

BACHELORTHESIS  
Jonas Kutsche

# Hardware-in-the-Loop- Implementierung einer eingebetten Regelung für ein Batteriespeichermodell zur Bereitstellung von Momentanreserve

---

FAKULTÄT TECHNIK UND INFORMATIK  
Department Informations- und Elektrotechnik

Faculty of Computer Science and Engineering  
Department of Information and Electrical Engineering

Jonas Kutsche

# Hardware-in-the-Loop-Implementierung einer eingebetten Regelung für ein Batteriespeichermodell zur Bereitstellung von Momentanreserve

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung  
im Studiengang *Bachelor of Science Regenerative Energiesysteme und Energie-  
management*  
am Department Informations- und Elektrotechnik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Michael Erhard  
Zweitgutachter: Dr.-Ing. Georg Pangalos

Eingereicht am: 02. Oktober 2020

**Jonas Kutsche**

**Thema der Arbeit**

Hardware-in-the-Loop-Implementierung einer eingebetten Regelung für ein Batteriespeichermodell zur Bereitstellung von Momentanreserve

**Stichworte**

Hardware-In-The-Loop, Momentanreserve, Frequenzgradient (RoCoF), PLECS RT Box, Schwingungsgleichungsregelung, Frequenzschätzung

**Kurzzusammenfassung**

In dieser Arbeit werden Hardware-in-the-Loop (HiL)-Simulationen des eingebetten Reglerentwurfs der Schwingungsgleichung durchgeführt, um simulativ Momentanreserve mit einem Batteriespeicher bereitzustellen. Der Prozess der HiL-Implementierung mit der *PLECS* RT Box und einem Mikrocontroller wird vorgestellt sowie die frequenzstabilisierende Reaktion des Batteriespeichers, infolge eines Lastsprungs, simuliert . . .

**Jonas Kutsche**

**Title of Thesis**

Hardware-in-the-Loop implementation of an embedded control for a model of a Battery Energy Storage System to provide syntetic inertia

**Keywords**

Hardware-In-The-Loop, Synthetic Inertia, Rate of Change of Frequency (RoCoF), PLECS RT Box, Swing Equation-based inertia response, Frequency Estimation

**Abstract**

In this thesis, hardware-in-the-loop (HiL) simulations of the embedded controller design of the swing equation are performed to simulatively provide virtual inertia with a battery storage. The process of HiL implementation with the *PLECS* RT Box and a microcontroller is presented, and the frequency stabilizing response of the battery storage after a load jump is simulated . . .

# Inhaltsverzeichnis

<b>Abkürzungen</b>	<b>x</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation und Hintergrund . . . . .	1
1.2 Problemstellung und Zielsetzung . . . . .	2
1.3 Aufbau der Arbeit . . . . .	2
<b>2 Grundlagen</b>	<b>4</b>
2.1 Momentanreserve im Energienetz . . . . .	4
2.2 Schwingungsgleichung . . . . .	4
2.3 Echtzeitsimulation transienter Vorgänge . . . . .	6
2.4 Modellbasierte Entwicklung . . . . .	6
2.4.1 Model-in-the-Loop . . . . .	8
2.4.2 Software-in-the-Loop . . . . .	8
2.4.3 Processor-in-the-Loop . . . . .	9
2.4.4 Hardware-in-the-Loop . . . . .	9
<b>3 Referenzmodell</b>	<b>10</b>
3.1 Streckenmodell des Inselnetzes . . . . .	10
3.2 Reglermodell mit Schwingungsgleichung . . . . .	12
3.3 Kennzahlen des Referenz-Szenarios . . . . .	13
<b>4 Hardware-in-the-Loop-Implementierung</b>	<b>14</b>
4.1 Hardwarekomponenten . . . . .	15
4.1.1 Echtzeitsimulator . . . . .	15
4.1.2 Regelungshardware . . . . .	16
4.2 Signalübertragung und -verarbeitung . . . . .	16
4.2.1 Skalierung der Signalwandler . . . . .	17
4.2.2 Minimierung der Wandlerfehler . . . . .	18

4.3	Zeitdiskretisierung der Signale . . . . .	23
4.3.1	Mittelung der Eingangsgröße des Reglers . . . . .	23
4.3.2	Bestimmung der Diskretisierungsschrittweite . . . . .	26
4.4	Zusammenfassung des Signalfusses . . . . .	31
4.4.1	RT Box zu Mikrocontroller . . . . .	31
4.4.2	Mikrocontroller zu RT Box . . . . .	32
<b>5</b>	<b>Hardware-in-the-Loop-Simulation</b>	<b>33</b>
5.1	Szenario 1 - Hardware-in-the-Loop-Simulation des Referenzmodells . . . . .	34
5.2	Szenario 2 - Latenzen in der Signalübertragung . . . . .	37
5.2.1	Latenzen im Millisekundenbereich . . . . .	38
5.2.2	Latenzen im Sekundenbereich . . . . .	43
5.3	Szenario 3 - Reglermodell mit Frequenzschätzung . . . . .	49
5.4	Diskussion der Ergebnisse . . . . .	56
5.4.1	Ergebnisse Szenario 1 . . . . .	56
5.4.2	Ergebnisse Szenario 2 . . . . .	57
5.4.3	Ergebnisse Szenario 3 . . . . .	58
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>59</b>
6.1	Fazit . . . . .	60
6.2	Ausblick . . . . .	61
	<b>Literaturverzeichnis</b>	<b>62</b>
<b>A</b>	<b>Anhang</b>	<b>67</b>
A.1	PLECS-Schaltbilder des Modells für Szenario 1 & 2 . . . . .	67
A.2	PLECS-Schaltbilder des Modells für Szenario 3 . . . . .	68
	<b>Glossar</b>	<b>70</b>
	<b>Selbstständigkeitserklärung</b>	<b>71</b>

# Abbildungsverzeichnis

2.1	Schema einer Batteriespeicherregelung für virtuelle Trägheit [17]	5
2.2	Model-based Design Workflow [6]	8
3.1	Schaubild des Regelkreises des Referenzmodells	11
3.2	Blockdiagramm des SEBIR-Reglers [15]	12
4.1	Aufbau der HiL-Implementierung	14
4.2	Aufbau der RT Box 1 mit einem Mikrocontroller und dem LaunchPad Interface [30]	16
4.3	Test der Wandlungsfehler des 12 Bit ADC	18
4.4	Messwerte des ADC Tests	19
4.5	Test der Wandlungsfehler des 12 Bit DAC	20
4.6	Messwerte des DAC Tests	21
4.7	<i>PLECS</i> -Implementierung der Skalierung und Kalibrierung mit den berechneten Koeffizienten für die Übertragung des Frequenzwertes	22
4.8	<i>PLECS</i> -Implementierung der Skalierung und Kalibrierung durch die berechneten Koeffizienten für die Übertragung des Wirkleistungs-Sollwerts	22
4.9	Signalverarbeitung des Reglers ohne Mittelung bei einer konstanten Eingangsgröße von 50 Hz. Die Ausgangsgröße ist oben dargestellt, da der Effekt auf diese im Fokus ist	24
4.10	Signalverarbeitung des Reglers mit gleitender Mittelung der Eingangsgröße mit einer Mittelung über 100 Werte, bei einer konstanten Eingangsgröße von 50 Hz. Die Ausgangsgröße ist oben dargestellt, da der Effekt auf diese im Fokus ist	25
4.11	Gezeigt sind die Ergebnisse der SiL-Tests mit unterschiedlichen Zeitschritten (siehe Tab. 4.1) im Vergleich zur MiL-Simulation	27
4.12	Gezeigt sind die Ergebnisse des HiL-Tests im Vergleich zur MiL-Simulation. Zeitschritte des HiL-Tests: $T_{a,RT} = 10 \mu s$ und $T_{a,\mu C} = 100 \mu s$	29
4.13	Signalfluss von der RT Box zum Mikrocontroller	31

---

4.14	Signalfluss von dem Mikrocontroller zu der RT Box . . . . .	32
5.1	Szenario 1 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei hoher Netzträgheit . . . . .	35
5.2	Szenario 1 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei geringer Netzträgheit . . . . .	36
5.3	Schaubild zur Verdeutlichung des Latenzszenarios . . . . .	38
5.4	Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei hoher Netzträgheit und $P_{\text{BESS}} = 70 \text{ W}$ sowie Latenzen im Millisekundenbereich . . . . .	39
5.5	Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei hoher Netzträgheit und $P_{\text{BESS}} = 350 \text{ W}$ sowie Latenzen im Millisekundenbereich . . . . .	40
5.6	Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei geringer Netzträgheit und $P_{\text{BESS}} = 70 \text{ W}$ sowie Latenzen im Millisekundenbereich . . . . .	41
5.7	Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei geringer Netzträgheit und $P_{\text{BESS}} = 350 \text{ W}$ sowie Latenzen im Millisekundenbereich . . . . .	42
5.8	Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei hoher Netzträgheit und $P_{\text{BESS}} = 70 \text{ W}$ sowie Latenzen im Sekundenbereich . . . . .	44
5.9	Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei hoher Netzträgheit und $P_{\text{BESS}} = 350 \text{ W}$ sowie Latenzen im Sekundenbereich . . . . .	45
5.10	Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei geringer Netzträgheit und $P_{\text{BESS}} = 70 \text{ W}$ sowie Latenzen im Sekundenbereich . . . . .	46
5.11	Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei geringer Netzträgheit und $P_{\text{BESS}} = 350 \text{ W}$ sowie Latenzen im Sekundenbereich . . . . .	47
5.12	Szenario 3 - Anpassung der HiL-Implementierung . . . . .	49
5.13	Szenario 3 - Gezeigt werden drei aufgenommene Messreihen von der un-geglätteten, geschätzten Frequenz der PLL nach Übertragung einer dreiphasigen Testspannung mit konstanter Frequenz von 50 Hz. Bei $T_{a,\mu C} = 100 \mu\text{s}$ wird mit dem <i>PLECS</i> -Oszilloskop jedes fünfte Sample abgetastet . . . . .	50

5.14	Szenario 3 - Anpassung der Trägheitskonstante $H_{\text{syn}}$ . . . . .	51
5.15	Szenario 3 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei hoher Netzträgheit im Vergleich zu Szenario 1 . . . . .	53
5.16	Szenario 3 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei geringer Netzträgheit im Vergleich zu Szenario 1 . . . . .	54
A.1	Szenario 1 & 2 - Gezeigt sind die beiden Subsysteme der Modelle des Reglermodells (Mikrocontroller) und des Netzmodells (RT Box) . . . . .	67
A.2	Szenario 1 & 2 - Gezeigt ist das Reglermodell (Mikrocontroller) . . . . .	67
A.3	Szenario 1 & 2 - Gezeigt ist das Netzmodell (RT Box) . . . . .	68
A.4	Szenario 3 - Gezeigt sind die beiden Subsysteme der Modelle des Reglermodells (Mikrocontroller) und des Netzmodells (RT Box) . . . . .	68
A.5	Szenario 3 - Gezeigt ist das Reglermodell (Mikrocontroller) . . . . .	69
A.6	Szenario 3 - Gezeigt ist das Netzmodell (RT Box) . . . . .	69



# Tabellenverzeichnis

3.1	Kennzahlen des Referenzmodells . . . . .	13
4.1	Übersicht der getesteten Zeitschritte im SiL-Test . . . . .	26
4.2	Übersicht der benutzten Zeitschritte sowie CPU-Last der RT Box und des Mikrocontrollers während der HiL-Tests für Szenario 1 & 2 . . . . .	30
5.1	Parameterraum der HiL-Simulationen . . . . .	33
5.2	Szenario 1 - Ergebnisse bei hoher Netzträgheit . . . . .	37
5.3	Szenario 1 - Ergebnisse bei geringer Netzträgheit . . . . .	37
5.4	Szenario 2 - Ergebnisse mit Latenzen im Millisekundenbereich bei hoher Netzträgheit . . . . .	43
5.5	Szenario 2 - Ergebnisse mit Latenzen im Millisekundenbereich bei geringer Netzträgheit . . . . .	43
5.6	Szenario 2 - Ergebnisse mit Latenzen im Sekundenbereich bei hoher Netzträgheit . . . . .	48
5.7	Szenario 2 - Ergebnisse mit Latenzen im Sekundenbereich bei geringer Netzträgheit . . . . .	48
5.8	Übersicht der benutzten Zeitschritte sowie CPU-Last der RT Box und des Mikrocontrollers während der HiL-Tests für Szenario 3 . . . . .	52
5.9	Szenario 3 - Ergebnisse bei hoher Netzträgheit im Vergleich zu Szenario 1	55
5.10	Szenario 3 - Ergebnisse bei geringer Netzträgheit im Vergleich zu Szenario 1	55

# Abkürzungen

**ADC** Analog-to-Digital Converter.

**DAC** Digital-to-Analog Converter.

**EMT** Electromagnetic Transients.

**FPGA** Field Programmable Gate Array.

**HIL** Hardware-in-the-Loop.

**MBD** Model Based Design.

**MIL** Model-in-the-Loop.

**PIL** Processor-in-the-Loop.

**PLL** Phase-locked loop.

**RMS** Root Mean Square.

**RoCoF** Rate of Change of Frequency.

**SEBIR** Swing equation-based inertial response.

**SIL** Software-in-the-Loop.

# 1 Einleitung

## 1.1 Motivation und Hintergrund

Der Einfluss einer reduzierten Trägheit auf die Frequenzstabilität eines Energienetzes kann als die Hauptherausforderung der Netzbetreiber betrachtet werden [37]. Durch die Abschaltung konventioneller Kraftwerke wird dem Energienetz zugleich auch Trägheit entnommen. Die rotierenden Massen dieser Kraftwerke haben einen frequenzstabilisierenden Effekt, welcher Momentanreserve genannt wird. Noch vor der Primärregelung wirkt diese Wirkleistungsungleichgewichten im Netz entgegen. Alternativen zur Erbringung von Netzträgheit durch regenerative Energieerzeugungseinheiten werden gesucht [10].

Die vorliegende Arbeit wurde im Rahmen der Forschungen des Fraunhofer-Instituts für Siliziumtechnologie (ISIT) in dem Projekt NEW 4.0 angefertigt. Sie befasst sich mit der Bereitstellung einer alternativen Momentanreserve mittels eines Batteriespeichers innerhalb eines Netzmodells. Mithilfe einer Regelung, basierend auf der Schwingungsgleichung eines Synchrongenerators, soll ein Batteriespeicher virtuelle Trägheit bereitstellen [12][17].

In verschiedenen Forschungsarbeiten wurde der Reglerentwurf, auch Swing equation-based inertial response (SEBIR) genannt, bereits simulativ untersucht [15][17]. Dabei konnte mit einem Netzmodell gezeigt werden, dass ein Batteriespeicher durch die SEBIR-Regelung virtuelle Trägheit bereitstellen und den Frequenzgradienten bzw. Rate of Change of Frequency (RoCoF) [13] reduzieren kann.

## 1.2 Problemstellung und Zielsetzung

Der Mehrwert dieser Arbeit besteht in der Implementierung des SEBIR-Reglers auf einem Mikrocontroller, sowie der Hardware-in-the-Loop (HIL)-Simulation mit einem Netzmodell, welches auch die Transienten abbildet.

Das vorliegende Netzmodell mit enthaltenem Batteriespeicher und SEBIR-Regelung wird in ein HIL-Modell umgeformt. Mit diesem werden verschiedene Simulationen durchgeführt, welche in drei Szenarien aufgeteilt sind. Durch diese Hardware-Tests soll der nächste Schritt in der Entwicklung des SEBIR-Reglers gegangen werden, um in Zukunft Feldtests durchführen zu können.

Folgende Forschungsfragen werden bearbeitet:

- Welche Herausforderungen sind mit der Implementierung des SEBIR-Reglers auf einem Mikrocontroller verbunden ?
- Hat die SEBIR-Regelung eines Batteriespeichers auch in einer Hardware-in-the-Loop-Simulation, mit Abbildung der transienten Vorgänge, eine frequenzstabilisierende Wirkung ?
- Wie verändert sich der RoCoF und der Frequenztiefpunkt in einer Hardware-in-the-Loop-Simulation bei Variation der Parameter: Batteriespeicherleistung, Netzträgheit und Übertragungslatenz ?

## 1.3 Aufbau der Arbeit

Die restliche Arbeit ist wie folgt aufgebaut:

In **Kapitel 2** werden die angewandten Grundlagen erläutert. Zum einen wird der größere Kontext der Arbeit noch einmal adressiert, zum anderen wird der modellbasierte Entwicklungsprozess und die Echtzeitsimulation vorgestellt.

In **Kapitel 3** wird das gegebene Referenzmodell eines Regelkreises beschrieben sowie die vorgenommenen Anpassungen.

**Kapitel 4** beschreibt den Prozess der HIL-Implementierung. Dabei werden die einzelnen Komponenten des Aufbaus vorgestellt sowie die Verarbeitung und Zeitdiskretisierung der Signale erläutert.

**Kapitel 5** gliedert sich in drei Szenarien, jedes mit unterschiedlichem Fokus für die HIL-Simulation. Die Ergebnisse der Simulationen werden vorgestellt und schließlich diskutiert.

**Kapitel 6** fasst die Erkenntnisse der Arbeit zusammen, zieht ein Fazit und gibt einen Ausblick.

## 2 Grundlagen

### 2.1 Momentanreserve im Energienetz

Die Trägheit des heutigen Energienetzes sinkt, da die Anzahl der Energieerzeuger, welche über Wechselrichter mit dem Netz verbunden sind, steigt. Das Resultat ist ein Energienetz welches sich im Verhalten von dem des herkömmlichen unterscheidet [37].

Die Trägheit eines sich bewegenden physikalischen Objektes bezeichnet dessen Widerstand gegen eine Veränderung seines Bewegungszustands [35]. In einem traditionellen elektrischen Energienetz sind die sich bewegenden Objekte die rotierenden Maschinen (Generatoren und Turbinen), welche an das Stromnetz angeschlossen sind. Der Widerstand gegen Drehzahländerungen befindet sich in dem Trägheitsmoment ihrer rotierenden Massen [37]. In dem ersten Moment einer Frequenzabweichung, als Folge eines Wirkleistungsungleichgewichts im Netz, tauschen diese rotierenden Massen kinetische Energie mit dem Netz aus (geben kinetische Energie ab oder nehmen diese auf) und steuern der Frequenzänderung entgegen [37]. Dadurch haben sie eine stabilisierende Wirkung auf die Netzfrequenz, welche auch als Momentanreserve bezeichnet werden kann [10].

Die Größe, die in diesem Zusammenhang betrachtet wird, ist der RoCoF [13], also der Frequenzgradient. Neue Lösungsansätze für die Bereitstellung von Momentanreserve werden gesucht und benötigt. Die Möglichkeit dafür einen Batteriespeichers zu nutzen, gilt als einer von verschiedenen Ansätzen [10][15].

### 2.2 Schwingungsgleichung

Um das dynamische, frequenz-stabilisierende Verhalten einer Synchronmaschine mit einem Batteriespeicher abzubilden, wird die Schwingungsgleichung einer Synchronmaschine in dessen Frequenzregelung implementiert [12] (siehe Abbildung 2.1). Damit arbeitet

dieser als sogenannte *virtuelle Synchronmaschine* und ahmt das transiente Verhalten einer Synchronmaschine nach [17][15]. Mithilfe von virtueller Trägheit soll die Netzfrequenz stabilisiert, genauer, der RoCoF minimiert werden. Im Folgenden wird die Schwingungsgleichung (2.1) näher beschrieben:

$$\Delta P = 2 \cdot H_{\text{syn}} \cdot f \cdot \frac{df}{dt} \cdot \frac{P_{\text{BESS}}}{f_n^2} \quad (2.1)$$

$\Delta P$  repräsentiert das Leistungsungleichgewicht zwischen mechanischer und elektrischer Leistung eines Synchrongenerators [17]. Im Falle eines Batteriespeichers repräsentiert  $\Delta P$  die Leistung, welche im Zuge einer Frequenzänderung ins Netz eingespeist oder diesem entnommen werden soll. Die Trägheitskonstante  $H_{\text{syn}}$  entspricht dem Wert der virtuellen Trägheit in Sekunden. Da  $H_{\text{syn}}$  im Falle eines Batteriespeichers kein direktes physikalisches Pendant hat, ist theoretisch jeder Wert möglich [17]. Jedoch wirken hier Ladegeschwindigkeit oder eventuelle thermische Effekte der Batterie begrenzend.  $P_{\text{BESS}}$  repräsentiert die Nennleistung des Batteriespeichers,  $f$  entspricht dem Momentanwert der Frequenz und  $f_n$  der Nennfrequenz.

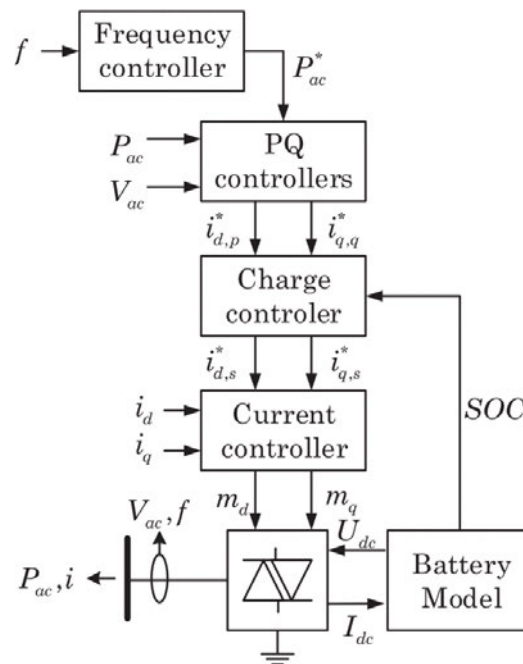


Abbildung 2.1: Schema einer Batteriespeicherregelung für virtuelle Trägheit [17]

## 2.3 Echtzeitsimulation transienter Vorgänge

Um transiente Vorgänge in Energienetzen simulativ zu analysieren, wird ein entsprechendes Modell benötigt. Im Gegensatz zu gemittelten Root Mean Square (RMS)-Modellen, welche mit den Effektivwerten der Ströme und Spannungen rechnen, wird in Electromagnetic Transients (EMT)-Modellen mit den Momentanwerten Größen gerechnet. Basierend auf Differentialgleichungen können mit diesen Modellen kleinere Zeitschritte abbildet werden, als es mit gemittelten Modellen möglich wäre [4].

Simulationsprogramme für EMT-Modelle können in zwei Kategorien aufgeteilt werden: Offline und Real-time (Echtzeit) [22]. Der Vorteil von Echtzeit-Modellen ist, dass sie einem festen Zeitschritt folgen und mit der realen Zeit synchronisiert sind. Das Ziel eines solchen Modells ist, daraus ausführbaren Programmcode zu generieren, welcher anschließend auf einer externen Hardware, sogenannten Echtzeitsimulatoren, ausgeführt werden kann [22].

Echtzeitsimulatoren besitzen Kommunikationsschnittstellen und können Signale mit anderen physischen Geräten austauschen [32]. So kann auf dem Echtzeitsimulator beispielsweise ein Motormodell ausgeführt werden und die Motoregelung auf einem Mikrocontroller implementiert sein. Durch eine Kopplung der beiden physischen Geräte kann ein Regelungsalgorithmus auf externer Hardware gegen ein abstrahiertes Modell der realen Strecke laufen und mit realer Kommunikation getestet werden.

