



EMV-konformes Design

Transient Simulation(MECH-M-3-SVE-TSI-ILV)

Master Mechatronic & Smart Technologies

3. Semester

Lehrveranstaltungsleiter: Thomas Gadner

Jahrgang: MA-MECH-23-BB

Verfasser: Lukas Sieß

14. Februar 2025

Inhaltsverzeichnis

1 Einleitung

Dieser Bericht dokumentiert die Umsetzung der zweiten Aufgabe im Bereich Regelungstechnik und Leistungselektronik. Ziel ist die praxisnahe Anwendung der theoretischen Grundlagen aus den ersten beiden Semestern, indem ein Leistungselektroniksystem modelliert und geregelt wird. Dazu wird ein digitaler Regler entworfen und auf einer geeigneten Plattform implementiert.

Die Aufgabenstellung gliedert sich in zwei wesentliche Arbeitsschritte:

- **Modellierung eines Leistungselektronik-Wandlers in PLECS** Zunächst wird ein Modell eines Leistungselektronik-Wandlers mit PLECS erstellt. Dieses Modell bildet die Basis für die spätere Regelung. Um die Komplexität der Regelung zu minimieren, wird das System so ausgelegt, dass die Dynamik möglichst träge ist. Dadurch kann eine Regelung mit klassischen kontinuierlichen Methoden erfolgen, sofern die Abtastrate hoch genug gewählt wird. (Siehe Abbildung:)
- **Implementierung des Modells auf einer RTBox mit einem STM-Mikrocontroller als Regler** Nach der Modellierung wird das System auf einer RTBox implementiert. Die Regelung erfolgt über einen STM-Mikrocontroller, der die Steueralgorithmen ausführt und das Systemverhalten in Echtzeit überprüft.

Dieser Bericht legt den Fokus auf die präzise Modellbildung, die Auswahl geeigneter Regelparameter und die Implementierung auf der RTBox mit einem STM-Mikrocontroller.

2 Kapitel 1: Modellierung eines Leistungselektronik-Wandlers in PLECS

Die Modellierung des Leistungselektronik-Wandlers ist ein zentraler Schritt zur Systemanalyse und Regelung. In diesem Kapitel wird die theoretische Grundlage erläutert, das Modell detailliert beschrieben und die Wahl der Modellstruktur sowie deren Auswirkungen auf das dynamische Verhalten diskutiert. Mögliche Vereinfachungen zur effizienten Implementierung des Reglers werden ebenfalls betrachtet. (Siehe Abbildung: ?? und ??)

