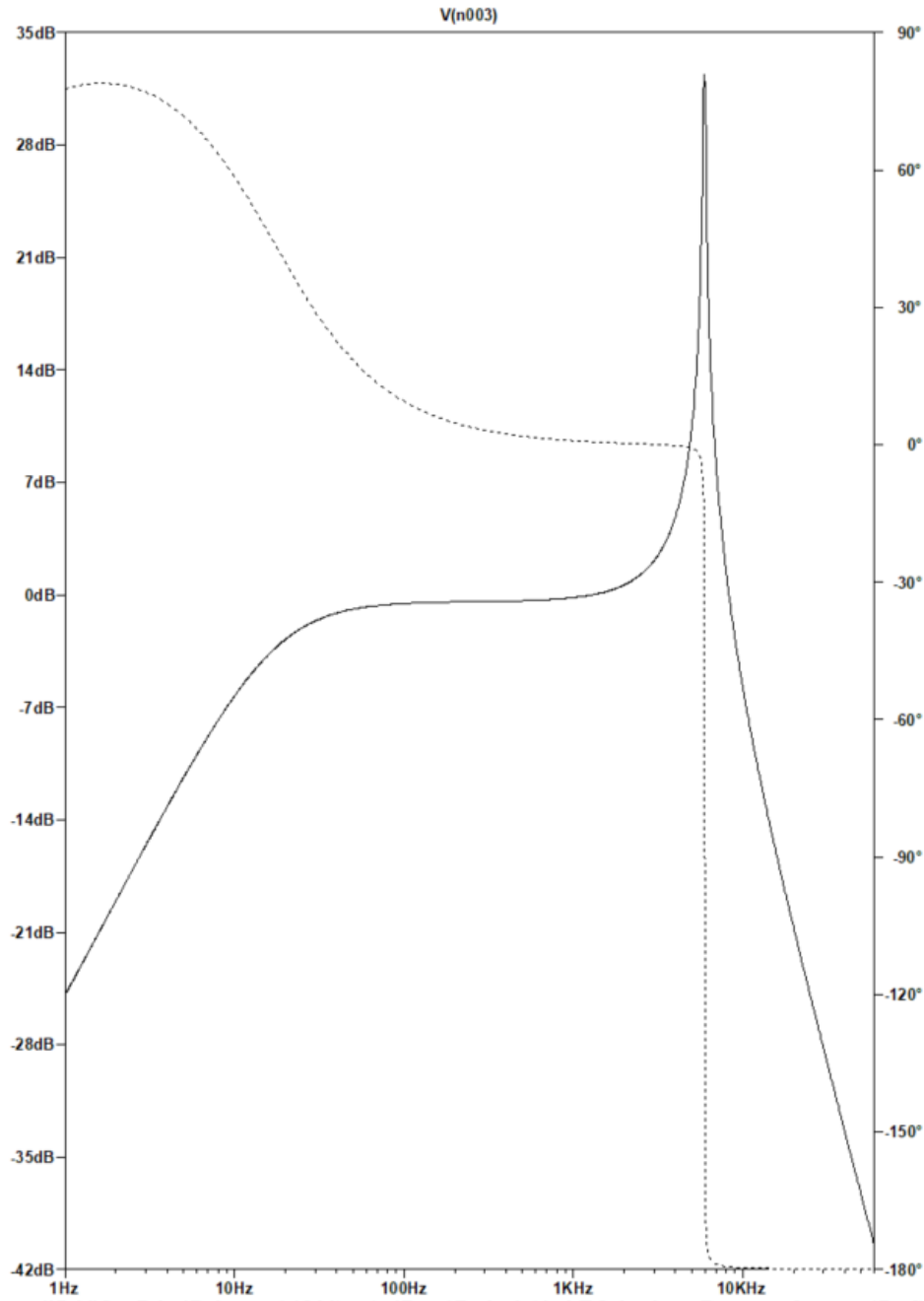


Abbildung 2: LT-Spice Ausgabe zur Resonanz des LC-Tiefpasses

4 Modellierung einer Lookup-Table

Um ein beliebiges PV-Modul in PLECS nachzubilden, wird ein Subsystem verwendet, dessen Kernelement eine 3D-Lookup-Tabelle ist, welche eine ideale Stromquelle steuert. Wie eine solche LUT für beliebige PV-Module erstellt und in PLECS eingebunden werden kann, ist in den angehängten Dokumenten beschrieben. Eingangsgrößen des Systems sind die Temperatur und die Sonneneinstrahlung. Innerhalb des Subsystems werden die Spannung und der Strom an den Ausgangsklemmen gemessen. Die Strommessung dient allein der Berechnung der Ausgangsleistung, welche durch eine Probe abgelesen werden kann. Der Block ist so aufgebaut, dass mehrere PV-Paneele in Reihe geschaltet werden können. Aus diesem Grund ist es notwendig, die gemess-