Mithilfe eines Echtzeitsimulators kann ein Regelungsalgorithmus schon während des Entwicklungsprozesses auf geeigneter Hardware ausgeführt und somit Fehler schneller erkannt werden. Kleine Änderungen sind ebenfalls mit geringem Testaufwand verbunden [22][23].

## 2.4 Modellbasierte Entwicklung

Modellbasierte Entwicklung, Model Based Design (MBD) oder Model Driven Software Development (modellgetriebene Softwareentwicklung) ist ein Ansatz für einen Entwicklungsprozess von eingebettetem Programmcode, auf Basis von Softwaremodellen und wird unter anderem in der Automobilindustrie häufig verwendet [7]. MBD hat zum Ziel, die Implementierung von Programmcode für eingebettete Systeme zeit- und kosteneffizienter



zu gestalten [34]. Die Grundidee: Anstatt handgeschriebenen Code z.B. für einen eingebetteten Temperaturregler zu entwickeln, wird mit geeigneten Bausteinen die Regelung grafisch modelliert, um sich aus diesem Modell Code generieren zu lassen [18]. Stufenweise wird so die Verifizierung einer eingebetteten Regelung für ein bestimmtes reelles Zielsystem durchlaufen [9].

Der essentielle Bestandteil von MBD ist die oben erwähnte automatische Codegenerierung. Softwaretools wie *Matlab/Simulink* von *MathWorks* oder *PLECS* von *Plexim*, bieten diese Funktion und so kann der MBD-Prozess durchlaufen werden. Mit der Kombination aus automatischer Codegenerierung und einem Echtzeitsimulator können mit geringem Aufwand Echtzeitsimulationen, wie HIL-Simulationen, durchgeführt werden [6] (siehe Kapitel 2.4.4). So wird der Entwicklungsprozess beschleunigt, da ein Regelungsalgorithmus schon in der Praxis erprobt wird, bevor eine physikalischer Teststand zur Verfügung steht.

Ein Vorteil der Tests mit Echtzeitsimulatoren ist, dass während der Ausführung Parameter geändert und Signale in Echtzeit beobachtet werden können. Änderungen, die mit dem realen System nicht möglich oder sehr aufwändig wären, können hier also getestet werden, wie z.B. eine Änderung des Drehmoments einer Turbine [6]. Außerdem kann auch ein Regler während der Simulation feinjustiert werden.

Die MBD-Prozess basiert auf einem Verfahren namens "V-Diagramm" (Abb. 2.2). Vier grundlegende Schritte sind notwendig in diesem Verfahren [6]:

- Aufbau des Modells für die Repräsentation des physischen Systems, welches geregelt, gesteuert oder überwacht werden soll [9] (Plant-Modell)
- Analyse des Plant-Modells und Aufbau des Modells für die Repräsentation der Software des eingebetteten Systems [9] (Controller-Modell)
- Simulation des Plant-Modells und Controller-Modells in Interaktion
- Einsatz der Regelungssoftware

Auf der folgenden Seite werden die einzelnen Stufen des MBD-Prozess beschrieben.

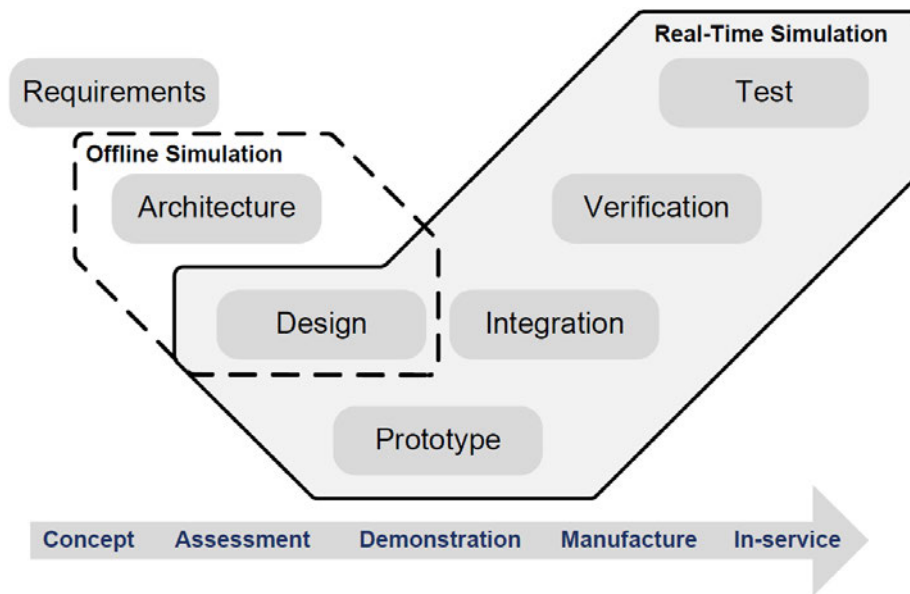


Abbildung 2.2: Model-based Design Workflow [6]

Nachdem die Anforderungen der Regelung ermittelt worden sind, kann modelliert werden. Grundlegend sind die vier beschriebenen Stufen, es können jedoch, je nach Anwendung, noch weitere hinzukommen [21].

### 2.4.1 Model-in-the-Loop

Der erste Stufe lautet Model-in-the-Loop (MIL). Hier werden Plant-Modell und Controller-Modell innerhalb einer Simulationumgebung, mithilfe von verschiedenen zur Verfügung stehenden Bausteinen, aufgebaut. Das Ziel dieser Stufe ist, das Plant-Modell von dem Controller-Modell regeln/steuern zu lassen [24]. In diesem Schritt wird somit die Logik und das Verhalten des Controller-Modells verifiziert (Architecture & Design, Abb. 2.2).

### 2.4.2 Software-in-the-Loop

Software-in-the-Loop (SIL) ist die zweite Stufe. Aus den Software-Modellen wird echtzeitfähiger Code generiert und in Interaktion simuliert. Plant- und Controller-Modell werden so in einer quasistetigen Simulation getestet und folgen einem festen Zeitschritt. Die Zeitschritte werden variiert, um einen geeigneten zu finden, welcher das gewünschte

Reglerverhalten abbildet. So deckt diese Stufe auf, inwiefern der generierte Code das Verhalten des Modells widerspiegelt [34][24] (Design, Abb. 2.2).

### **2.4.3 Processor-in-the-Loop**

Die nächste Stufe lautet Processor-in-the-Loop (PIL). Hier wird der Code des Controller-Modells auf einem eingebetteten Prozessor ausgeführt. Das Plant-Modell wird weiter in der Simulationsumgebung ausgeführt. Bei dieser Stufe wird die Lauffähigkeit des Codes auf dem Prozessor analysiert. Bei ungewünschtem Verhalten werden bei den vorherigen Stufen Anpassungen vorgenommen [34][24] (Prototype, Abb. 2.2).

### **2.4.4 Hardware-in-the-Loop**

HIL-Simulationen werden durchgeführt, bevor der eingebettete Controller mit der realen Hardware verbunden wird. Das Entscheidende dabei ist, dass das Plant-Modell auf einem Echtzeitsimulator ausgeführt und auch die tatsächliche Kommunikation zwischen Plant und Controller getestet wird. Der Echtzeitsimulator emuliert das Plant-Modell und eine Regelungshardware das Controller-Modell [6][19] (Integration bis Test, Abb. 2.2).

## 3 Referenzmodell

Die Grundlage dieser Arbeit bildet ein gegebenes EMT-Modell eines Regelkreises, modelliert mit dem Simulationsplattform *PLECS*. Die Dimensionierung der Komponenten sowie die zu übertragenen Bänder werden angepasst. Dieses Modell kann als MIL im MBD-Prozess angesehen werden.

Die Strecke des Regelkreises ist das Modell eines Inselnetzes, bestehend aus einem Synchrongenerator, einem vereinfachten Batteriespeicher (ideale Stromquelle) mit abgebildetem PQ- und Frequenzregler (siehe Abb. 2.1) und einer Last (siehe Abb. 3.1). Die betrachtete Regler-Komponente des Regelkreises ist der Frequenzregler des Batteriespeichers, in welchem die Schwingungsgleichung implementiert ist (siehe Abb. 2.1).

Das Eingangssignal des Frequenzreglers ist die Netzfrequenz  $f$  und das Ausgangssignal ist die Sollwertvorgabe für den Batteriespeicher  $P_{\text{soll}}$  (siehe Abb. 3.1). Im Hinblick auf zukünftige Feldtest, z.B. in Brunsbüttel [3] und Curslack [26] aus dem NEW 4.0 Projekt oder der Batteriespeicher in dem Inselnetz der *Universitat Politècnica de València* [1], soll die Regelung des Batteriespeichers über eine Sollwertvorgabe erfolgen.  $P_{\text{soll}}$  wird an den PQ-Regler des Batteriespeichers übermittelt, welcher in diesem Regelkreis Teil der Strecke ist (siehe Abb. 3.1).

### 3.1 Streckenmodell des Inselnetzes

Das Netzmodell wurde in Anlehnung an das Inselnetz der *Universitat Politècnica de València* dimensioniert [1]. Es entspricht einem Netz im niedrigen Kilowattbereich. In der Simulation soll ein Störsprung in der Last simuliert werden. Der Störsprung soll bei geringer Netzträgheit zu einem Fall der Frequenz von etwa -800 mHz führen, welche der maximalen (negativen) dynamischen Frequenzabweichung entspricht [14]. Ferner soll der Störsprung nach dem Einschwingen des Systems auf eine Frequenz von 50 Hz erfolgen, was nach 15,0s der Fall ist. Zuvor wird der Batteriespeicher, nach 13,0s, eingeschaltet.

Die Netzträgheit wird durch Veränderung des Massenträgheitsmoments des Synchrongenerators variiert, da dieser die einzige Komponente des Inselnetzes ist, welche zur Netzträgheit beiträgt. Die Höhe der Netzträgheit wird in der Tabelle 3.1 als Netzanlaufzeitkonstante  $T_A$  in Sekunden angegeben [16]. Es wurden zwei Fälle angenommen, jeweils für eine geringe ( $T_A = 7,0\text{ s}$ ) und eine hohe Netzträgheit ( $T_A = 13,6\text{ s}$ ), orientiert an Schätzungen aus [37]. Da der Synchrongenerator die frequenz-bildende Komponente im System ist, wird der zu übertragende Frequenzwert  $f$  aus dessen Drehzahl berechnet.

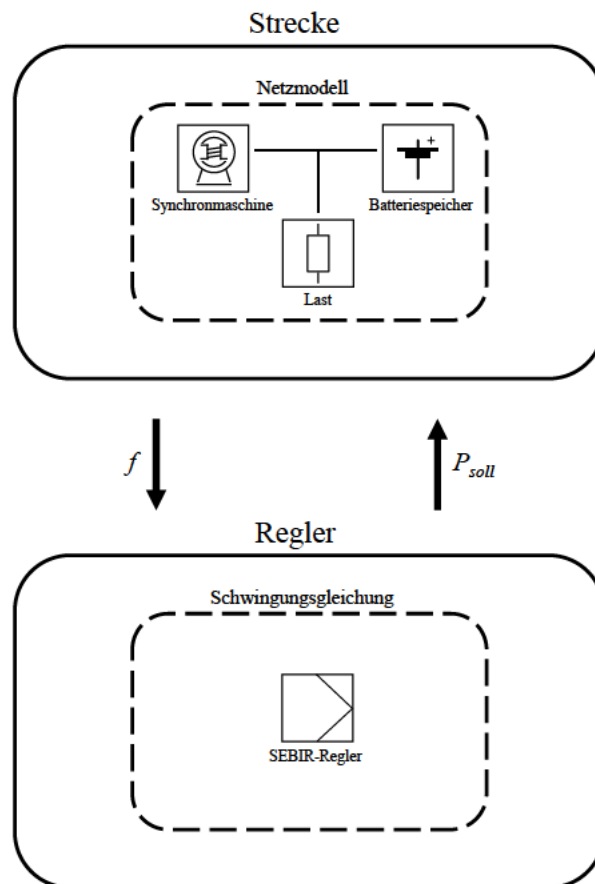


Abbildung 3.1: Schaubild des Regelkreises des Referenzmodells

### 3.2 Reglermodell mit Schwingungsgleichung

Der SEBIR-Regler soll für das Band der dynamischen Frequenzabweichung von 49,2 Hz bis 50,8 Hz ausgelegt sein [14]. Der Störsprung bei geringer Netzträgheit soll für eine maximale Leistungseinspeisung des Batteriespeichers sorgen. Hierbei muss das Verhältnis zwischen Gesamtlast des Netzes und Batteriespeichers sinnvoll dimensioniert sein, d.h. kein verhältnismäßig großer Speicher. Ansonsten würde der Batteriespeicher zu einem Zusammenbruch des Netzes führen. Ebenso muss die Trägheitskonstante des Reglers  $H_{\text{syn}}$  für die jeweilige Größe des Batteriespeichers speziell eingestellt werden.

Der Batteriespeicher wird für diese Arbeit so dimensioniert, dass dieser jeweils etwa 1% und 5% der Gesamtleistung des Netzes vor dem Störsprung entspricht ( $P_{\text{BESS}}$ ). Außerdem wird eine Verzögerung von 20 ms vor dem Regler platziert ( $e^{sT_D}$ ), stellvertretend für die Latenz eines Gerätes für eine Frequenzschätzung. Die **20 ms** entsprechen einer Periode eines **50 Hz** Signals. Die Frequenzschätzung eines idealen, verzerrungsfreien Spannungssignals kann mithilfe einer rekursiven diskreten Fourier-Transformation innerhalb einer Periode erfolgen [8] [25]. Ein Tiefpassfilter wird nach einem numerischen Differenzierer des Reglers ( $d/dt$ -Block), welcher den Frequenzgradient berechnet, platziert, um hochfrequente Anteile des Signal herauszufiltern ( $1/(1 + s \cdot T_R)$ ). Der Frequenzgradient wird mit einer Samplezeit von 10 ms berechnet. Die erwähnten Parameter stammen aus dem Blockdiagramm des SEBIR-Reglers (vgl. Abb. 3.2).

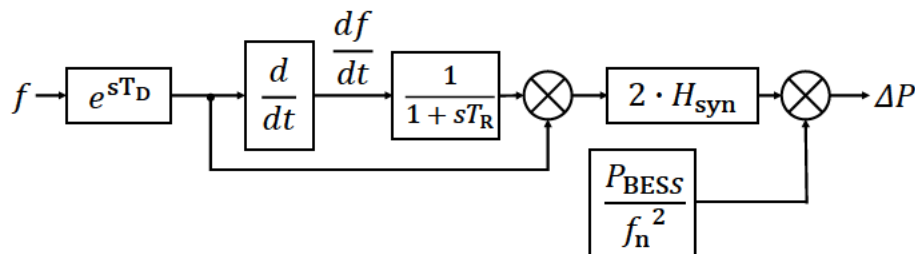


Abbildung 3.2: Blockdiagramm des SEBIR-Reglers [15]

### 3.3 Kennzahlen des Referenz-Szenarios

In der folgenden Tabelle werden alle relevanten Kennzahlen des Referenzmodells dargestellt. Für  $H_{\text{syn}}$ ,  $T_A$  und  $P_{\text{BESS}}$  werden jeweils die zwei Varianten angegeben, die in den späteren HIL-Simulationen verglichen werden.

Tabelle 3.1: Kennzahlen des Referenzmodells

Parameter	Wert	Einheit
Nennfrequenz $f_n$	50	Hz
Latenz Frequenzschätzung $T_D$	0,02	s
Samplezeit des numerischen Differenzierers $\Delta t$	0,01	s
Zeitkonstante des Tiefpassfilters $T_R$	0,01	s
Trägheitskonstante $H_{\text{syn}}$	15,0 ( $P_{\text{BESS}} = 70 \text{ W}$ ) oder 17,0 ( $P_{\text{BESS}} = 350 \text{ W}$ )	s
Nenndrehmoment Synchrongenerator $M_n$	46,488	Nm
Störsprung der Last (LSP)	7,2 auf 8,2	kW
Netzanlaufzeitkonstante $T_A$	7,0 (6,0 nach LSP) oder 13,6 (12,0 nach LSP)	s
Leistung Batteriespeicher $P_{\text{BESS}}$	70 oder 350	W

## 4 Hardware-in-the-Loop-Implementierung

Wie in Kapitel 2.4.4 beschrieben, wird für eine HiL-Simulation ein Echtzeitsimulator und eine Regelungshardware benötigt. In diesem Kapitel werden die benutzten Hardwares für die Implementierung sowie die Kommunikation über deren Schnittstellen beschrieben. Außerdem werden die Fehler, welche durch die Wandlung auftreten und die Möglichkeit der Fehlerminimierung bzw. -kompensation, beschrieben. Zum Schluss wird erläutert, wie beim Übergang von MIL über SIL eine geeignete Diskretisierungsschrittweite für die HiL-Implementierung gefunden wird.

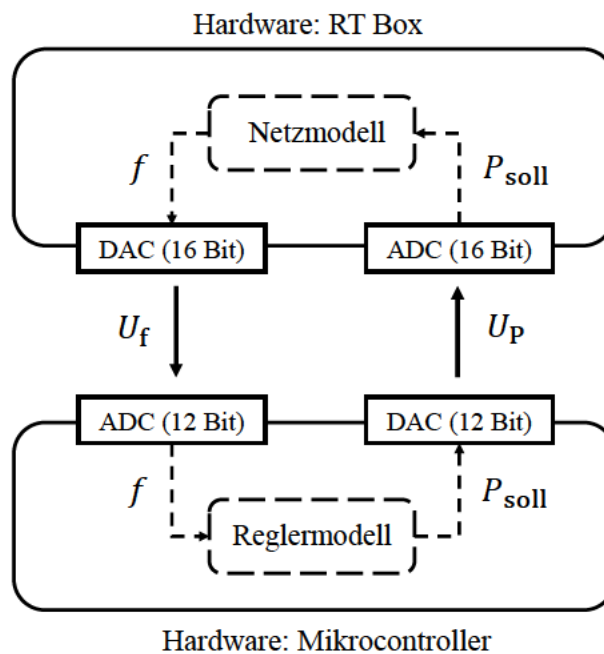


Abbildung 4.1: Aufbau der HiL-Implementierung



Der grundlegende Aufbau (siehe Abb. 4.1): der berechnete Frequenzwert  $f$  aus dem Netzmodell, wird auf einen passenden Spannungsbereich skaliert. Anschließend wird die gewandelte Spannung  $U_f$  über den Digital-to-Analog Converter (DAC) der RT Box an einen Analog-to-Digital Converter (ADC) des Mikrocontrollers geschickt und eingelesen. Auf dem Mikrocontroller wird der gewandelte Wert wieder auf den Frequenzwert aus dem Netzmodell  $f$  hochskaliert und im Reglermodell verarbeitet.

Ein Wirkleistungs-Sollwert  $P_{\text{soll}}$  wird berechnet und für die Übertragung an die RT Box passend skaliert. Über einen DAC des Mikrocontrollers wird die skalierte Spannung  $U_P$  an einen ADC der RT Box gesendet (vgl. Abb. 4.1). Der gewandelte Wert des ADC wird auf der RT Box wieder auf  $P_{\text{soll}}$  hochskaliert und an den Batteriespeicher im Netzmodell geleitet, welcher, je nach Sollwertvorgabe, Leistung ins Netz einspeist.

Somit ist das Ziel, dass in dem Augenblick, wenn die Frequenz im Netzmodell fällt, der Batteriespeicher möglichst unverzüglich Leistung ins Netz einspeist und so den Frequenzgradienten bzw. RoCoF abschwächt. Am Ende dieses Kapitels wird der gesamte Signalfluss noch einmal detailliert beschrieben.

## 4.1 Hardwarekomponenten

### 4.1.1 Echtzeitsimulator

Als Echtzeitsimulator dient die *RT Box 1* der Firma *Plexim* [32], weshalb auch das Referenzmodell mit der dazugehörigen Software *PLECS* aufgebaut wurde. Die *RT Box 1* besitzt einen Xilinx Zynq System-on-Chip. Dieser enthält sowohl einen Field Programmable Gate Array (FPGA) als auch mehrere CPU-Kerne [32]. 16 analoge Ein- und Ausgänge, mit je einer Auflösung von 16 Bit, befinden sich auf der Vorderseite der *RT Box 1* sowie 32 digitale Ein- und Ausgänge.

Die RT Box wurde speziell für Anwendungen in der Leistungselektronik entwickelt. Sie kann als Emulator eines Systems oder als Regler eingesetzt werden. Typischerweise wird in einem HiL-Test der Leistungsteil eines leistungselektronischen Systems mit der RT Box emuliert. PWM-Signale können von den digitalen Eingängen der RT Box mit einer zeitlichen Auflösung von weniger als 10 ns eingelesen werden, so kann eine Regelung, welche PWM-Signale zum Ansteuern von z.B. IGBTs erzeugt, mit sehr geringer Latenz getestet werden [32].

### 4.1.2 Regelungshardware

*Plexim* bietet mit dem *RT Box LaunchPad Interface* eine Plug-and-Play-Lösung für Mikrocontroller der Firma *Texas Instruments*. Da diese Schnittstellen-Karte zur Verfügung steht, wird ein Mikrocontroller der Firma *Texas Instruments* genutzt, genauer der *LAUNCHXL-F280049C*. Dieser ist Teil einer speziellen Prozessor-Familie, die besonders für Echtzeitanwendungen geeignet ist [36]. Der Controller besitzt eine 32-Bit CPU mit einer Taktrate von 100 MHz. Außerdem relevant für diese Arbeit sind: Zwei 12 Bit DAC Ausgänge, drei 12 Bit ADC, sowie 100 KB RAM und 256 KB Flash Speicher (genauere Informationen können dem Datenblatt entnommen werden [36]). Die Abbildung 4.2 zeigt den HiL-Teststand fertig zusammengesteckt:



Abbildung 4.2: Aufbau der RT Box 1 mit einem Mikrocontroller und dem LaunchPad Interface [30]

Der Mikrocontroller wird in dieser Arbeit mithilfe von *PLECS* und *PLECS Coder* direkt programmiert, dabei wird der RAM-Speicher des Mikrocontrollers benutzt. Es kann auch alternativ ein Code für das Programm *Code Composer Studio* generiert werden, dass von *Texas Instruments* zur Verfügung gestellt wird.

## 4.2 Signalübertragung und -verarbeitung

Bevor verschiedene HIL-Simulationen durchgeführt werden können, muss zunächst die MIL-Simulation modifiziert und die Kommunikations-Bausteine für die Signalübertragung konfiguriert werden. Das Regler- und Netzmodell wird jeweils mit Bausteinen für

die ADC und DAC der Geräte erweitert. Es muss jeweils der richtige ADC des Mikrocontrollers mit dem richtigen DAC der RT Box verbunden werden und andersherum. Eine Übersicht über die Pin-Belegung des *RT Box LaunchPad Interface*, bei verschiedenen Mikrocontrollern, bietet das dazugehörige Datenblatt [29].