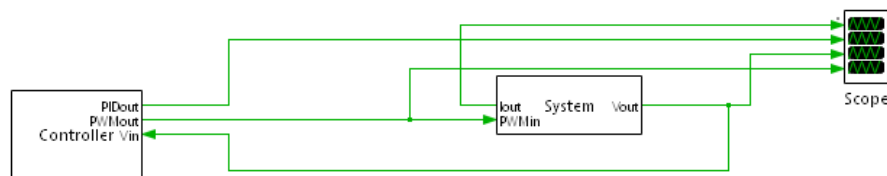


Abbildung 2.1: FFT LTC3639 mit Filter

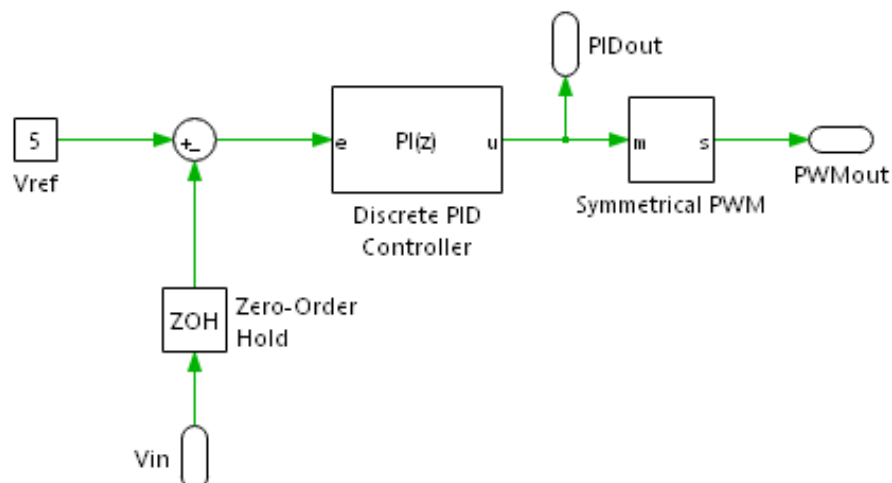


Abbildung 2.2: FFT LTC3639 mit Filter

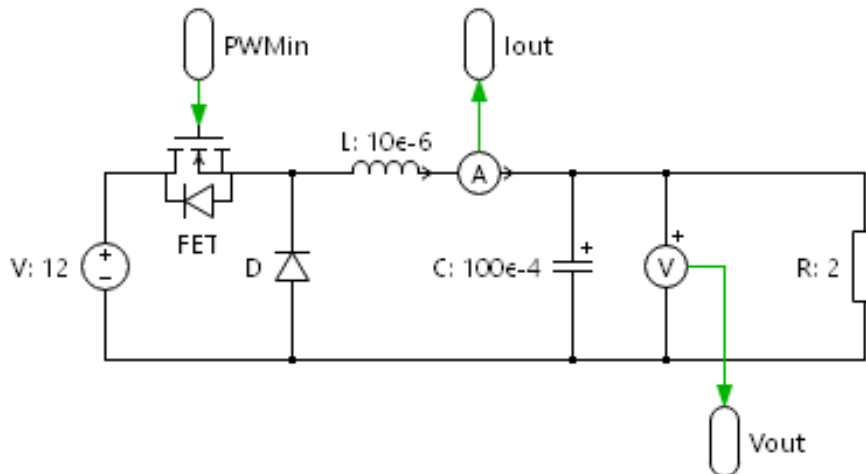


Abbildung 2.3: FFT LTC3639 mit Filter

2.1 PID Auslegung

Ein besonderer Schwerpunkt lag auf der Auslegung eines PID-Reglers. Verschiedene Methoden zur Reglerparametrierung wurden getestet, darunter System Identification, State-Space-Modelle, Pole Placement und PID-Tuning. Keine dieser Methoden lieferte die gewünschten Ergebnisse. Auch die Übernahme von Parametern aus der Fachliteratur – z. B. aus dem Paper "Optimally Designed PID Controller for a DC-DC Buck Converter via a Hybrid Whale Optimization Algorithm with Simulated Annealing" – führte nicht zum Erfolg. Daher wurden die PID-Parameter empirisch angepasst, um eine stabile und performante Regelung zu gewährleisten. Die finalen Werte lauten:

- Reglertyp: Discrete PI-Regler
- P-Anteil: 0.002
- I-Anteil: 7

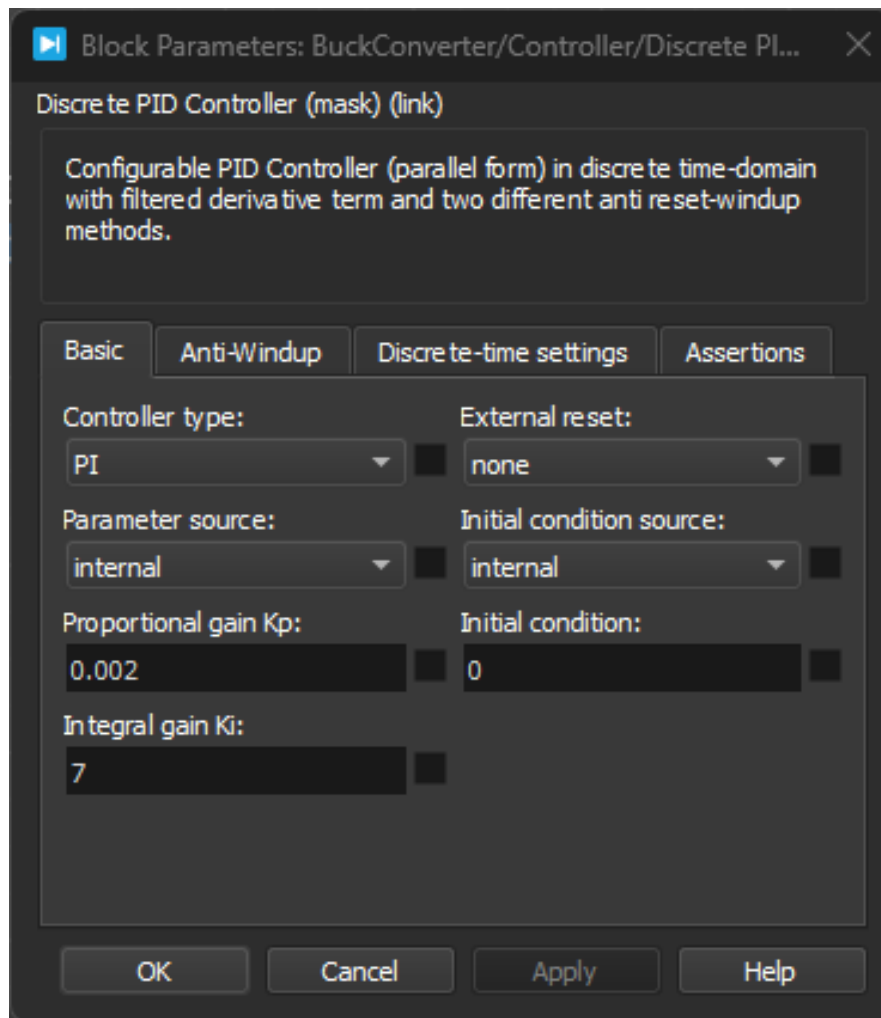


Abbildung 2.4: FFT LTC3639 mit Filter

3 Task 2

Nach der erfolgreichen Modellierung in PLECS wurde die Hardware-Implementierung vorbereitet. In diesem Kapitel werden die notwendigen Schritte zur Umsetzung erläutert. Dabei stehen die Echtzeitfähigkeit der RTBox, die Anbindung des Mikrocontrollers sowie die Entwicklung der Steueralgorithmen im Fokus. Die Herausforderungen der Echtzeitregelung und die erforderlichen Anpassungen werden ebenfalls behandelt. Abschließend erfolgt eine Analyse der Performance basierend auf Simulationen und experimentellen Ergebnissen.

Zur Umsetzung wurden zwei Subsysteme definiert – eines für den Controller, eines für das geregelte System. Diese Trennung ermöglicht eine separate Parametrierung der Coder-Optionen für eine optimierte Code-Generierung. Die Ein- und Ausgänge wurden wie folgt konfiguriert und miteinander verschaltet:

- **System-Subsystem:** Enthält PLECS PWM Capture sowie PLECS Analog Out für Strom- und Spannungsmessung (Siehe Abbildung: *ModelTask2System.png*)
- **Controller-Subsystem:** Enthält STMPWMOut für die Ansteuerung sowie STMAIn (Siehe Abbildung: *ModelTask2Controller.png*)

Die Simulation wurde mit zwei verschiedenen Lasten ($R_{Load} = 2\Omega$ und) durchgeführt, um die Performance des Reglers unter unterschiedlichen Bedingungen zu bewerten. Dabei wurde das PLECS-Modell schrittweise für die Hardware-Simulation mit der RT-Box und dem STM32 Nucleo angepasst. Die in den Coder-Optionen festgelegten Parameter waren:

- Controller: Target = STM32G4x; Scheduling Step Size = (Siehe Abbildung: Settings CoderOptions *ModelTask2ControllerTarget.png*)
- System: Target = PLECS RT Box 2; Scheduling Step Size = (Siehe Abbildung: Settings CoderOptions *ModelTask2SystemTarget.png*)
- Gesamtsystem-Ansicht: Überblick der vollständigen Implementierung (Siehe Abbildung: *ModelTask2.png*)