sene Spannung durch die Anzahl der Panele zu teilen, bevor sie als Eingangsgröße der 3D-LUT verwendet werden kann. Die Anzahl der PV-Module wird in der Variable „pv_strings“ unter den „Simulation Parameters“ initialisiert und in den Blockparametern dem Subsystem übergeben. Bei der Berechnung der LUT kann die Anzahl ebenfalls berücksichtigt werden, jedoch muss dann „pv_strings“ in PLECS mit „1“ initialisiert sein. Neben den Eigenschaften des Moduls können auch die elektrischen Eigenschaften der verwendeten Anschlussleitungen berücksichtigt werden. Dazu sind ersatzweise eine Spule, ein Kondensator sowie ein Widerstand mit typischen Größen angelegt, welche wiederum abhängig von der Leitungslänge sind. Diese Länge wird ebenfalls in den „Simulation Parameters“ unter „l_wire“ initialisiert.

Für weitere Informationen wird an dieser Stelle auf den Unterordner "PV-LUT" verwiesen.

5 STM32 Einbindung in PLECS

Mit der STM32 Plecs-Library ist eine schnelle Adaption von existierenden Simulationen auf reale Hardware möglich. Ein- und Ausgänge des uCs werden durch Blöcke in der Simulation definiert. Die Codegenerierung und Kompilierung findet dann direkt in Plecs statt. Während das Programm auf der Hardware ausgeführt wird, ist ein Betrachten und Setzen von Größen in der bestehenden Simulationsumgebung möglich. In diesem Projekt wurde versucht, den STM als "Hardware in the Loop"(HIL) zu betreiben. HIL bedeutet ein Ausführen des Codes auf der eigentlichen Hardware während der Leistungsteil simuliert wird. Für einfache Schaltungen ohne Induktivitäten, Kapazitäten Dioden war der Versuch erfolgreich. Da die Werte der simulierten Ein- und Ausgänge des uCs jedoch nicht in Echtzeit und zu langsam übertragen werden, sind Simulationen komplizierterer Schaltungen nicht möglich. Hierfür muss auf für diese Anwendung entwickelte Hardware zurückgegriffen werden. Es wurde jedoch festgestellt, dass die Anpassung der Simulation des Gleichrichters an den STM sehr einfach ist und mit vorhandener Hardware ein Testen und Debuggen gut möglich wäre. Da Plecs den generierten Code zur Verfügung stellt, könnte dieser auch als Basis für eine von Plecs unabhängige Programmierung verwendet werden. Für weitere Informationen wird an dieser Stelle auf den Unterordner STM32 verwiesen.

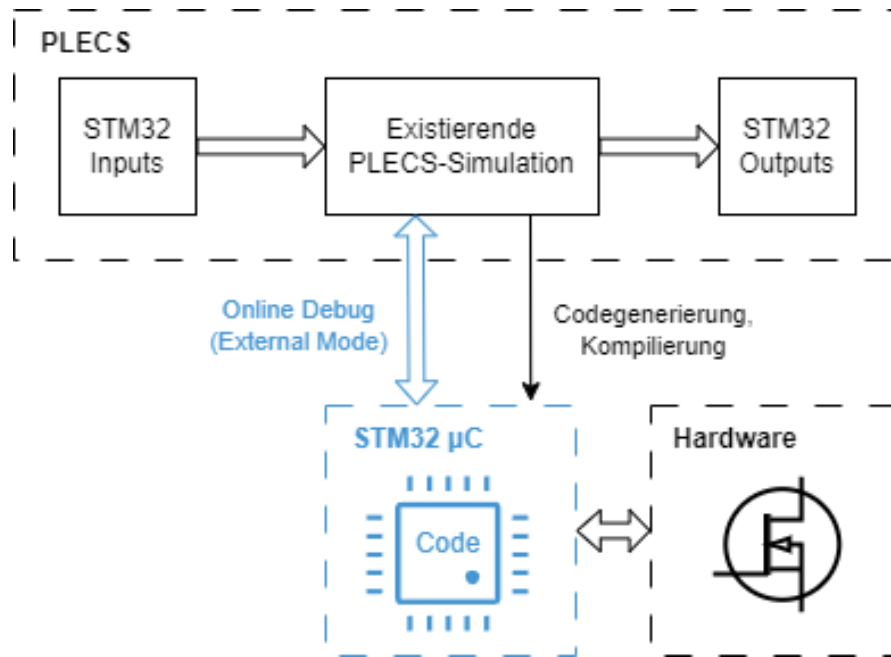


Abbildung 3: Übersicht Anbindung des STM32 an PLECS