#### 4.2.1 Skalierung der Signalwandler

Wie im Kapitel 3.2 beschrieben, soll der Regler ein Frequenzband von 49,2 Hz bis 50,8 Hz verarbeiten können. Die ADC und DAC-Bausteine müssen jeweils auf das Frequenzband und den passenden Spannungsbereich skaliert werden. Der jeweilige Frequenzwert wird von der RT Box, in Form einer Spannung, an den Mikrocontroller gesendet.

Da der Spannungsbereich des ADC des Mikrocontrollers auf 0 V bis 3,0 V beschränkt ist und der mögliche Bereich der RT Box etwas größer ist, liegt die Beschränkung auf Seite des Mikrocontrollers. Die Skalierung der 49,2 Hz bis 50,8 Hz auf 0 V bis 3,0 V wird berechnet und man erhält folgende Umrechnung:

$$U_f = m_{\text{ADC}} \cdot f + b_{\text{ADC}} = 1,875 \text{ Hz V}^{-1} \cdot f - 92,25 \text{ V} \quad (4.1)$$

Da der Regler für eine maximale Leistungseinspeisung des Batteriespeichers ausgelegt wird, ist das zu übertragende Band  $-P_{\text{BESS}}$  bis  $+P_{\text{BESS}}$  (siehe Kapitel 3.2). Die Idee ist hier, den Anteil der Maximalleistung zu übertragen mit  $\pm 1,0$  für  $\pm 100 \%$  der Leistung des Batteriespeichers. Der DAC bietet einen Spannungsbereich von 0 V bis 2 V (siehe Kapitel 4.2.2). Berechnet man die nötige Skalierung von  $P_{\text{soll}}$  zu  $U_P$  erhält man:

$$U_P = m_{\text{DAC}} \cdot P_{\text{soll}} + b_{\text{DAC}} = \frac{P_{\text{soll}}}{P_{\text{BESS}}} \cdot 1 \text{ V} + 1 \text{ V} \quad (4.2)$$

Die Skalierung des ADC wird für das Szenario 3 angepasst, da dort nicht der Frequenzwert übertragen wird, sondern die dreiphasige Netzspannung. Die Skalierung des DAC bleibt gleich. Nach dem gleichen Prinzip wird auch hier die Skalierung berechnet:

$$U_{3\text{ph},\mu\text{C}} = m_{\text{ADC}} \cdot U_{3\text{ph}} + b_{\text{ADC}} = \frac{3}{650} \text{ V} \cdot U_{3\text{ph}} + 1,5 \text{ V} \quad (4.3)$$

Nachdem der skalierte Wert an die jeweils andere Hardware übertragen wurde, wird die Skalierung dort wieder rückgängig gemacht, dazu muss die Umrechnung umgekehrt werden (siehe Abb. 4.7 & 4.8)

### 4.2.2 Minimierung der Wandlerfehler

Der benutzte Mikrocontroller weist einige Wandlerfehler auf. In diesem Unterkapitel sollen die Fehler des ADC und des DAC des Mikrocontrollers sowie die Minimierung dieser beschrieben werden. Da die Wandler der RT Box eine hohe 16 Bit Auflösung besitzen, werden diese Wandlungsfehler in dieser Arbeit nicht betrachtet.

#### Wandlungsfehler des ADC

Die Referenzspannung des ADC vom Mikrocontroller wird durch die Codegenerierung von PLECS mit der internen 3,3 V Spannung konfiguriert. Wenn man für den ADC Gain Error (Internal reference), Offset Error sowie Linearitäts-Fehler aufaddiert, wird ein maximaler Fehler von 53 LSB möglich. Bezogen auf 12 Bit (4096 LSB) sind dies 1,29 % (siehe Datenblatt [36]). Aus unbekanntem Gründen konnte trotz einer Referenzspannung von 3,3 V nur ein Spannungsbereich von 0 V bis 3,0 V abgedeckt werden.

Um die Wandlungsfehler zu testen, werden verschiedene Testwerte  $U_{\text{ideal,ADC}}$  ohne Skalierung von der RT Box an den DAC des Mikrocontrollers gesendet und dort gemessen  $U_{\text{mess,ADC}}$  (siehe Abb. 4.3).

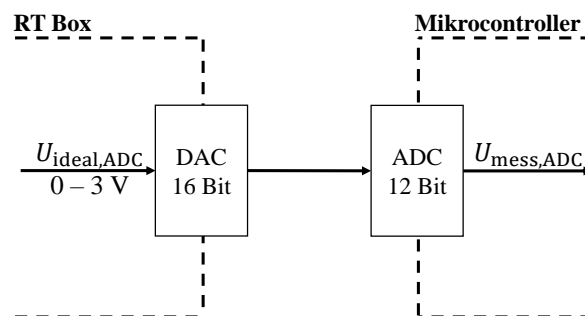


Abbildung 4.3: Test der Wandlungsfehler des 12 Bit ADC

Idealerweise sind  $U_{\text{ideal,ADC}}$  und  $U_{\text{mess,ADC}}$  identisch. Für verschiedene Spannungswerte von 0 V bis 3,0 V wurden jeweils der Idealwert  $U_{\text{ideal,ADC}}$  und der Messwert  $U_{\text{mess,ADC}}$  aufgenommen (siehe Abb. 4.4).

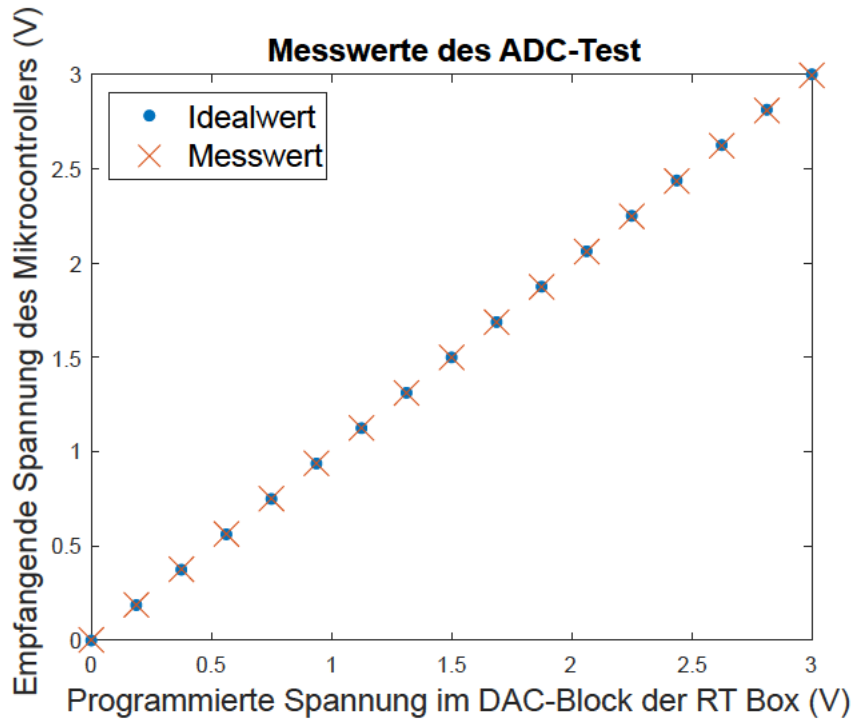


Abbildung 4.4: Messwerte des ADC Tests

Zur Minimierung des ADC-Fehlers wurde aus den Messwerten und mithilfe eines Online-Tools zur Linearen Regression [5] eine Regressionsfunktion berechnet (4.4).

$$U_{\text{ideal,ADC}} = m_{\text{ADC,Err}} \cdot U_{\text{mess,ADC}} + b_{\text{ADC,Err}} = (1,0025 \cdot U_{\text{mess,ADC}}) - 0,00313\text{V} \quad (4.4)$$

### Wandlungsfehler des DAC

Für den DAC-Baustein wird durch die *PLECS* Codegenerierung die interne 2,5 Volt Referenzspannung des Mikrocontrollers konfiguriert. Auch bei dem DAC treten ein Offset-, Gain- und Linearitäts-Fehler auf (siehe Datenblatt [36]). Aus dem Datenblatt geht ebenso hervor, dass bei einer maximalen Vorgabe von 4096 LSB, lediglich etwa 3500

LSB, also 85,6 %, tatsächlich vom DAC ausgegeben werden [36]. Bei einer Referenzspannung von 2,5 V entsprechen 3500 LSB etwa 2,14 V. Der DAC des Mikrocontrollers wurde deshalb in dieser Arbeit nur im Bereich von 0 V bis 2,0 V genutzt.

Um auch die Wandlungsfehler des DAC des Mikrocontrollers zu testen, werden dem DAC-Baustein in *PLECS* Werte zwischen 0 V bis 2,0 V vorgegeben, die gewandelte Spannung an die RT Box gesendet und dort mit dem 16 Bit ADC eingelesen und gemessen (siehe Abb. 4.5). Dadurch sollen der gesendete Wert des Mikrocontrollers  $U_{\text{ideal,DAC}}$  und der auf der RT Box gemessene  $U_{\text{mess,DAC}}$  verglichen werden.

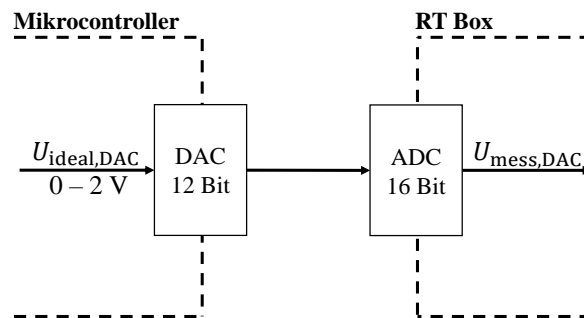


Abbildung 4.5: Test der Wandlungsfehler des 12 Bit DAC

Wie erwartet, wird ein maximaler Spannungswert von etwa 2,14 V ausgegeben. Für die Signalübertragung wird mit dem Spannungsbereich 0 V bis 2,0 V gearbeitet. Die Messwerte des DAC-Tests sind in der Abbildung 4.6 zu sehen.

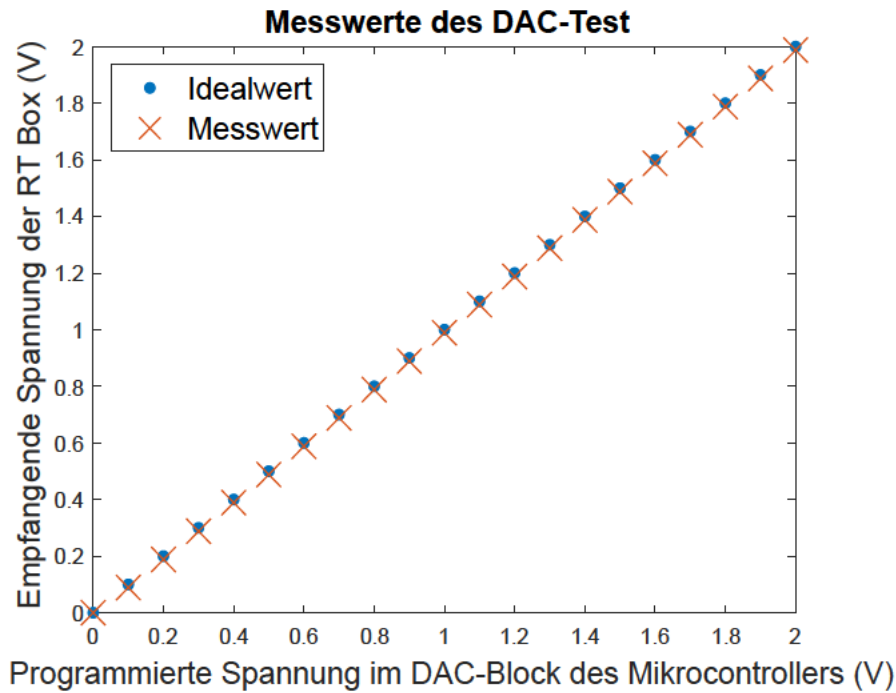


Abbildung 4.6: Messwerte des DAC Tests

Aus den aufgenommenen Werten konnte nun eine Regressionsfunktion für die Fehlerminimierung des Wandlungsfehlers des DAC berechnet werden ( 4.5).

$$U_{\text{ideal,DAC}} = m_{\text{DAC,Err}} \cdot U_{\text{mess,DAC}} + b_{\text{DAC,Err}} = (1,00038 \cdot U_{\text{mess,DAC}}) + 0,01136\text{V} \quad (4.5)$$

### Implementierung der Fehlerminimierung und Skalierung mit *PLECS*

Die Skalierung der DAC und ADC geschieht innerhalb der *PLECS*-Bausteine durch die Eingabe der Koeffizienten einer linearen Funktion nach der Logik: *Scale · Input + Offset* bzw.  $mx + b$ . Da jedoch auch die Wandlerfehler ausgeglichen werden sollen, wird lediglich der DAC-Block mit der Skalierung parametrisiert. Die ADC sollen die gewandelte Spannung zunächst ohne Änderung an eine Kalibrierung (zur Fehlerminimierung) weiterleiten und danach wird der Wert dann hochskaliert. In den folgenden Abbildungen soll deutlich werden, wie die berechneten Koeffizienten aus (4.1), (4.2), (4.4) und (4.5) innerhalb von

*PLECS* implementiert werden (siehe Abb. 4.7 und 4.8). In gleicher Weise wird auch (4.3) für Szenario 3 in *PLECS* eingetragen.

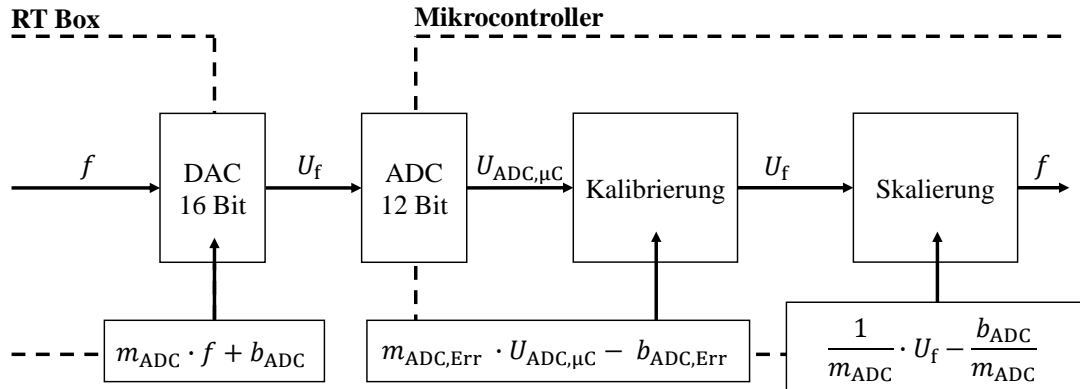


Abbildung 4.7: *PLECS*-Implementierung der Skalierung und Kalibrierung mit den berechneten Koeffizienten für die Übertragung des Frequenzwertes

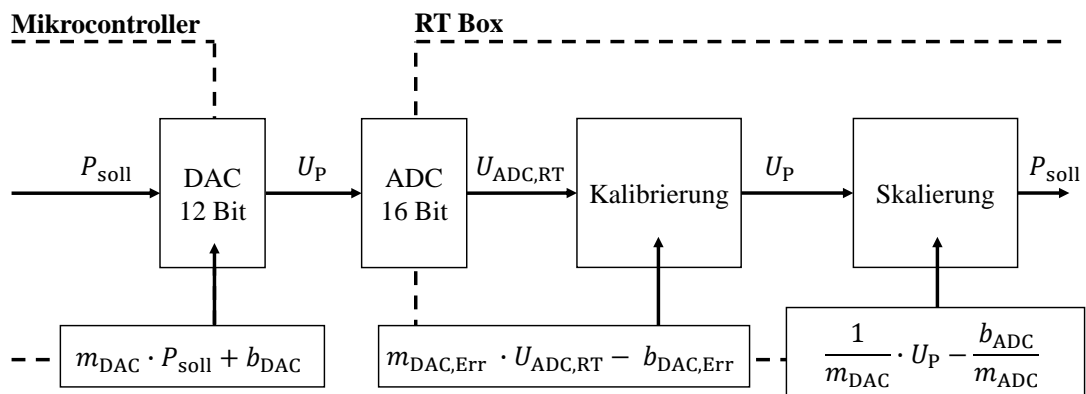


Abbildung 4.8: *PLECS*-Implementierung der Skalierung und Kalibrierung durch die berechneten Koeffizienten für die Übertragung des Wirkleistungs-Sollwerts



## 4.3 Zeitdiskretisierung der Signale

### 4.3.1 Mittelung der Eingangsgröße des Reglers

Nach der Implementierung der korrekten Skalierung und Fehlerminimierung/Kalibrierung kann der Signalfluss des Aufbaus in einem ersten HiL-Test getestet werden. Hierzu wird von der Strecke (RT Box) ein konstanter Wert von 50 Hz vorgegeben, um eine konstante Stellgröße zu erhalten. Der Regler ist bei diesem Test für einen Batteriespeicher von  $P_{\text{BESS}} = 350 \text{ W}$  mit einer Trägheitskonstante von 17,0s ausgelegt (siehe Kapitel 3.3). Es werden 6 Messreihen aufgenommen, je drei mit und drei ohne Mittelung der gewandelten Spannung  $U_{\text{ADC},\mu\text{C}}$ . Dadurch soll der Effekt einer Mittelung getestet werden. Das Reglermodell wird für diesen Test mit einer Schrittweite von 100  $\mu\text{s}$  gerechnet. Im Oszilloskop in *PLECS* wird nur jedes zehnte Sample abgetastet, deshalb ist ein Zeitschritt von 1 ms in den Abbildungen zu erkennen (siehe Abb. 4.9 und 4.10).

Es wird ein Bit-Rauschen des ADC, beim Einlesen der übertragenen Spannung, festgestellt. Da der Regler auf Frequenzänderungen reagieren soll, sorgt der differenzierende Algorithmus der Schwingungsgleichung für eine Verstärkung dieses Rauschens. Nach Kalibrierung und Skalierung der Spannung  $U_{\text{ADC},\mu\text{C}}$  besitzt der skalierte Frequenzwert  $f$  ein Rauschen in der vierten Nachkommastelle (siehe Abb. 4.9). Dieses Rauschen der Eingangsgröße bewirkt ein verstärktes Rauschen der Ausgangsgröße des Reglers (Wirkleistungs-Sollwert) von etwa  $\pm 10 \text{ W}$  (siehe Abb. 4.9). Der erwartete Wert ist jedoch 0 W, da ein konstanter Wert von 50 Hz vorgegeben wird.

Um die Abweichung vom erwarteten Wert zu minimieren, wird der Frequenzwert nach dem Einlesen des ADC gemittelt. Durch eine gleitende Mittelung über 100 Werte kann das Rauschen minimiert werden. Mit Mittelung beträgt die maximale Abweichung nun  $\pm 2$  Watt (siehe Abb. 4.10). Je höher jedoch die Leistung des Batteriespeicher  $P_{\text{BESS}}$  ist, desto höher ist auch die Verstärkung des Rauschens (siehe Abb. 3.2).

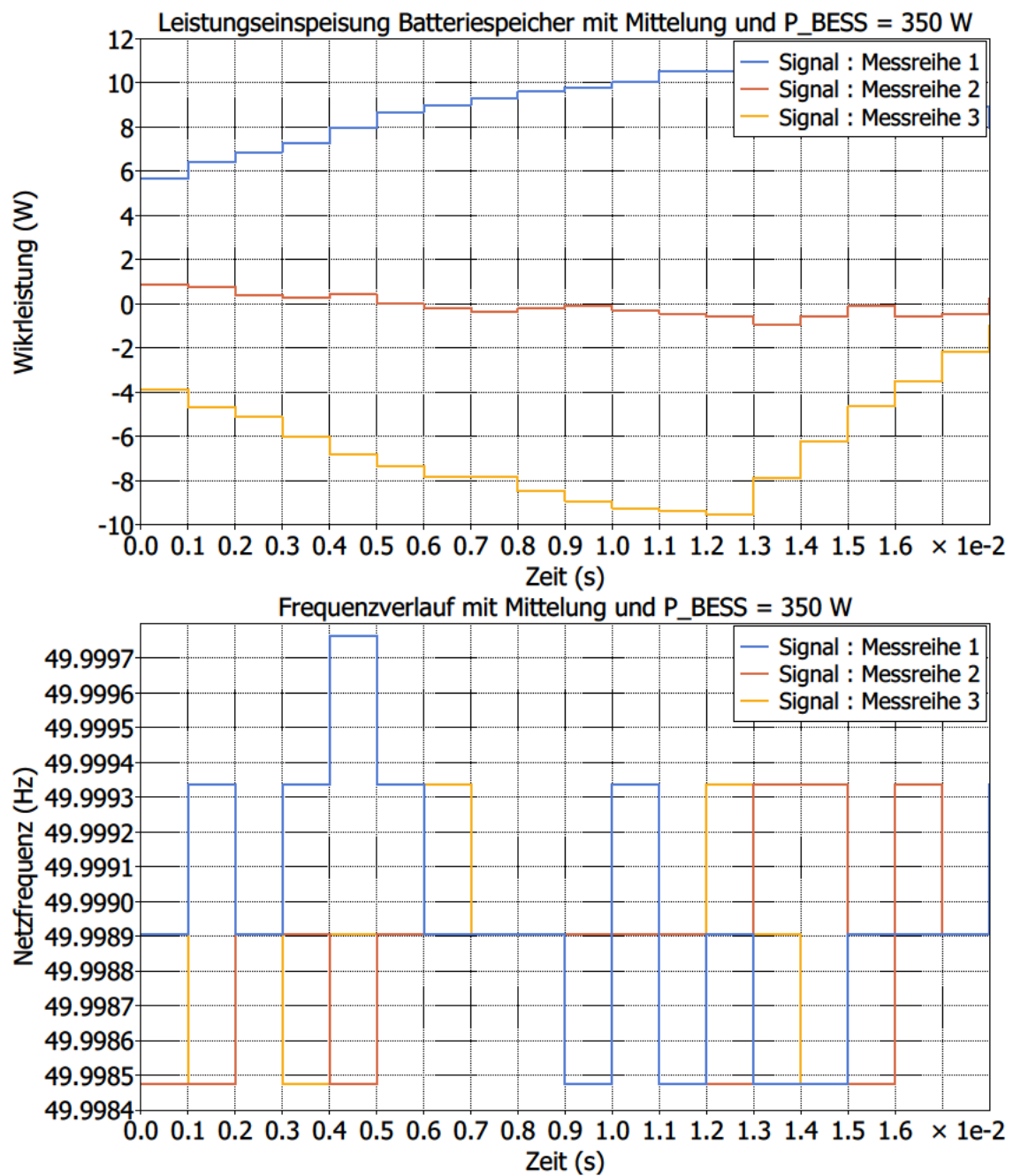


Abbildung 4.9: Signalverarbeitung des Reglers ohne Mittelung bei einer konstanten Eingangsgröße von 50 Hz. Die Ausgangsgröße ist oben dargestellt, da der Effekt auf diese im Fokus ist