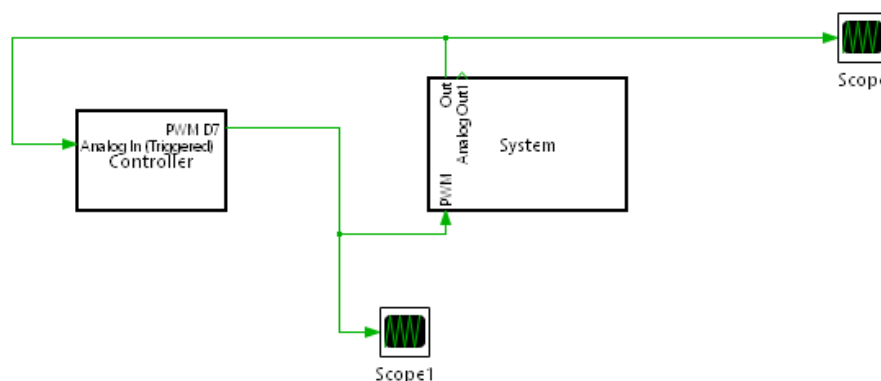


Abbildung 3.1: FFT LTC3639 mit Filter



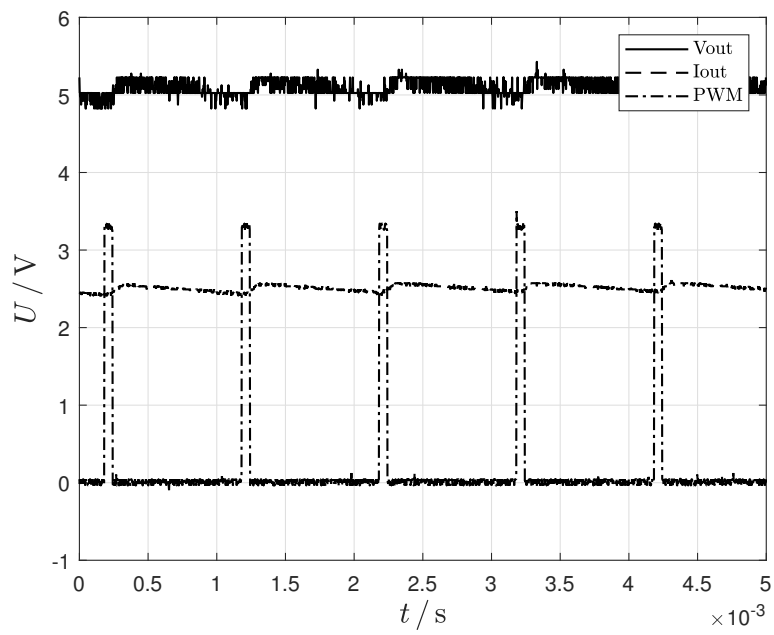


Abbildung 3.4: Messung mit $R_{load} = 2 \Omega$

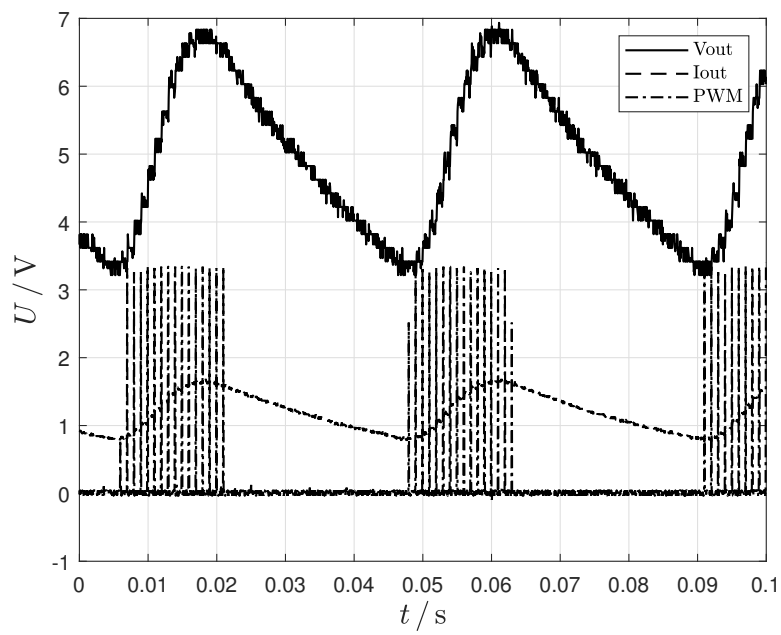


Abbildung 3.5: Messung mit $R_{load} = 4 \Omega$

A Anhang

A.1 MATLAB Skript