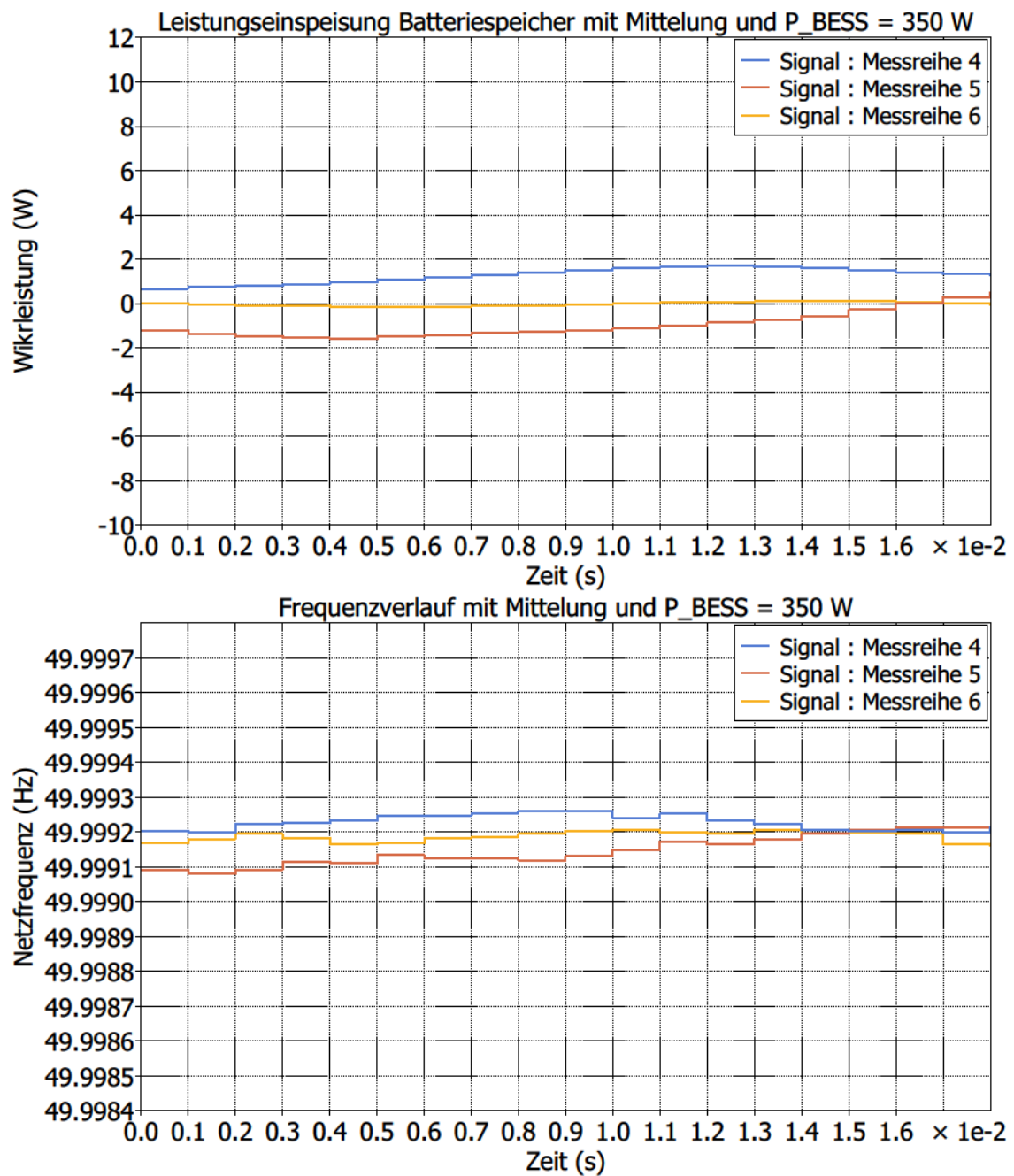


Abbildung 4.10: Signalverarbeitung des Reglers mit gleitender Mittelung der Eingangsgröße mit einer Mittelung über 100 Werte, bei einer konstanten Eingangsgröße von 50 Hz. Die Ausgangsgröße ist oben dargestellt, da der Effekt auf diese im Fokus ist

### 4.3.2 Bestimmung der Diskretisierungsschrittweite

Damit Netz- und Reglermodell auf externer Hardware ausgeführt werden können, müssen diese diskretisiert werden. Durch den *PLECS Coder* werden die Gleichungen des zeitkontinuierlichen Zustandsraumes in Gleichungen des zeitdiskreten Zustandsraums transformiert [31]. *PLECS* bietet für den Test einer geeigneten Schrittweite den sogenannten *CodeGen-Modus*. Dabei aus den Modellen generischer C-Code generiert, welcher die Modelle im diskreten Zustandsraum repräsentiert. Dieser *CodeGen-Modus* ist zu vergleichen mit der SiL-Simulation im Model-Based-Design-Prozess bzw. einer quasistetigen Simulation. In diesem Unterkapitel soll das Vorgehen beschrieben werden, um einen geeigneten Zeitschritt zur Ausführung auf der Hardware zu finden.

Es wird ein SiL-Test und im Anschluss ein HiL-Test durchgeführt. In beiden Tests wird ein Störgrößensprung (Störsprung in der Last) simuliert und die Ergebnisse des kontinuierlichen Modells (MiL), mit denen des diskreten Modells (SiL oder HiL) verglichen. In dem SiL-Test können verschiedene Diskretisierungsschrittweiten ohne externe Hardware in der Simulationsumgebung getestet werden.

Wenn eine Schrittweite zu einer guten Übereinstimmung mit den Ergebnissen der Referenzsimulation (MiL-Simulation) führt, kann in einem HiL-Test überprüft werden, ob Speicherplatz und Rechenleistung der genutzten Hardware ausreichend sind. Wenn nicht, muss ein etwas kleinerer Zeitschritt gewählt, das Modell vereinfacht oder zur Not andere Hardware genutzt werden. In den *Coder Options* von *PLECS* wird die Diskretisierungsmethode *Radau* gewählt und *double* als Fließkomma-Format eingestellt. Folgende Zeitschritte werden getestet:

Tabelle 4.1: Übersicht der getesteten Zeitschritte im SiL-Test

Test	$T_{a,Plant}$	$T_{a,Ctrl}$
1	20 $\mu$ s	10 ms
2	10 $\mu$ s	10 ms
3	10 $\mu$ s	1 ms
4	10 $\mu$ s	100 $\mu$ s

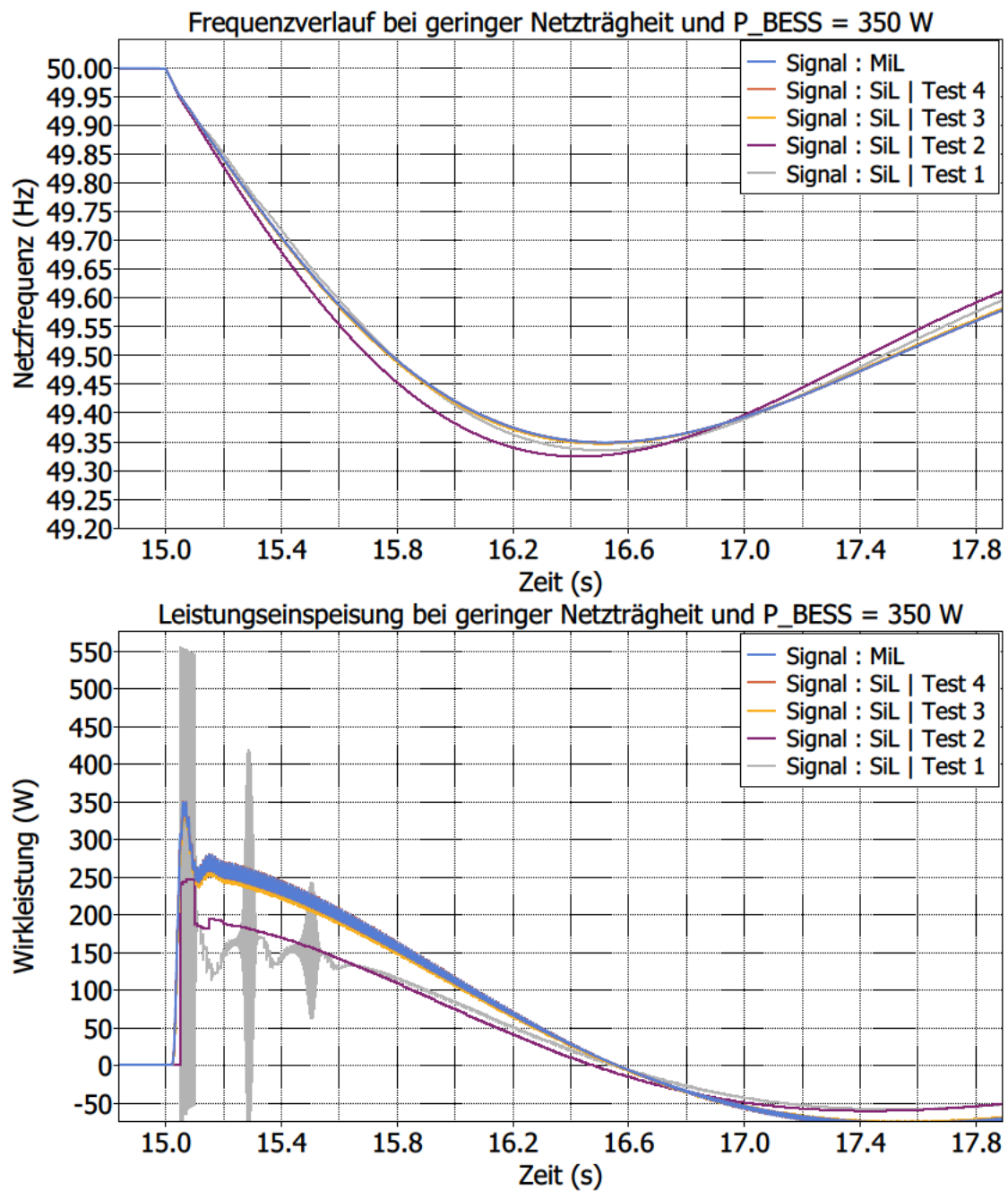


Abbildung 4.11: Gezeigt sind die Ergebnisse der SiL-Tests mit unterschiedlichen Zeitschritten (siehe Tab. 4.1) im Vergleich zur MiL-Simulation

Abbildung 4.11 zeigt die Ergebnisse der SiL-Tests. Vier verschiedene Kombinationen werden getestet, wobei unterschiedliche Zeitschritte für das Reglermodell ( $T_{a, \text{Ctrl}}$ ) und das Netzmodell ( $T_{a, \text{Plant}}$ ) getestet werden (siehe Tab. 4.1). Das Ziel ist hierbei einen Zeitschritt zu finden, mit welchen die Ergebnisse der MiL-Simulation treu abgebildet werden. Für die Tests gelten Demo-Modelle von *PLECS* als Orientierung.

Zunächst werden zwei Zeitschritte  $T_{a, \text{Plant}} = 20 \mu\text{s}$  und  $T_{a, \text{Plant}} = 10 \mu\text{s}$  für das Netzmodell verglichen, bei einem konstanten Zeitschritt des Reglermodells von  $T_{a, \text{Ctrl}} = 10 \text{ms}$ . Die Ergebnisse mit  $T_{a, \text{Plant}} = 20 \mu\text{s}$  sorgen für ein sehr unerwünschtes Verhalten der Leistungseinspeisung des Reglers. Mit  $T_{a, \text{Plant}} = 10 \mu\text{s}$  ist der Verlauf deutlich näher am MiL-Ergebnis, doch auch hier ist Optimierungspotential zu erkennen (siehe Abb. 4.11).

Der Zeitschritt des Reglermodells wird deshalb bei einem konstantem Zeitschritt des Netzmodells von  $T_{a, \text{Plant}} = 10 \mu\text{s}$  variiert. Ein Zeitschritt von  $T_{a, \text{Ctrl}} = 100 \mu\text{s}$  führt zu der besten Übereinstimmung mit den Ergebnissen aus der MiL-Simulation, weshalb die dazugehörige Kurve in der Abbildung kaum zu erkennen ist (siehe Abb. 4.11). So werden die Ergebnisse aus Test 4, Diskretisierungsschrittweiten von  $T_{a, \text{Plant}} = 10 \mu\text{s}$  und  $T_{a, \text{Ctrl}} = 100 \mu\text{s}$ , anschließend im HiL-Test geprüft.

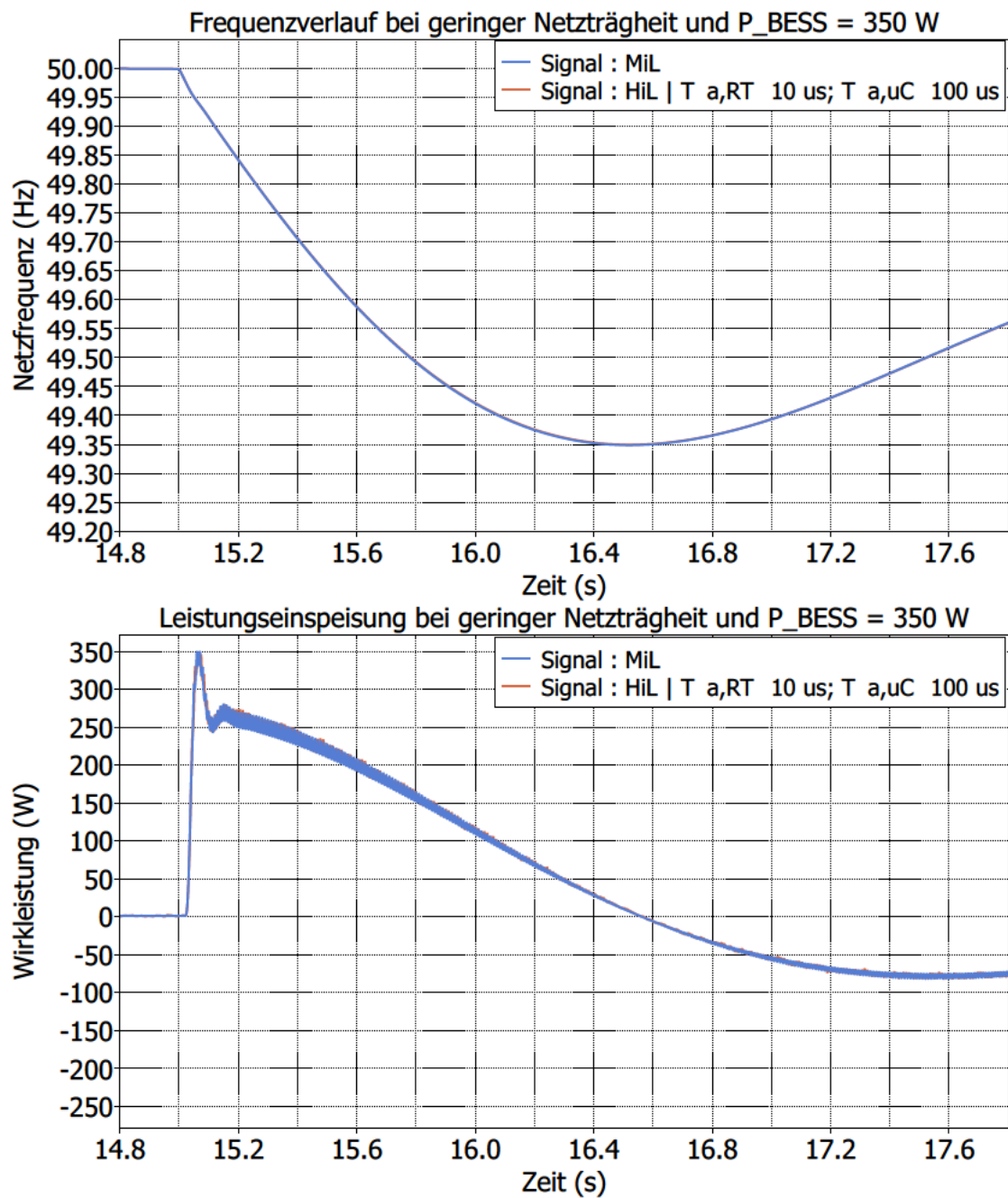


Abbildung 4.12: Gezeigt sind die Ergebnisse des HiL-Tests im Vergleich zur MiL-Simulation. Zeitschritte des HiL-Tests:  $T_{a,RT} = 10 \mu\text{s}$  und  $T_{a,\mu\text{C}} = 100 \mu\text{s}$

Tabelle 4.2: Übersicht der benutzten Zeitschritte sowie CPU-Last der RT Box und des Mikrocontrollers während der HiL-Tests für Szenario 1 & 2

$T_{a,RT}$	$CPU_{RT}$	$T_{a,\mu C}$	$CPU_{\mu C}$
10 $\mu s$	$\sim 66\%$	100 $\mu s$	$\sim 50\%$

Die Ergebnisse des HiL-Tests (siehe Abb. 4.12) zeigen, dass die Zeitschritte aus SiL-Test 4 auch hier zur gewünschten Übereinstimmung führen und der Speicherplatz und die Rechenleistung der Hardware ausreichen (siehe Tab. 4.2). Deshalb werden die Diskretisierungsschrittweiten von  $T_{a,RT} = 10 \mu s$  für die RT Box und  $T_{a,\mu C} = 100 \mu s$  für den Mikrocontroller gewählt. Somit wurde die HiL-Implementierung abgeschlossen. Alle durchgeführten HiL-Simulationen in Kapitel 5 werden mit diesen Zeitschritten gerechnet.



## 4.4 Zusammenfassung des Signalflusses

Im Folgenden wird der Signalfluss des Regelkreises noch einmal zusammengefasst. Zunächst der Signalfluss von der RT Box zum Mikrocontroller (① bis ⑦ siehe Abb. 4.13) und im Anschluss von dem Mikrocontroller zu der RT Box. (⑧ bis ⑫ siehe Abb. 4.14)

### 4.4.1 RT Box zu Mikrocontroller

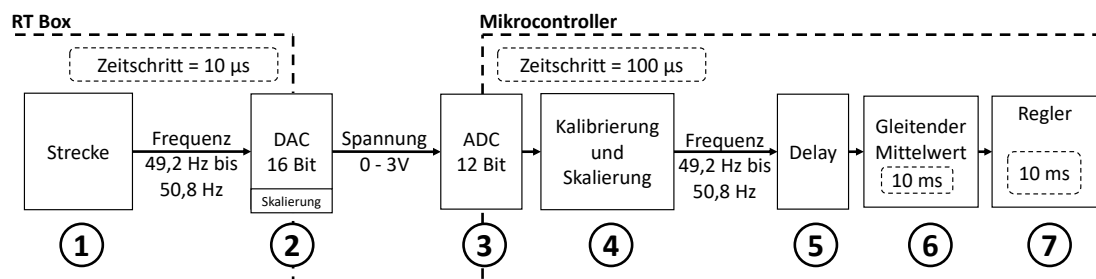


Abbildung 4.13: Signalfluss von der RT Box zum Mikrocontroller

- ① Zuerst wird von der RT Box in dem Netzmodell (Strecke) die Netzfrequenz berechnet. Die RT Box rechnet dieses Modell mit einer Diskretisierungsschrittweite von  $10 \mu\text{s}$ .
- ② In dem Baustein des 16 Bit DAC der RT Box wird das zu übertragende Frequenzband von 49,2 bis 50,8 Hz auf 0 V bis 3,0 V skaliert (4.1). Dies ist der Spannungsbereich des ADC auf der Seite des Mikrocontrollers.
- ③ Über den 12-Bit ADC liest der Mikrocontroller den Spannungswert mit einer Diskretisierungsschrittweite von  $100 \mu\text{s}$  ein.
- ④ Der Spannungswert wird dann kalibriert bzw. der Übertragungsfehler minimiert (4.4) und wieder auf den vorherigen Frequenzbereich hochskaliert (4.1).
- ⑤ Der Delay-Block simuliert die Latenz eines Gerätes für die Frequenzschätzung.
- ⑥ Nach dem Delay erfolgt eine gleitende Mittelung des Signals über 100 Werte. Dies

entspricht bei  $T_{a,RT} = 100 \mu\text{s}$  einer Periode von 10 ms. Damit wird das Rauschen, welches beim Einlesen des ADC auftritt, geglättet.

⑦ Im Anschluss wird der gemittelte Wert an den Regler übergeben.

#### 4.4.2 Mikrocontroller zu RT Box

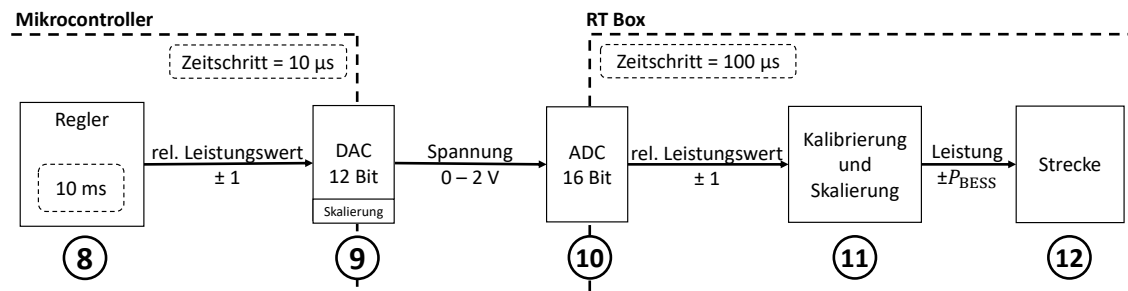


Abbildung 4.14: Signalfluss von dem Mikrocontroller zu der RT Box

⑧ Der Regler berechnet mit einer Samplezeit von 10 ms einen Wirkleistungs-Sollwert zwischen  $-P_{\text{BESS}}$  und  $+P_{\text{BESS}}$ , welcher für die Übertragung auf den relativen Leistungswert zwischen -1 und +1 umgerechnet wird.

⑨ In dem PLECS-Softwarebaustein des 12 Bit DAC des Mikrocontrollers wird das zu übertragende Band von -1 bis +1 (relative Leistung) auf den Spannungsbereich des DAC 0 V bis 2,0 V skaliert (4.2).

⑩ Der übertragene Wert wird auf der RT Box Seite mit dem 16 Bit ADC eingelesen und wieder auf -1 bis 1 gesetzt.

⑪ Der Übertragungsfehler wird durch Kalibrierung minimiert (4.5) und danach durch Skalierung (4.2) auf  $-P_{\text{BESS}}$  bis  $+P_{\text{BESS}}$  gebracht.

⑫ Der skalierte Wert wird dann an die Strecke übergeben, wo sich das Modell des Batteriespeicher befindet. Der Batteriespeicher speist Leistung ins Netz ein und beeinflusst die Frequenz und die Schleife beginnt wieder bei ①.

## 5 Hardware-in-the-Loop-Simulation

In diesem Kapitel werden die Ergebnisse der durchgeführten HiL-Simulationen vorgestellt und diskutiert. Es werden drei Szenarien definiert und immer die Auswirkung des Batteriespeichers mit SEBIR-Regelung auf die Netzfrequenz bei geringer ( $T_A = 7,0\text{ s}$ ) und hoher Netzträgheit ( $T_A = 13,6\text{ s}$ ) untersucht. Auch der Frequenzverlauf bei einem Netz ohne Batteriespeicher wird zum Vergleich mit dargestellt. Für diesen Fall werden die Parameter in den Ergebnistabellen mit einem "-" gekennzeichnet. In jedem Szenario wurden jeweils der RoCoF und der Tiefpunkt der Netzfrequenz  $f_{\min}$  untersucht. Für die RoCoF-Betrachtung wird jeweils die Steigung der Frequenz für die 500 ms ( $\Delta t$ ) unmittelbar nach dem Störgrößensprung berechnet, also  $\Delta f/\Delta t$  [13]. Grundsätzlich werden folgende Kombinationen der Parameter getestet (siehe Tab. 5.1):

Tabelle 5.1: Parameterraum der HiL-Simulationen

$P_{\text{BESS}}$ (W)	$T_A$ (s)
–	7,0
70	7,0
350	7,0
–	13,6
70	13,6
350	13,6

Bei allen Szenarien wird das Netzmodell mit einer Diskretisierungsschrittweite von  $10\text{ }\mu\text{s}$  und das Reglermodell mit  $100\text{ }\mu\text{s}$  gerechnet. Szenario 1 und 2 werden mit der HiL-Implementierung aus Kapitel 4 durchgeführt. Bei Szenario 3 muss die Implementierung angepasst werden, da ein anderes Eingangssignal von der RT Box an den Mikrocontroller übertragen werden soll. Diese und andere spezifische Änderungen sowie Anpassungen der Parametrierung, werden im jeweiligen Unterkapitel der Szenarien beschrieben. Zum

Ende dieses Kapitels werden die Ergebnisse aus den Szenarien diskutiert. Der Anhang zur Arbeit befindet sich auf CD und kann beim Erstgutachter eingesehen werden.

## **5.1 Szenario 1 - Hardware-in-the-Loop-Simulation des Referenzmodells**

In Szenario 1 wird das Referenzmodell aus Kapitel 3, nach erfolgreicher HiL-Implementierung, als HIL-Modell simuliert. Die Simulation wird mit den in Tabelle 5.1 aufgelisteten Kombinationen der Parameter durchgeführt.

Die Abbildung 5.2 zeigt die Signalverläufe der Netzfrequenz und der Leistungseinspeisung des Batteriespeichers bei geringer und die Abbildung 5.1 bei hoher Netzträgheit. Die Ergebnisse für den RoCoF, die prozentuale Verringerung des RoCoF und der Tiefpunkt der Frequenz  $f_{\min}$  sind in den Tabellen 5.2 und 5.3 aufgeführt.

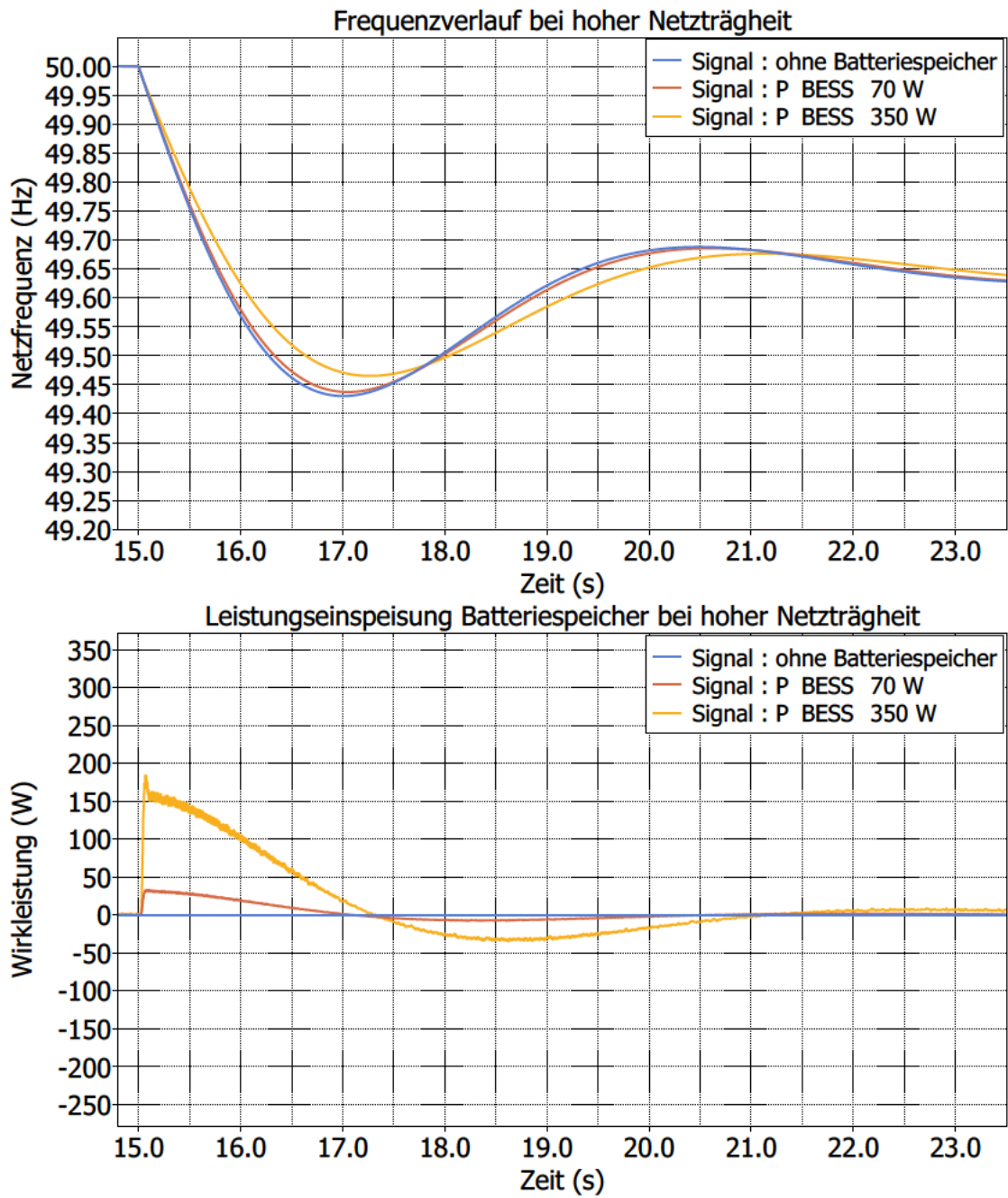


Abbildung 5.1: Szenario 1 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei hoher Netztr agheit

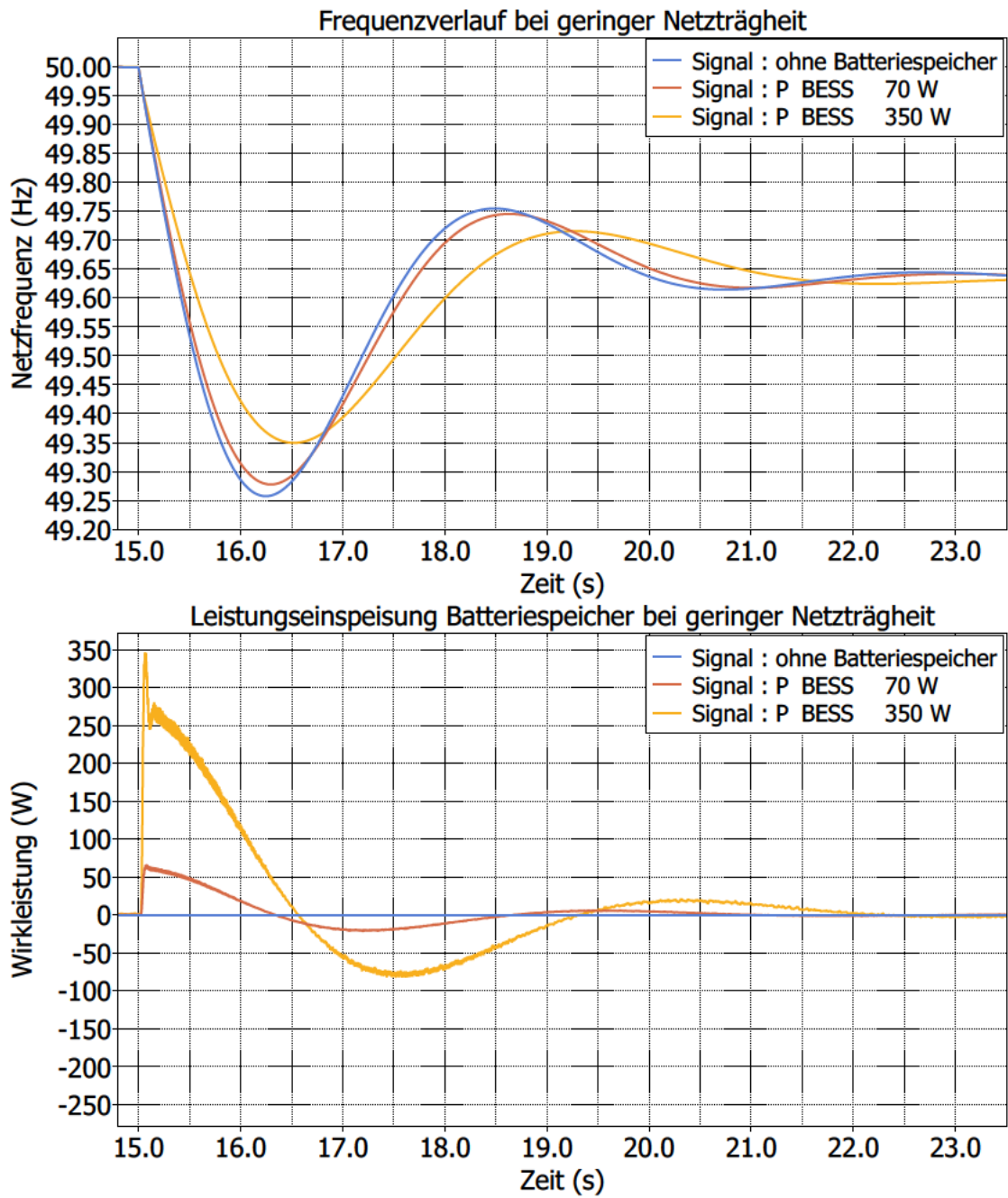


Abbildung 5.2: Szenario 1 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei geringer Netzträgheit

Tabelle 5.2: Szenario 1 - Ergebnisse bei hoher Netztr agheit

$P_{\text{BESS}}$ (W)	RoCoF (Hz s <sup>-1</sup> )	Verringerung RoCoF (%)	$f_{\text{min}}$ (Hz)
–	-0,489	–	49,431
70	-0,475	3	49,437
350	-0,421	14	49,465

Tabelle 5.3: Szenario 1 - Ergebnisse bei geringer Netztr agheit

$P_{\text{BESS}}$ (W)	RoCoF (Hz s <sup>-1</sup> )	Verringerung RoCoF (%)	$f_{\text{min}}$ (Hz)
–	-0,925	–	49,258
70	-0,877	5	49,278
350	-0,708	23	49,350

## 5.2 Szenario 2 - Latenzen in der Signal ubertragung

Bei einer Anwendung f ur virtuelle Tr agheit ist eine latenzarme Kommunikation anzustreben, da herkommliche Momentanreserve per Definition unverz ogert zur Verf ugung steht [10]. In der Realit at werden jedoch aufgrund der Signal ubertragung und der Tatsache, dass die Frequenz erst gesch atzt werden muss, immer Latenzen auftreten. In Szenario 2 soll deshalb der RoCoF und Frequenztiefpunkt, nach einem St orsprung bei einer latenzbehafteten Signal ubertragung zwischen Regler und Strecke, untersucht werden.

Im Netzmodell, auf der RT Box ausgef uhrt, wird jeweils vor dem DAC und nach dem ADC eine Latenz eingef ugt (siehe Abb. 5.3). Zum einen werden Latenzen im Millisekundenbereich und zum anderen im Sekundenbereich untersucht. Bei jeder Simulation wird zum Vergleich der Signalverlauf ohne Batteriespeicher sowie der Signalverlauf mit Batteriespeicher bei latenzfreier Kommunikation (0s) ebenfalls als Vergleich dargestellt. Bei der Angabe der Latenz ist im Folgenden immer die Summe aus beiden Latenz-Bl ocken gemeint (siehe Abb. 5.3).

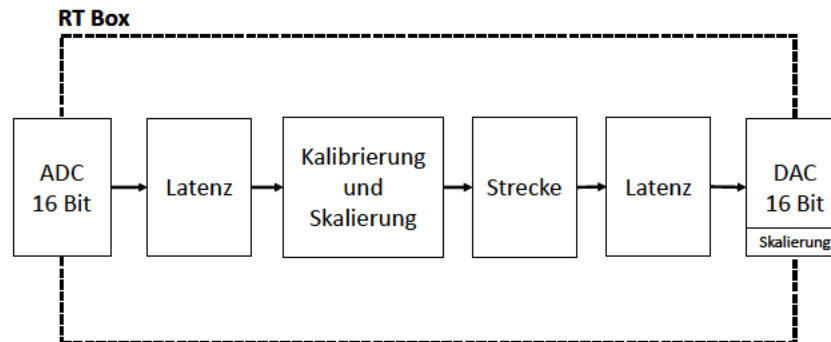


Abbildung 5.3: Schaubild zur Verdeutlichung des Latenzszenarios

### 5.2.1 Latenzen im Millisekundenbereich

Da die Kommunikation mit dem Batteriespeicher bzw. dessen Wechselrichter in Brunsbüttel [3][2] und an der *Universitat Politècnica de València* [28][1] über das Modbus-TCP-Protokoll erfolgt, sollen die Latenzen dieses Protokolls als Bezug für die Dimensionierung der Latenzen im Millisekundenbereich dienen. Aus Gesprächen mit der Firma *opcsa* [27] geht hervor, dass bei einer Kommunikation mit dem Modbus-TCP-Protokoll Latenzen von **100 ms** auftreten können. Eine weitere Quelle bestätigt Übertragungslatenzen in dieser Größenordnung [33]. Deswegen wird **100 ms** als eine realistische Latenz bei einer Kommunikation über Modbus-TCP angenommen.

Aus dem NEW 4.0 Projekt am Batteriespeicher in Curlsack [26] geht hervor, dass die Latenz durch die Kommunikation sehr stark schwankt. Um die Auswirkung einer etwas größeren Latenz auf die Frequenzstabilität zu untersuchen, wird ebenso eine Verzögerung von **300 ms** simuliert. Die folgenden Abbildungen zeigen die Ergebnisse aus den Simulationen mit Latenzen von **100 ms** und **300 ms** bei hoher und geringer Netzträgheit.



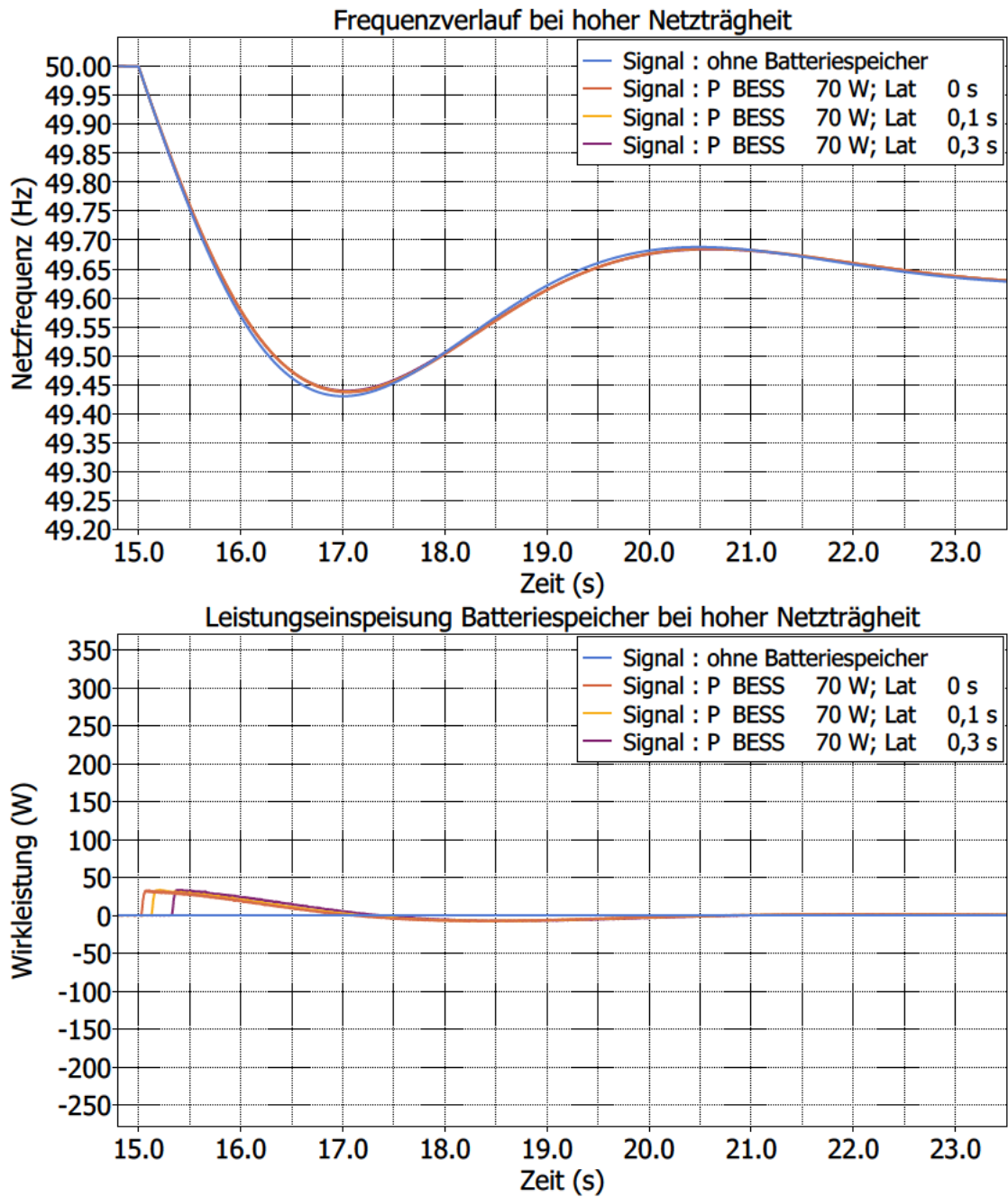


Abbildung 5.4: Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei hoher Netztr agheit und  $P_{\text{BESS}} = 70 \text{ W}$  sowie Latenzen im Millisekundenbereich

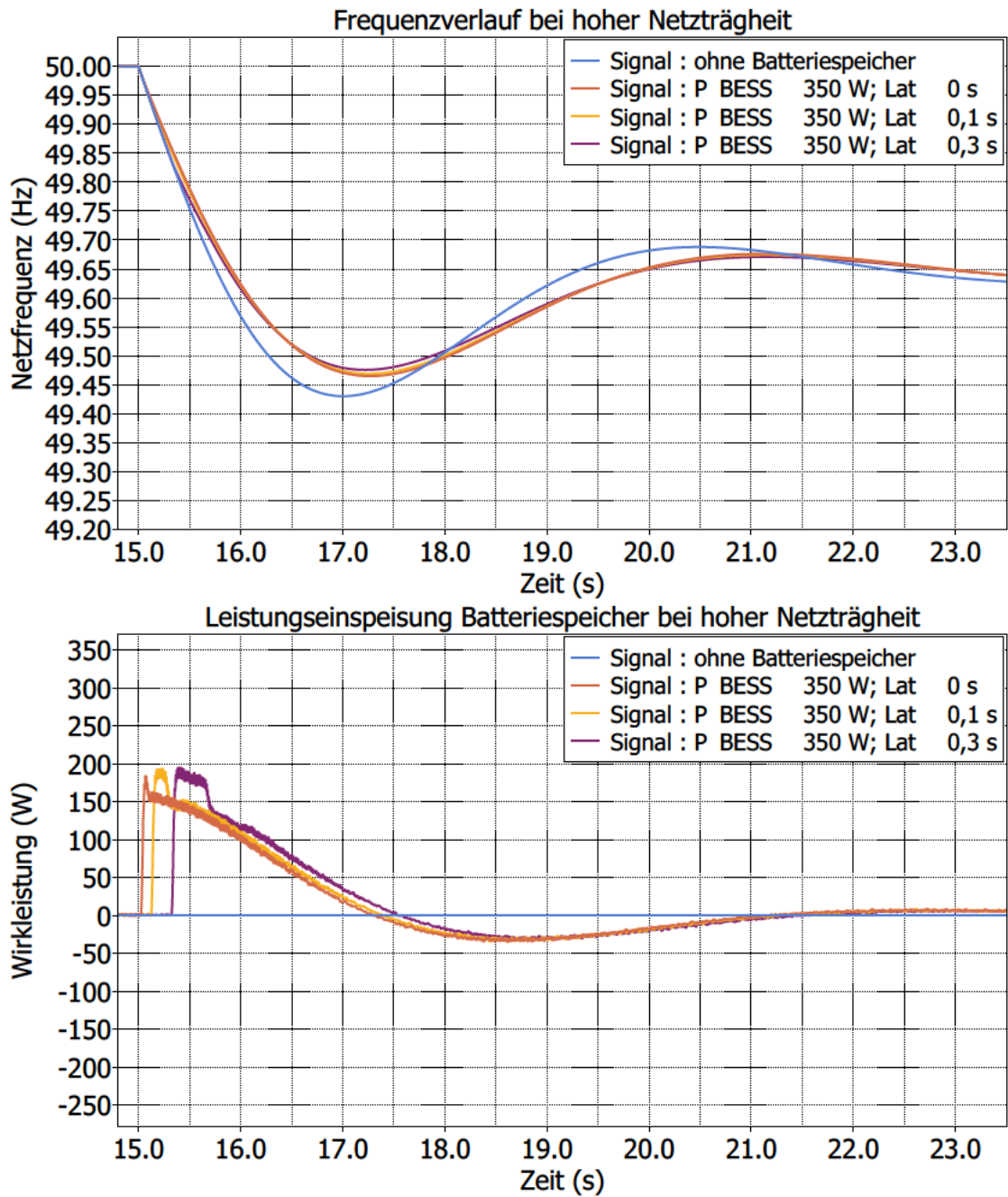


Abbildung 5.5: Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei hoher Netztr agheit und  $P_{\text{BESS}} = 350 \text{ W}$  sowie Latenzen im Millisekundenbereich

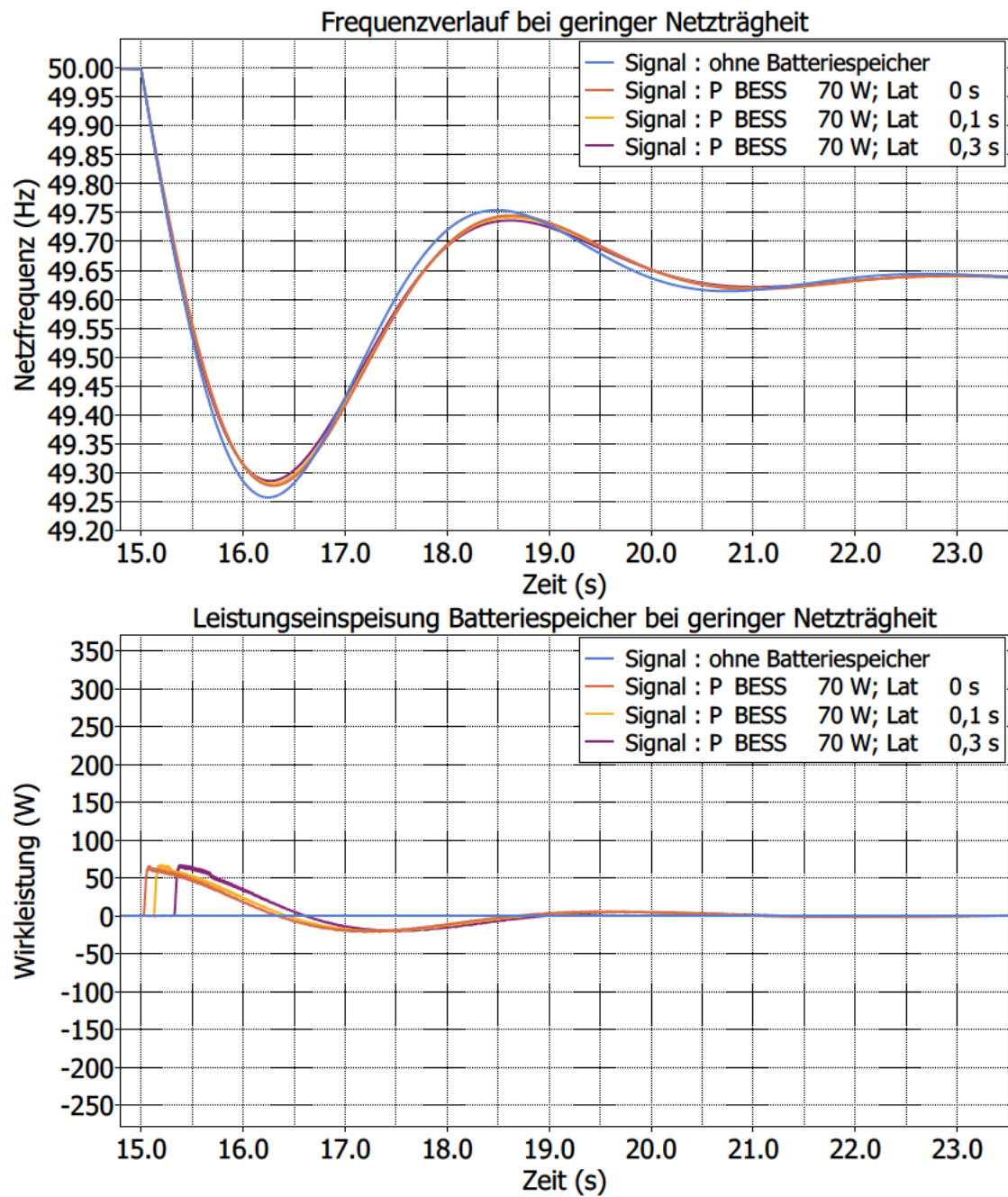


Abbildung 5.6: Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei geringer Netztr agheit und  $P_{\text{BESS}} = 70 \text{ W}$  sowie Latenzen im Millisekundenbereich

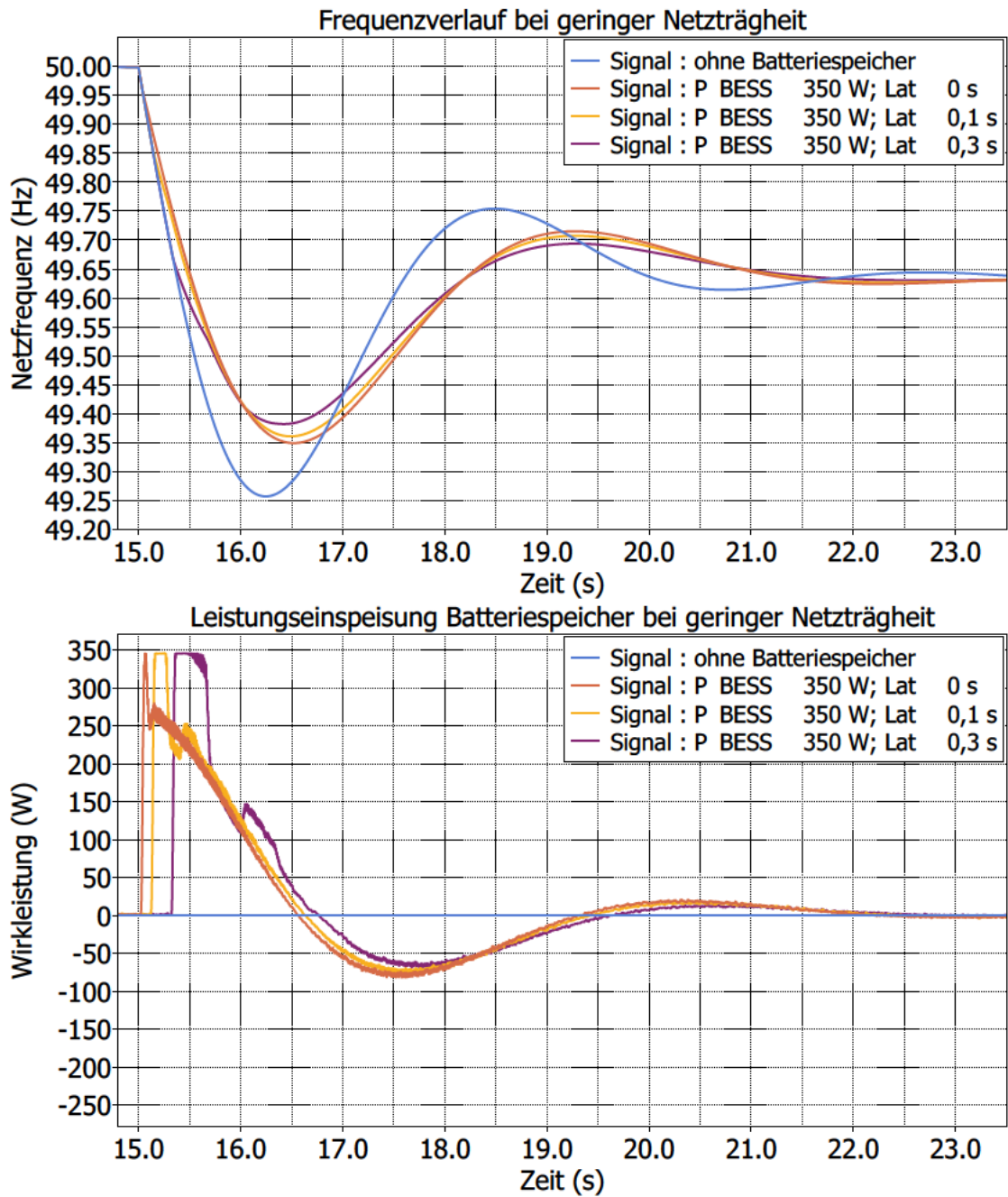


Abbildung 5.7: Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei geringer Netztr agheit und  $P_{\text{BESS}} = 350 \text{ W}$  sowie Latenzen im Millisekundenbereich

Tabelle 5.4: Szenario 2 - Ergebnisse mit Latenzen im Millisekundenbereich bei hoher Netzträgheit

Latenz (s)	$P_{\text{BESS}}$ (W)	RoCoF (Hz s <sup>-1</sup> )	Verringerung RoCoF(%)	$f_{\text{min}}$ (Hz)
–	–	-0,489	–	49,431
frei	70	-0,475	3	49,437
0,1	70	-0,478	2	49,438
0,3	70	-0,484	1	49,440
frei	350	-0,421	14	49,465
0,1	350	-0,432	12	49,469
0,3	350	-0,460	6	49,476

Tabelle 5.5: Szenario 2 - Ergebnisse mit Latenzen im Millisekundenbereich bei geringer Netzträgheit

Latenz (s)	$P_{\text{BESS}}$ (W)	RoCoF (Hz s <sup>-1</sup> )	Verringerung RoCoF(%)	$f_{\text{min}}$ (Hz)
–	–	-0,925	–	49,258
frei	70	-0,877	5	49,278
0,1	70	-0,885	4	49,281
0,3	70	-0,905	2	49,286
frei	350	-0,708	23	49,350
0,1	350	-0,737	20	49,361
0,3	350	-0,815	12	49,382

### 5.2.2 Latenzen im Sekundenbereich

In einer vorherigen Forschungsarbeit des Fraunhofer ISIT [20], konnte in einer RMS-Simulation gezeigt werden, dass bei geringer Netzträgheit und einer Latenz von 2s, der SEBIR-Regler die Frequenzstabilität negativ beeinflusst.

Deshalb sollen in dieser Arbeit in einer HIL-Simulation die Auswirkungen der SEBIR-Regelung bei Übertragungslatenzen von 1,0s und 2,0s überprüft werden. Dies geschieht wieder unter Variation der Parameter  $P_{\text{BESS}}$  und Netzträgheit bzw. Netzanlaufzeitkonstante  $T_A$ .

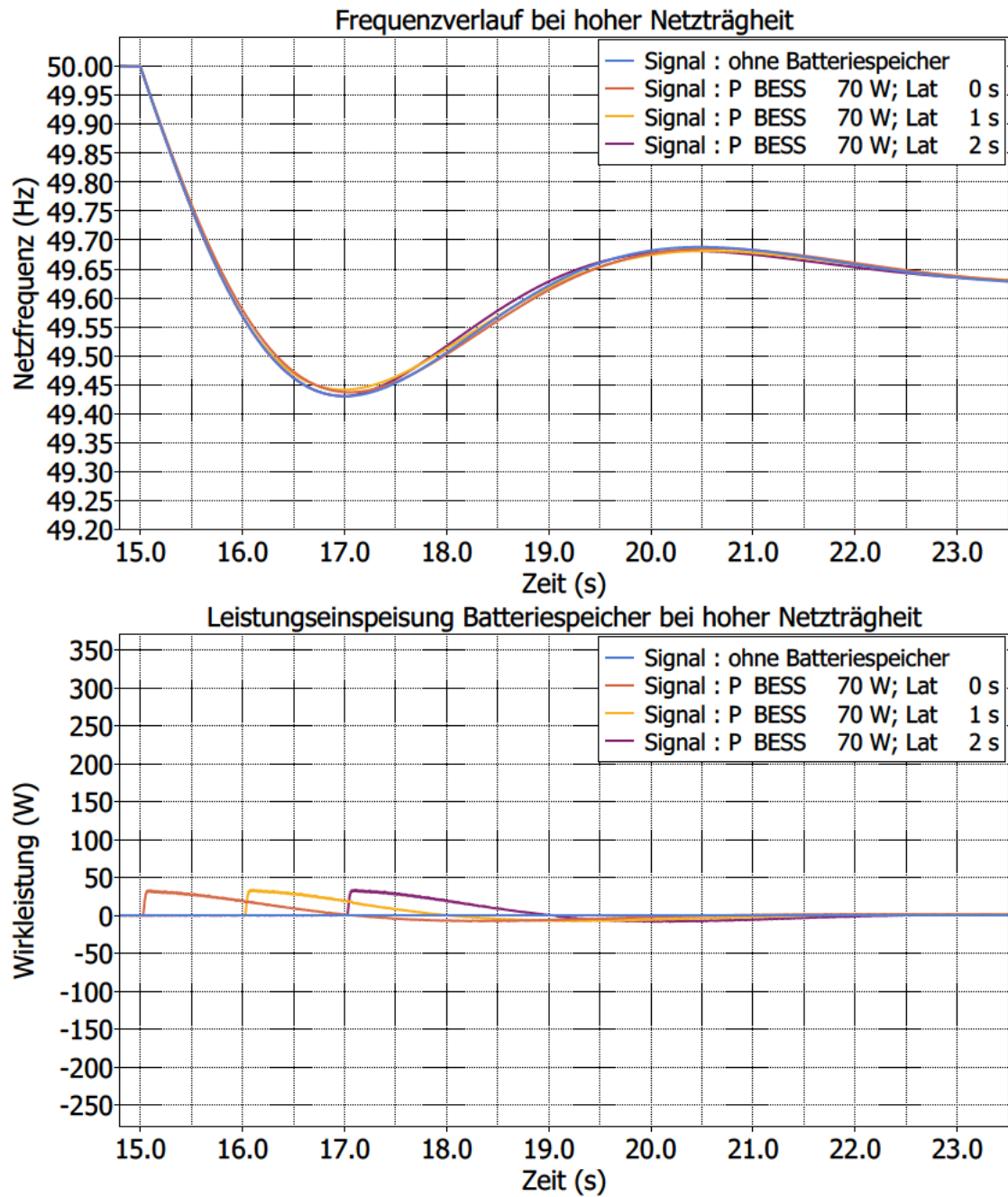


Abbildung 5.8: Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei hoher Netztr agheit und  $P_{\text{BESS}} = 70 \text{ W}$  sowie Latenzen im Sekundenbereich



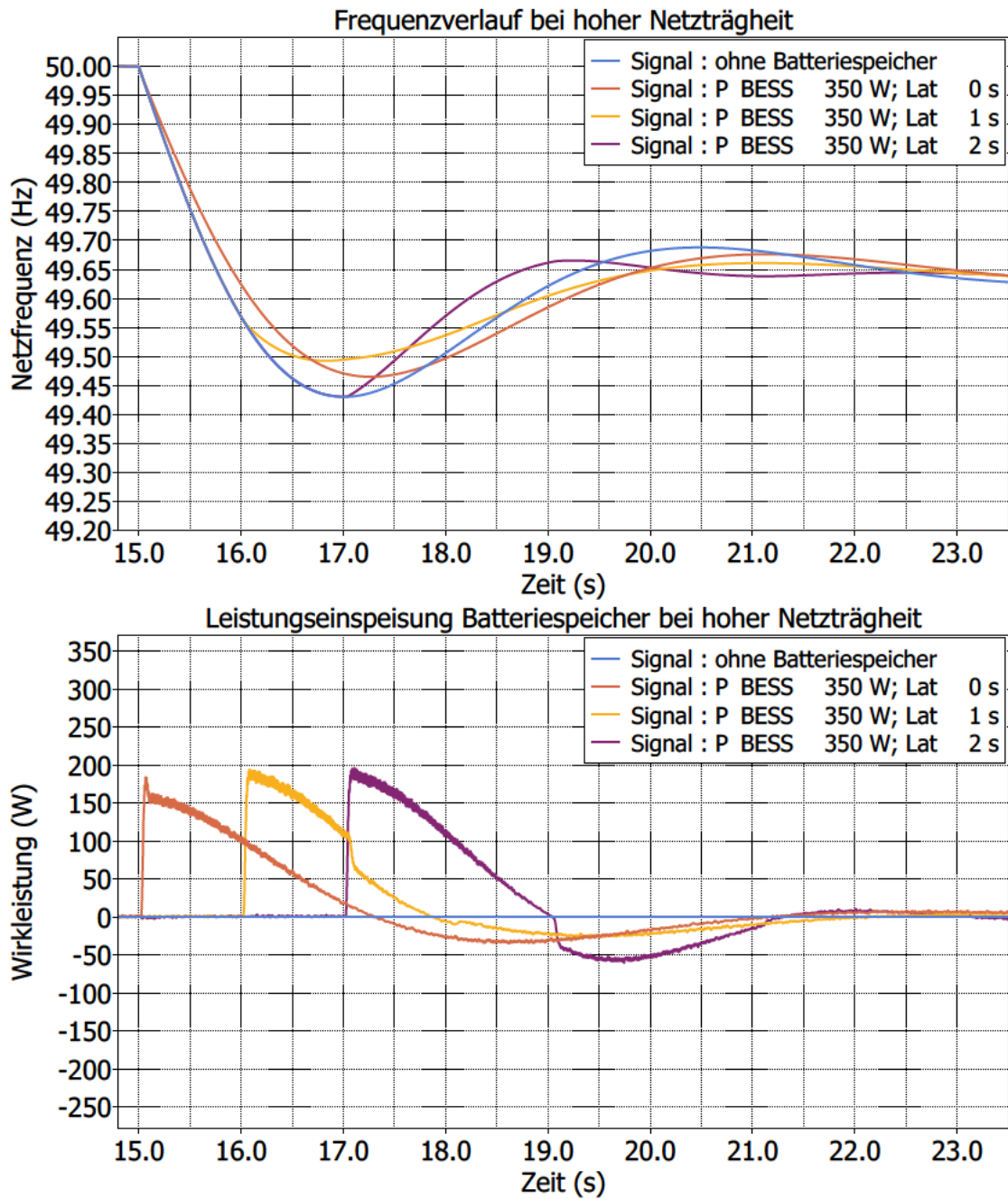


Abbildung 5.9: Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei hoher Netztr agheit und  $P_{\text{BESS}} = 350 \text{ W}$  sowie Latenzen im Sekundenbereich

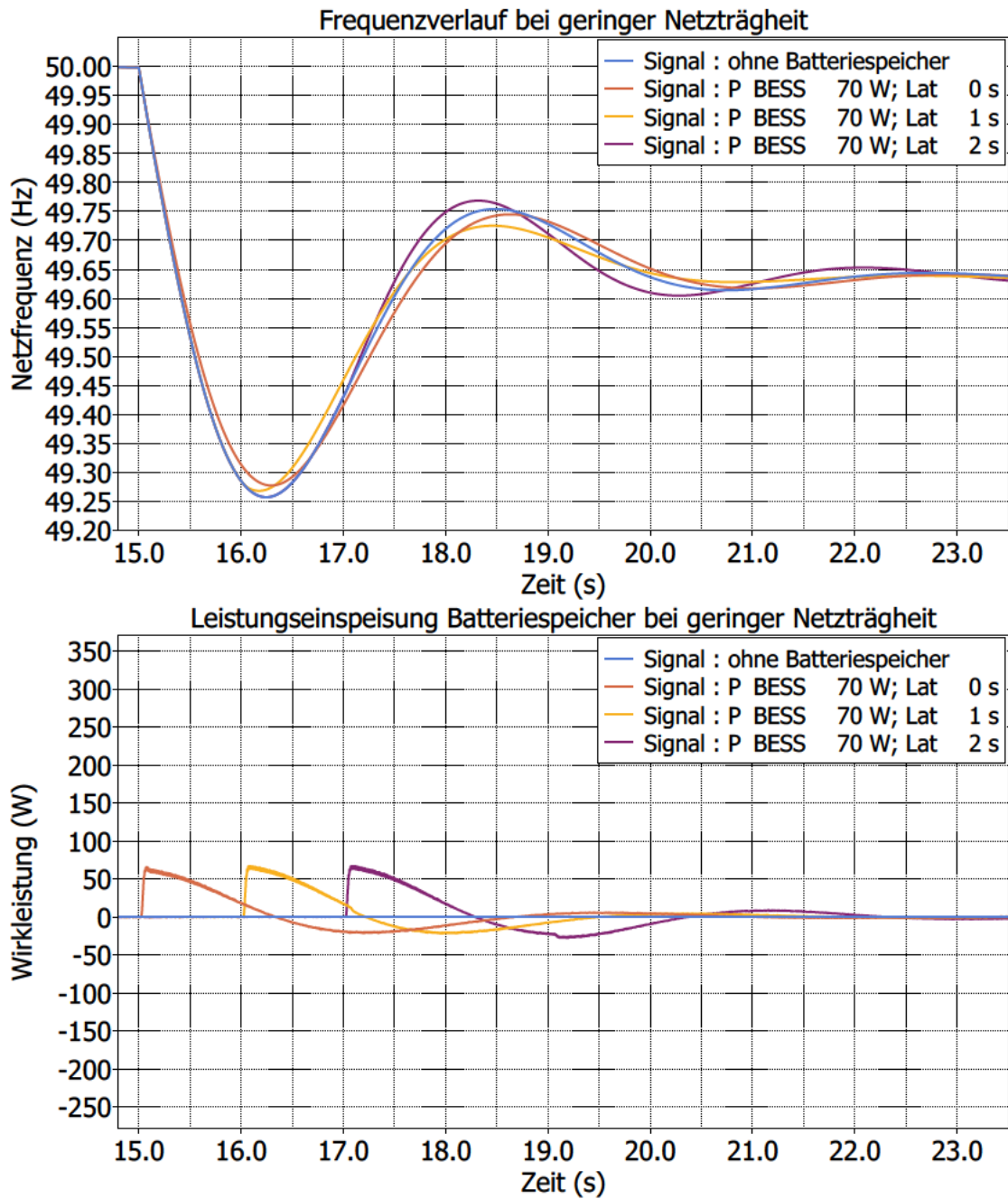


Abbildung 5.10: Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei geringer Netztr agheit und  $P_{\text{BESS}} = 70 \text{ W}$  sowie Latenzen im Sekundenbereich



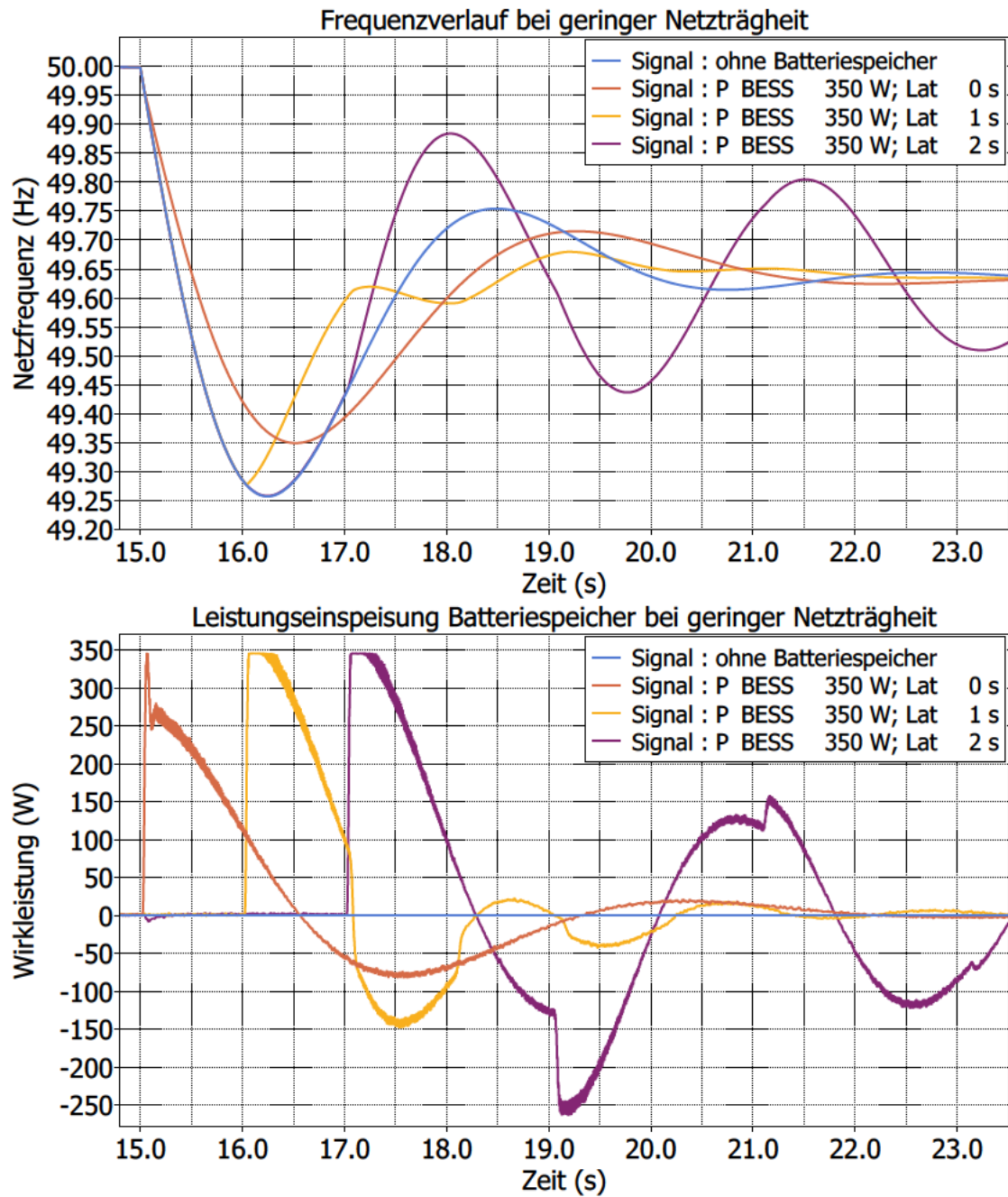


Abbildung 5.11: Szenario 2 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei geringer Netztr agheit und  $P_{\text{BESS}} = 350 \text{ W}$  sowie Latenzen im Sekundenbereich

Tabelle 5.6: Szenario 2 - Ergebnisse mit Latenzen im Sekundenbereich bei hoher Netzträgheit

Latenz (s)	$P_{\text{BESS}}$ (W)	RoCoF (Hz s <sup>-1</sup> )	Verringerung RoCoF(%)	$f_{\text{min}}$ (Hz)
–	–	-0,489	–	49,431
frei	70	-0,475	3	49,437
1,0	70	-0,489	0	49,442
2,0	70	-0,489	0	49,431
frei	350	-0,421	14	49,465
1,0	350	-0,489	0	49,493
2,0	350	-0,489	0	49,431

Tabelle 5.7: Szenario 2 - Ergebnisse mit Latenzen im Sekundenbereich bei geringer Netzträgheit

Latenz (s)	$P_{\text{BESS}}$ (W)	RoCoF (Hz s <sup>-1</sup> )	Verringerung RoCoF(%)	$f_{\text{min}}$ (Hz)
–	–	-0,925	–	49,258
frei	70	-0,877	5	49,278
1,0	70	-0,925	0	49,269
2,0	70	-0,925	0	49,258
frei	350	-0,708	23	49,350
1,0	350	-0,925	0	49,279
2,0	350	-0,925	0	49,258

### 5.3 Szenario 3 - Reglermodell mit Frequenzschätzung

Für Szenario 3 wird eine Frequenzschätzung auf dem Mikrocontroller implementiert. Der Baustein, der die Latenz einer Frequenzschätzung simuliert (siehe Abb. 3.2), wird aus dem Modell entfernt und durch eine Phasenregelschleife (Phase-locked loop (PLL)) aus der *PLECS*-Bibliothek ersetzt. In der Phasenregelschleife wird das Eingangssignal mit einem internen Referenzsignal verglichen. Die Regeldifferenz des Phasenwinkels beider Signale wird über Veränderung der Phasenlage bzw. Frequenz des Referenzsignals auf eine konstante Größe geregelt. So kann mit einer PLL die Frequenz des Eingangssignals geschätzt werden [11]. Damit die Frequenz geschätzt werden kann, wird die dreiphasige Netzspannung  $U_{3\text{ph}}$  als Eingangssignal für die PLL benötigt. Dementsprechend wird diese auf den richtigen Spannungsbereich skaliert ( $U_{3\text{ph},\mu\text{C}}$ ) und an den Mikrocontroller übertragen (siehe Abb. 5.12). Die Skalierung wurde mit der Umrechnung aus (4.3) vorgenommen.

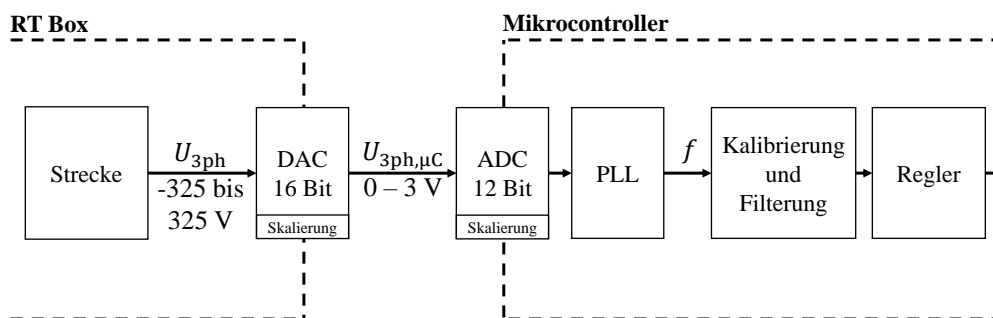


Abbildung 5.12: Szenario 3 - Anpassung der HiL-Implementierung

Die geschätzte Frequenz der Phasenregelschleife besitzt ein Grundrauschen und einen Offset in der ersten Nachkommastelle (siehe Abb. 5.13), daher muss das Signal für dieses Szenario stärker geglättet werden im Vergleich zu den vorherigen.

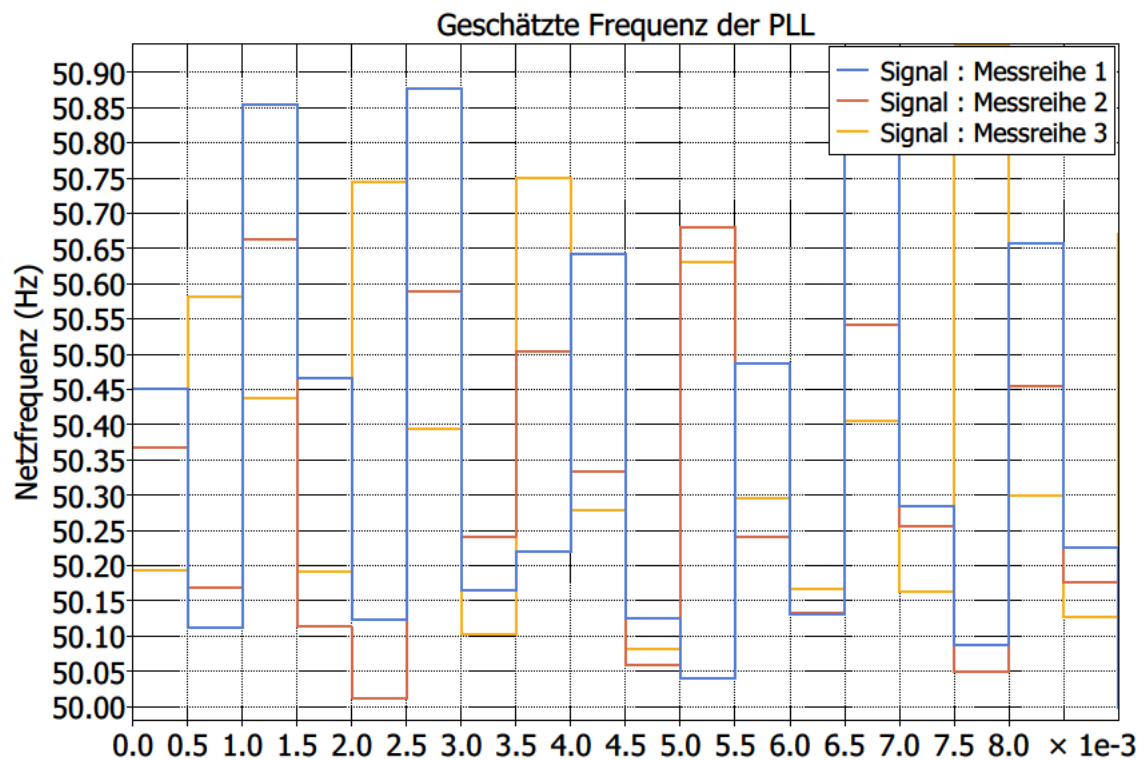


Abbildung 5.13: Szenario 3 - Gezeigt werden drei aufgenommene Messreihen von der ungeglätteten, geschätzten Frequenz der PLL nach Übertragung einer dreiphasigen Testspannung mit konstanter Frequenz von 50 Hz. Bei  $T_{a,\mu C} = 100 \mu s$  wird mit dem *PLECS*-Oszilloskop jedes fünfte Sample abgetastet

Eine stärkere Glättung des Signals, mittels Filterung und Mittelung, benötigt mehr Speicherplatz und Rechenleistung des Mikrocontrollers. Deshalb wird der *PLECS*-Baustein für eine periodische Mittelung, welcher effizienter programmiert ist und weniger Rechenleistung benötigt als der Baustein für eine gleitende Mittelung, für dieses Szenario genutzt (siehe entsprechende Beschreibung des 'Periodic Average' in der *PLECS*-Bibliothek).

Heuristisch kann so eine Implementierung für Szenario 3 gefunden werden, die eine ausreichende Genauigkeit der Stellgröße bewirkt und gleichzeitig mit der begrenzten Rechenleistung des Mikrocontrollers auskommt. Dabei wurde herausgefunden: ab einer periodischen Mittelung über 800 Werte (entspricht bei  $T_{a,\mu C} = 100 \mu s$  einer Periode von 80 ms) stößt der Mikrocontroller an seine Grenzen. Um die Schwankungen der PLL zu minimieren, wird deshalb mit einer Periode von 50 ms über 500 Werte gemittelt und ein Tiefpassfilter mit  $T_R = 50 ms$  noch vor dem SEBIR-Regler plaziert (siehe Abb. 5.12). Der

Tiefpassfilter nach dem Differenzierer des SEBIR-Regler wird ebenfalls auf  $T_R = 50$  ms gesetzt. Für die Kalibrierung des geschätzten Frequenzwertes und Ausgleich des Offsets, wird mit  $0,36$  Hz subtrahiert.

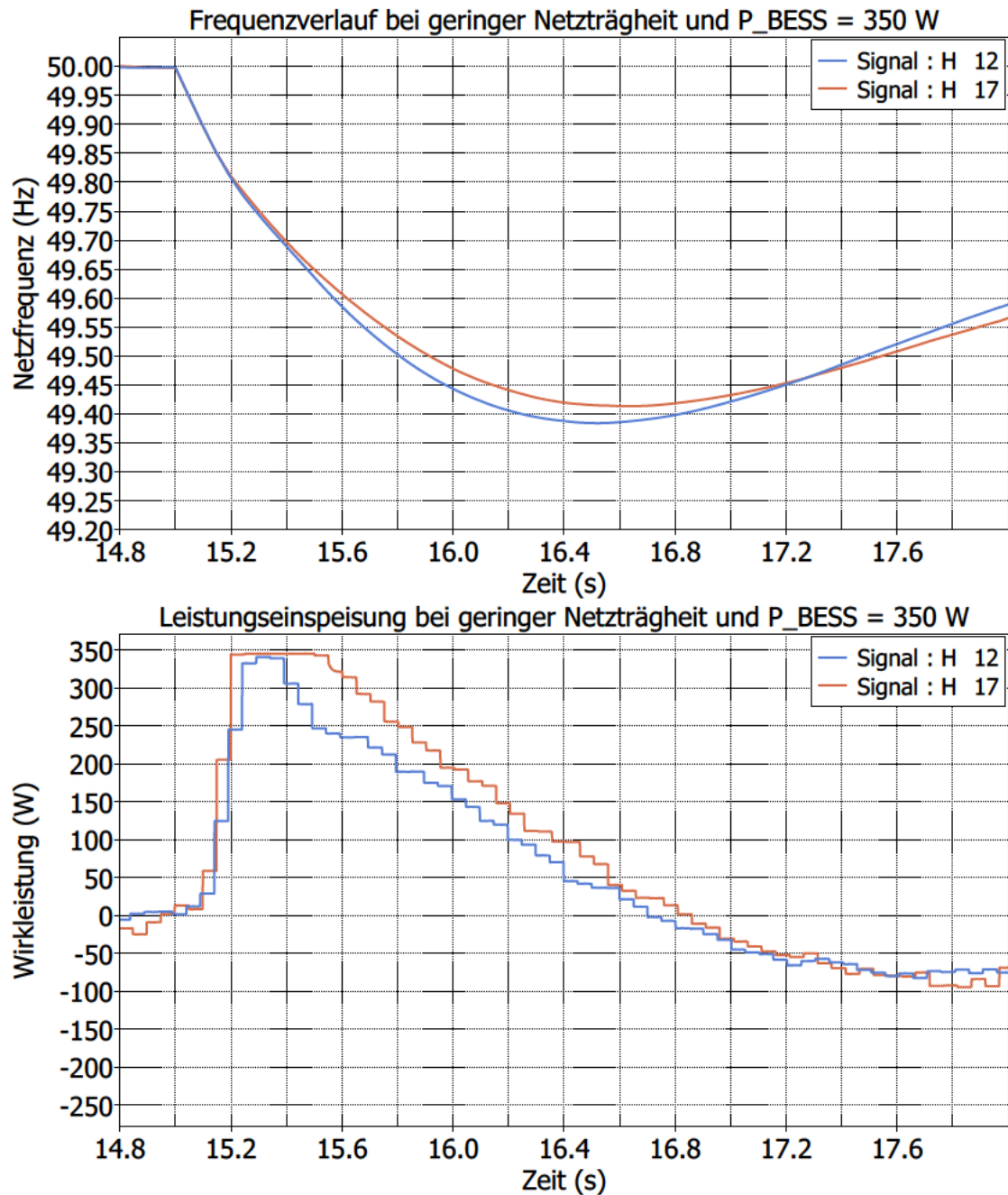


Abbildung 5.14: Szenario 3 - Anpassung der Trägheitskonstante  $H_{syn}$

Die Trägheitskonstante  $H_{\text{syn}}$  des SEBIR-Reglers muss ebenfalls angepasst werden, da der Regler z.B. bei  $P_{\text{BESS}} = 350 \text{ W}$  mit einem Wert von  $H_{\text{syn}} = 17,0 \text{ s}$  an die Leistungsbeschränkung stößt (siehe Abb. 5.14). Wie zuvor soll der Störsprung der Last bei geringer Netzträgheit die maximale Leistungseinspeisung des Batteriespeichers bewirken. So wurde für Szenario 3 bei  $P_{\text{BESS}} = 350 \text{ W}$  die Trägheitskonstante  $H_{\text{syn}}$  auf  $12,0 \text{ s}$  gesetzt und bei  $P_{\text{BESS}} = 70 \text{ W}$  auf  $10,0 \text{ s}$ .

Die CPU-Last des Mikrocontroller ist in Szenario 3 deutlich geringer als in Szenario 1 & 2, was an der periodischen Mittelung liegt (vgl.  $\text{CPU}_{\mu\text{C}}$  aus Abb. 4.2 und 5.8).

Tabelle 5.8: Übersicht der benutzten Zeitschritte sowie CPU-Last der RT Box und des Mikrocontrollers während der HiL-Tests für Szenario 3

$T_{\text{a,RT}}$	$\text{CPU}_{\text{RT}}$	$T_{\text{a},\mu\text{C}}$	$\text{CPU}_{\mu\text{C}}$
$10 \mu\text{s}$	$\sim 66 \%$	$100 \mu\text{s}$	$\sim 17 \%$

Im Folgenden werden die Ergebnisse des Szenario 3 jeweils bei hoher und geringer Netzträgheit im Vergleich zu Szenario 1 darstellt.

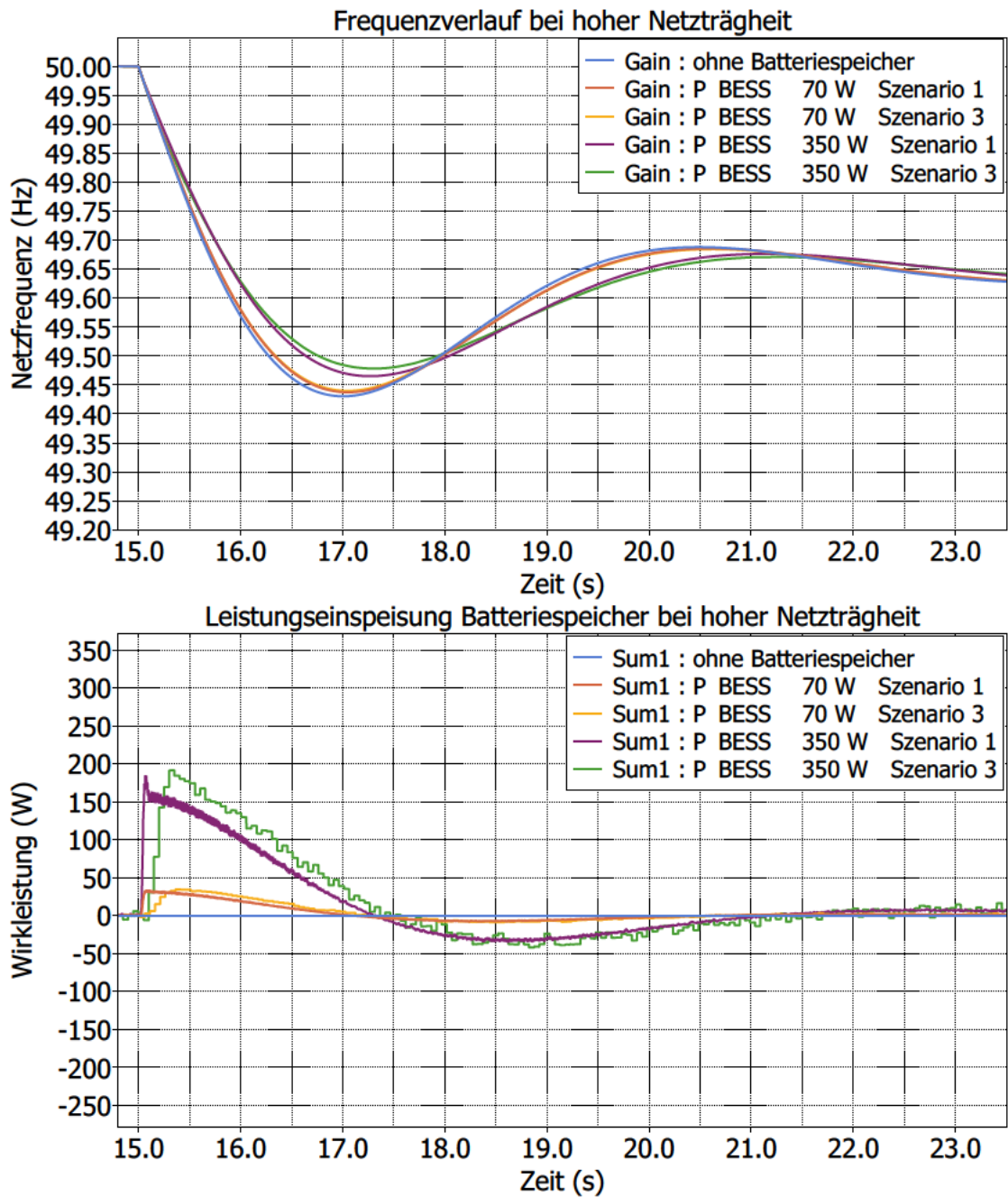


Abbildung 5.15: Szenario 3 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei hoher Netzträgheit im Vergleich zu Szenario 1

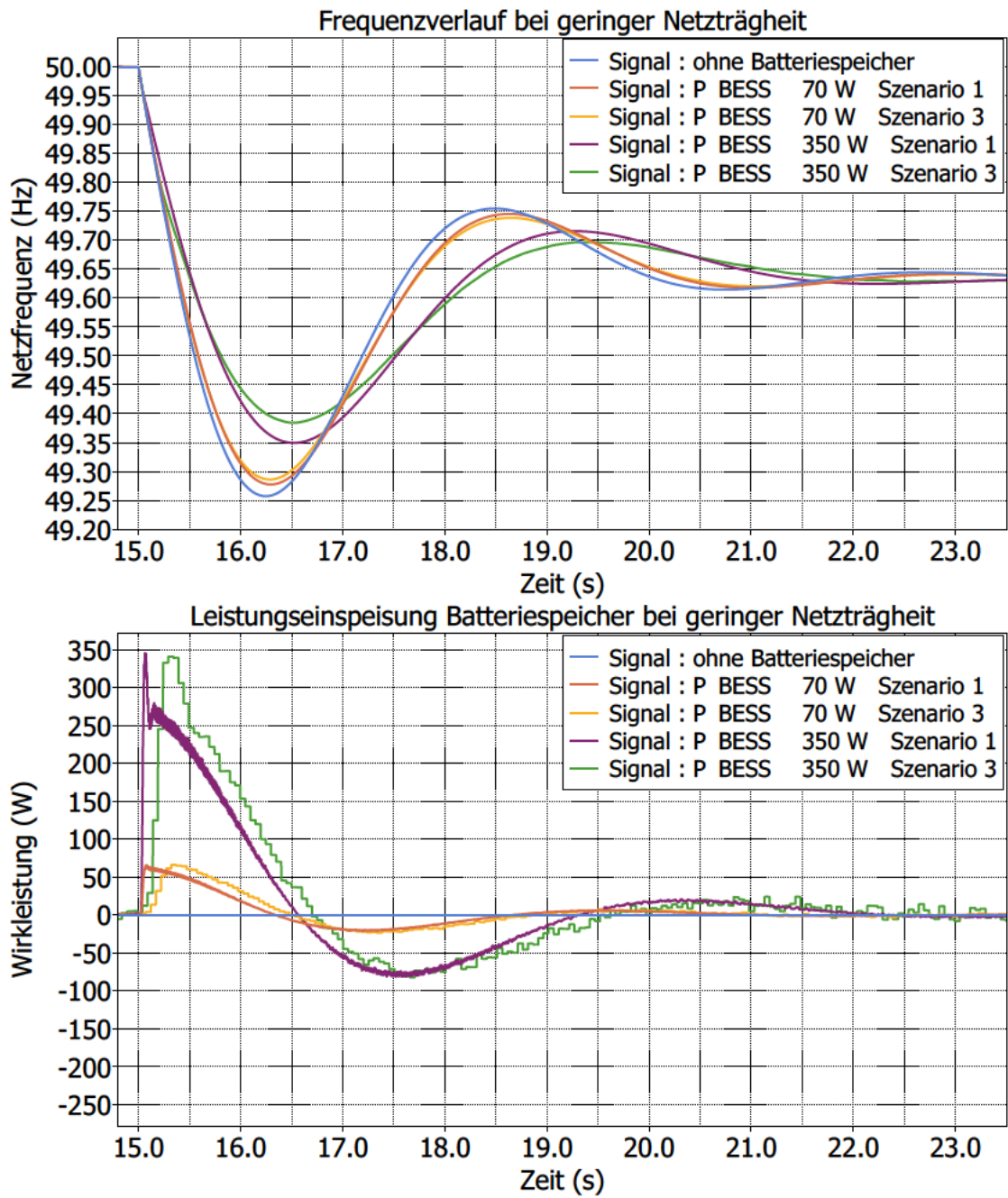


Abbildung 5.16: Szenario 3 - Frequenzverlauf und Leistungseinspeisung des Batteriespeichers bei geringer Netzträgheit im Vergleich zu Szenario 1



Tabelle 5.9: Szenario 3 - Ergebnisse bei hoher Netztr agheit im Vergleich zu Szenario 1

Szenario	$P_{\text{BESS}}$ (W)	RoCoF (Hz s <sup>-1</sup> )	Verringerung RoCoF (%)	$f_{\text{min}}$ (Hz)
–	–	-0,489	–	49,431
1	70	-0,475	3	49,437
3	70	-0,479	2	49,440
1	350	-0,421	14	49,465
3	350	-0,431	12	49,478

Tabelle 5.10: Szenario 3 - Ergebnisse bei geringer Netztr agheit im Vergleich zu Szenario 1

Szenario	$P_{\text{BESS}}$ (W)	RoCoF (Hz s <sup>-1</sup> )	Verringerung RoCoF (%)	$f_{\text{min}}$ (Hz)
–	–	-0,925	–	49,258
1	70	-0,877	5	49,278
3	70	-0,887	4	49,286
1	350	-0,708	23	49,350
3	350	-0,724	22	49,384

## 5.4 Diskussion der Ergebnisse

Im diesem Kapitel werden die Ergebnisse der Simulationen diskutiert. Dazu wird Bezug auf die in der Einleitung gestellten Forschungsfragen genommen. Da in dieser Arbeit ein Störgrößensprung simuliert wird, der dazu führt, dass die Frequenz fällt, treten nur negative RoCoF-Werte auf. Wenn im Folgenden von einer "Verringerung des RoCoF" geschrieben wird, ist damit eine Verringerung des absoluten RoCoF-Wertes gemeint.

### 5.4.1 Ergebnisse Szenario 1

Aus den Ergebnissen von Szenario 1 kann geschlossen werden, dass der SEBIR-Regler auch in einer HIL-Simulation die Netzfrequenz stabilisieren kann. Bei Betrachtung der Abbildung 5.1 und der Tabelle 5.2 kann man festhalten, dass bei hoher Netzträgheit beide Leistungsgrößen des Batteriespeichers zu einer Verringerung des RoCoF führen (um 3 % und 14 %) sowie den Frequenztiefpunkt anheben im Vergleich zum Referenzfall ohne Batteriespeicher. Jedoch könnte man anmerken, dass nicht die gesamte Leistung des Batteriespeichers eingespeist wird. Dieser wird in dieser Arbeit jedoch für den Fall von geringer Netzträgheit ausgelegt. An diesem Punkt wird deutlich, dass die Parametrierung des Reglers spezifisch angepasst werden könnte, um auch bei hoher Netzträgheit eine maximale Leistungseinspeisung des Batteriespeichers zu erreichen.

Bei geringer Netzträgheit (siehe Abb. 5.2 und Tabelle 5.3) wird jeweils die gesamte Leistung des Batteriespeichers genutzt. Beide Batteriespeicher haben einen größeren frequenzstabilisierenden Effekt im Vergleich zu einer Situation bei hoher Netzträgheit. Der 70 W Batteriespeicher verringert den RoCoF um 5 %, bei hoher Netzträgheit hat dieser den RoCoF um 3 % verringert. Der 350 W Batteriespeicher führt zu einer Verringerung des RoCoF von 23 %

Vergleicht man die Ergebnisse der Simulationen bei hoher und geringer Netzträgheit kann man festhalten: je größer der Batteriespeicher, desto größer der stabilisierende Effekt auf die Frequenz. Außerdem haben beide Speicher bei geringer Netzträgheit eine fast doppelt so hohe Verringerung des RoCoF (z.B. 14 % verglichen mit 23 % bei  $P_{\text{BESS}} = 350 \text{ W}$ , siehe Tab. 5.3) im Vergleich zu dem Fall bei hoher Trägheit im Netz. Das liegt auch daran, dass die definierte Netzanlaufzeitkonstante  $T_A$  für den Fall von geringer Netzträgheit etwa doppelt so hoch ist wie das  $T_A$  für den Fall hoher Netzträgheit (vgl. Tabelle 3.1).

## 5.4.2 Ergebnisse Szenario 2

### Latenzen im Millisekundenbereich

Betrachtet man die Ergebnisse der Simulationen mit Latenzen im Millisekundenbereich bei hoher Netzträgheit, fällt auf, dass die Ergebnisse mit Übertragungslatenzen von 100 ms und bei latenzfreier Übertragung nah beieinander liegen. Die Verringerung des RoCoF unterscheidet sich um 1 %. Mit  $P_{\text{BESS}} = 350 \text{ W}$  und gleicher Latenz ist der Unterschied mit 2 % ähnlich klein (siehe Tabelle 5.4 und Abb. 5.4 und 5.5)

Bei Latenzen von 300 ms ist der Unterschied zwischen  $P_{\text{BESS}} = 350 \text{ W}$  und  $P_{\text{BESS}} = 70 \text{ W}$  größer. Diese Latenz ist bei einem 350 W Batteriespeicher deutlich gravierender, da die Verringerung des RoCoF um 8 % sinkt. Bei  $P_{\text{BESS}} = 70 \text{ W}$  sinkt die Verringerung des RoCoF lediglich um 2 % (siehe Tab. 5.4 und Abb. 5.4 und 5.5).

Bei geringer Netzträgheit (siehe Tab. 5.5 und Abb. 5.6 und 5.7) ist ebenfalls zu beobachten, dass die Latenzen von 300 ms deutlich gravierender für die RoCoF-Verringerung sind, als 100 ms. Besonders ist dies zu erkennen bei  $P_{\text{BESS}} = 350 \text{ W}$ . Dort sinkt die Verringerung des RoCoF um 11 % im Vergleich zur latenzfreien Übertragung.

In allen Ergebnissen bei Latenzen im Millisekundenbereich ist interessant zu sehen: Je höher die Latenz ist, desto höher ist auch der Frequenztiefpunkt  $f_{\text{min}}$ , mit beiden Parametrierungen für  $P_{\text{BESS}}$ . Aus den Abbildungen ist zu erkennen, dass der Batteriespeicher mehr Energie eingespeist, wenn die Übertragungslatenzen größer werden (siehe Tab. 5.4 und 5.5 sowie Abb. 5.4 bis 5.7).

### Latenzen im Sekundenbereich

In der Analyse der Ergebnisse der Simulationen mit Latenzen im Sekundenbereich soll untersucht werden, inwieweit diese zu einer negativen Beeinflussung der Netzfrequenz führen. In den Abbildungen 5.8 und 5.10 kann man erkennen, dass ein 70 W Speicher mit Übertragungslatenzen bis 2 s den Verlauf der Frequenz nicht deutlich verändert, im Vergleich zu  $P_{\text{BESS}} = 350 \text{ W}$ . Dies gilt bei hoher als auch bei geringer Netzträgheit.

Obwohl der berechnete RoCoF nicht verringert wird, können Latenzen von 1,0 s zur Anhebung des Frequenztiefpunktes führen. In diesem Zusammenhang besonders zu erwähnen ist das Ergebnis mit 1,0 s in Abbildung 5.9. Der Frequenztiefpunkt wird um 30 mHz gehoben im Vergleich zur latenzfreien Übertragung.

In Abbildung 5.11 wird deutlich, dass ein Batteriespeicher ab einer bestimmten Größe, bei einer geringen Netzträgheit und mit einer Übertragungslatenz von 2,0s die Frequenzstabilität negativ beeinflussen kann. Auch wenn die Frequenz etwas schwingt, ist kein Aufschwingen zu beobachten, denn die Amplitude der Schwingung nimmt zeitlich ab.

### 5.4.3 Ergebnisse Szenario 3

Bei den Ergebnissen aus Szenario 3 ist besonders auffällig, dass der Verlauf der Leistungseinspeisung im Vergleich zu Szenario 1 deutlich stufenartiger verläuft (siehe Abb. 5.15 und 5.16). Das liegt unter anderem an der genutzten Filterung und Mittelung, welche sich von der in Szenario 1 & 2 benutzten unterscheidet. Außerdem ist ein stärkeres Grundrauschen zu erkennen, wenn ein Sollwert von 0 W vorgegeben wird. Dies wird etwa ab 22,0s an dem Verlauf der Leistungseinspeisung des Batteriespeichers deutlich (siehe Abb. 5.15 und 5.16).

Trotz verzögerter Einspeisung von etwa 0,25s in Szenario 3 (aufgrund der Glättung und Frequenzschätzung) im Vergleich zu Szenario 1 kann der RoCoF verringert werden (siehe Tab. 5.10 und 5.9). Die Unterschiede in der Verringerung im Vergleich zu Szenario 1 liegen bei 1% außer beim Fall  $P_{\text{BESS}} = 350 \text{ W}$  und geringer Netzträgheit. Dort liegt der Unterschied bei 2%.

Interessant ist, dass der Frequenztiefpunkt sowohl bei hoher als auch bei geringer Netzträgheit über dem aus Szenario 1 liegt (siehe Abb. 5.15 und 5.16). Gleiches lässt sich auch in den Ergebnissen von Szenario 2 bei einer Latenz von 0,3s beobachten (siehe z.B. Abb. 5.7). Das lässt sich dadurch erklären, dass insgesamt etwas mehr Leistung über die Zeit, bzw. Energie, durch den Batteriespeicher ins Netz eingespeist wird.

## 6 Zusammenfassung und Ausblick

Klassische Momentanreserve wird dem Energienetz in Zukunft vermehrt fehlen, aufgrund der fehlenden rotierenden Massen konventioneller Kraftwerke, welche dem Energienetz Trägheit bereitstellen [10] [37]. Die Herausforderung liegt in der Erschließung neuer Möglichkeiten um Momentanreserve bereitzustellen. Die Implementierung der Schwingungsgleichung einer Synchronmaschine in den Frequenzregler eines Batteriespeichers ist ein diskutierter Ansatz für die Bereitstellung virtueller Trägheit [17] [15]. Das Ziel ist, dass über die sogenannte SEBIR-Regelung eines Batteriespeichers die Trägheit eines Netzes erhöht und der RoCoF reduziert wird.

In der vorliegenden Arbeit wird der SEBIR-Regler in verschiedenen Hardware-in-the-Loop-Simulationen getestet. Zuvor wird mithilfe eines Mikrocontrollers und der *RT Box 1* der Firma *Plexim* der Teststand aufgebaut. Das zur Verfügung stehende EMT-Modell eines Netzmodells wird in ein HIL-Modell umgeformt und die Hardwarekomponenten konfiguriert, sodass die gewünschten Signale übertragen werden. Es werden 3 Szenarien definiert. Zum einen wird gezeigt, dass ein Batteriespeicher mit SEBIR-Regelung auch in einer HIL-Simulation Momentanreserve bereitstellen kann. Zum anderen wird der Einfluss von Latenzen in der Übertragung untersucht. Zuletzt wird eine Frequenzschätzung auf dem Mikrocontroller implementiert und die Ergebnisse mit und ohne eigene Frequenzschätzung verglichen.

## 6.1 Fazit

In dieser Arbeit wird gezeigt, dass ein Batteriespeicher mit SEBIR-Regelung in einer HIL-Simulation mit Abbildung der elektromagnetischen Transienten Momentanreserve bereitstellen und die Frequenz stabilisieren kann (siehe Kapitel 5.1). Das  $H_{\text{syn}}$  wird jeweils für den Fall von geringer Netzträgheit ausgelegt und sorgt in diesem Fall für maximale Leistungseinspeisung bei Störgrößensprung. Es kann argumentiert werden, dass eine Anpassung von  $H_{\text{syn}}$  im Fall hoher Netzträgheit auch zu maximaler Einspeisung führen würde (Abb. 5.1 bzw. 5.14).

Über den Einfluss von Latenzen lässt sich anhand der Ergebnisse aus Szenario 2 schließen: Wenn bei einem Feldtest des SEBIR-Reglers mit einer Kommunikation über Modbus-TCP Latenzen bis 100 ms auftreten würden, sinkt die RoCoF-Verringerung um maximal 2% im Vergleich zu latenzfreier Kommunikation. Bei größerer Latenz im Sekundenbereich sinkt die Verringerung deutlich stärker. Eine Latenz in der Signalübertragung von 2 s kann die Frequenzstabilität negativ beeinflussen, wenn ein Batteriespeicher eine kritische prozentuale Größe (mindestens 5%) im Netz überschreitet. Wenn der Zeitpunkt des Frequenztiefpunktes zeitlich nach der ersten latenzbehafteten Reaktion des Reglers liegt, führt dies zu einer erhöhten Leistungseinspeisung des Batteriespeichers und einer Erhöhung des Frequenztiefpunktes (siehe z.B. Abb. 5.9 bei einer Latenz von 1,0 s).

Aus Szenario 3 kann das Fazit geschlossen werden, dass die Rechenleistung des gewählten Mikrocontrollers ausreichend ist, um eine Frequenzschätzung zusätzlich zum Regelungsalgorithmus auszuführen. Jedoch muss der geschätzte Frequenzwert deutlich stärker geglättet werden als der Eingangswert bei Szenario 1 & 2. Hier sollte untersucht werden, ob die geschätzte Frequenz rauschärmer sein könnte, evtl. wenn der ADC die dreiphasige Spannung mit einer feineren Schrittweite wandelt. Interessant ist, dass die CPU-Last des Mikrocontrollers in Szenario 3 deutlich geringer ist als bei Szenario 1 & 2, was an den benutzten Bausteinen zur Mittelung liegt (vgl.  $CPU_{\mu C}$  aus Abb. 4.2 und 5.8). Die stärkere Glättung der Eingangsgröße des Reglers führt zu einer verzögerten Reaktion von etwa 0,25 s. Die RoCoF-Verringerung ist vergleichbar mit der bei Übertragungslatenzen von 0,1 s aus Szenario 2 (vgl. Tabellen 5.4 & 5.5 mit 5.9 & 5.10) und liegt bei höchstens 2% im Vergleich zur latenzfreien Übertragung.

Die HIL-Simulationen können in diesem Aufbau mit einem Zeitschritt von 10  $\mu\text{s}$  des Netzmodells und 100  $\mu\text{s}$  des Reglermodells durchgeführt werden.

## 6.2 Ausblick

Im Bezug auf die Implementierung sollte ein HiL-Teststand mit Kommunikation über Modbus-TCP aufgebaut werden. Da in zukünftigen Feldtests über dieses Protokoll die Sollwertvorgabe für den Batteriespeicher erfolgen soll, wäre es so vorher möglich zu testen, ob der SEBIR-Regler auch mit Kommunikation über Modbus-TCP den RoCoF verringern kann. Für diesen Test könnte eventuell die Middleware *opcsa* [27] genutzt werden, mit MATLAB Simulink und einem Echtzeitsimulator der Firma Speedgoat. Die nächsten Generationen der RT Box könnten auch fähig sein verschiedene Kommunikationsprotokolle, wie Modbus-TCP, zu testen. Ebenso sollte ein Mikrocontroller mit einer höheren Auflösung des ADC (16-Bit) getestet werden, da dieser für den Empfang der Eingangsgröße verantwortlich ist, welche möglichst rauscharm sein sollte. Darüber hinaus sollte genauer untersucht werden, wie viel Speicherplatz und Rechenleistung einzelne Software-Bausteine in *PLECS* bei der Implementierung auf einem Mikrocontroller benötigen, da die Grenzen des gewählten Mikrocontrollers bei einer Mittelung über 800 Werte erreicht werden, was bei  $T_{\mu C} = 100 \mu s$  einer Periode von 80 ms entspricht.

Im Bezug auf die Modellierung sollte im Anschluss an diese Arbeit das Batteriespeichermodell noch komplexer ausgebaut, z.B. ein State-of-Charge hinzugefügt werden. Ebenso wäre es sinnvoll auch die Leistungselektronik des Batteriespeichers zu modellieren und mit dem Regler, durch die Erzeugung von PWM-Signalen, die IGBTs anzusteuern und die hohe Auflösung der digitalen Eingänge der RT Box zu nutzen.

Außerdem könnte eine geregelte Anpassung der Trägheitskonstante  $H_{syn}$  entwickelt werden, welche diese für die momentane Netzträgheit einstellt. Ebenso könnte der SEBIR-Regler in einem Netzmodell ohne rotierende Massen getestet werden, sodass der Batteriespeicher die einzige Komponente ist, welche Momentanreserve bereitstellt.

# Literaturverzeichnis

- [1] ACEVEDO-ARENAS, César Y. ; CORRECHER, Antonio ; SÁNCHEZ-DÍAZ, Carlos ; ARIZA, Eduardo ; ALFONSO-SOLAR, David ; VARGAS-SALGADO, Carlos ; PETIT-SUÁREZ, Johann F.: MPC for optimal dispatch of an AC-linked hybrid PV/wind/biomass/H2 system incorporating demand response. In: *Energy Conversion and Management* 186 (2019), S. 241–257. – ISSN 01968904
- [2] ADS-TEC: *StoraXe® Industrial & Infrastructure - Skalierbare Batteriespeicher*. – URL [https://www.ads-tec.de/fileadmin/download/doc/brochure/Datenblatt\\_Energy\\_Industrial\\_DE.pdf](https://www.ads-tec.de/fileadmin/download/doc/brochure/Datenblatt_Energy_Industrial_DE.pdf). – Zugriffsdatum: 18.09.2020
- [3] ADS-TEC: *ADS-TEC Hochleistungsbatterie mit 2,5 MW in Schleswig-Holstein am Netz - ads-tec News*. 18.09.2020. – URL <https://www.ads-tec.de/unternehmen/presse/press/artikel/ads-tec-hochleistungsbatterie-mit-25-mw-in-schleswig-holstein-am-netz.html>. – Zugriffsdatum: 18.09.2020
- [4] AEMO: *Power System Model Guidelines*. – URL [https://aemo.com.au/-/media/files/stakeholder\\_consultation/consultations/electricity\\_consultations/2018/psm/power\\_system\\_model\\_guidelines\\_dfc.pdf?la=en&hash=F7301DE2450A0B803E8F92D348E3D519](https://aemo.com.au/-/media/files/stakeholder_consultation/consultations/electricity_consultations/2018/psm/power_system_model_guidelines_dfc.pdf?la=en&hash=F7301DE2450A0B803E8F92D348E3D519). – Zugriffsdatum: 28.09.2020
- [5] ARNDT BRÜNNER: *Lineare Regression – Methode der kleinsten Fehlerquadrate*. – URL <https://www.arndt-bruenner.de/mathe/scripts/regr.htm>. – Zugriffsdatum: 28.09.2020
- [6] BÉLANGER, Jean ; VENNE, P. ; PAQUIN, Jean-Nicolas: The What, Where and Why of Real-Time Simulation. In: *Planet Rt* (2010), Nr. 1, S. 37–49. – URL [https://blob.opal-rt.com/medias/L00161\\_0436.pdf](https://blob.opal-rt.com/medias/L00161_0436.pdf). – Zugriffsdatum: 01.09.2020



- [7] BRINGMANN, Eckard ; KR, Andreas: Model-Based Testing of Automotive Systems. In: *2008 International Conference on Software Testing, Verification, and Validation*, IEEE, 09.04.2008 - 11.04.2008, S. 485–493. – ISBN 978-0-7695-3127-4
- [8] CATERIANO YÁÑEZ, Carlos ; KAUFMANN, Christoph ; PANGALOS, Georg: Fast Power System Frequency Estimation by Shape Class Approximation for Synthetic Inertia Provision by Battery Energy Storage Systems. In: *IEEE 11th International Symposium on Power Electronics for Distributed Generation Systems*, IEEE, 2020
- [9] CHAD JACKSON: *What is Model-Based Software Development? | Lifecycle Insights*. 11.09.2019. – URL <https://www.lifecycleinsights.com/tech-guide/model-based-development/>. – Zugriffsdatum: 16.06.2020
- [10] DEUTSCHE ENERGIE-AGENTUR GMBH: Momentanreserve 2030. Bedarf und Erbringung von Momentanreserve 2030. (2016). – URL [https://www.dena.de/fileadmin/dena/Dokumente/Pdf/9142\\_Studie\\_Momentanreserve\\_2030.pdf](https://www.dena.de/fileadmin/dena/Dokumente/Pdf/9142_Studie_Momentanreserve_2030.pdf). – Zugriffsdatum: 16.06.2020
- [11] DEWIKI: *Phasenregelschleife*. 30.09.2020. – URL <https://dewiki.de/Lexikon/Phasenregelschleife>. – Zugriffsdatum: 30.09.2020
- [12] ENTSO-E: *Nordic report Future system inertia - ENTSO-E Docs*. – URL [https://eepublicdownloads.blob.core.windows.net/public-cdn-container/clean-documents/Publications/SOC/Nordic/Nordic\\_report\\_Future\\_System\\_Inertia.pdf](https://eepublicdownloads.blob.core.windows.net/public-cdn-container/clean-documents/Publications/SOC/Nordic/Nordic_report_Future_System_Inertia.pdf). – Zugriffsdatum: 02.09.2020
- [13] ENTSO-E: *Rate of Change of Frequency (RoCoF) withstand capability*. – URL [https://eepublicdownloads.blob.core.windows.net/public-cdn-container/clean-documents/Network%20codes%20documents/NC%20RfG/IGD\\_RoCoF\\_withstand\\_capability\\_final.pdf](https://eepublicdownloads.blob.core.windows.net/public-cdn-container/clean-documents/Network%20codes%20documents/NC%20RfG/IGD_RoCoF_withstand_capability_final.pdf). – Zugriffsdatum: 06.08.2020
- [14] ENTSO-E: *UCTE Operation Handbook*. – URL [https://eepublicdownloads.blob.core.windows.net/public-cdn-container/clean-documents/pre2015/publications/entsoe/Operation\\_Handbook/Policy\\_1\\_final.pdf](https://eepublicdownloads.blob.core.windows.net/public-cdn-container/clean-documents/pre2015/publications/entsoe/Operation_Handbook/Policy_1_final.pdf). – Zugriffsdatum: 03.09.2020
- [15] FRANZ, Julian ; CATERIANO YANEZ, Carlos ; PANGALOS, Georg: Optimized Swing Equation Control for Battery Energy Storage Systems. In: *2019 IEEE 13th International Conference on Compatibility, Power Electronics and Power Engineering*

- (*CPE-POWERENG*), IEEE, 23.04.2019 - 25.04.2019, S. 1–6. – ISBN 978-1-7281-3202-0
- [16] GAWLIK, Wolfgang ; KERDEGARBAKHS, Alireza ; PEŠEK, Markus ; ALÁCS, Christian: Entwicklung des Bedarfs an schneller Regelleistung im europäischen Verbundsystem. In: *e & i Elektrotechnik und Informationstechnik* 135 (2018), Nr. 8, S. 507–513. – ISSN 0932-383X
- [17] GONZALEZ-LONGATT, Francisco M. ; ALHEJAJ, Samir M.: Enabling inertial response in utility-scale battery energy storage system. In: *2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, IEEE, 28.11.2016 - 01.12.2016, S. 605–610. – ISBN 978-1-5090-4303-3
- [18] JACOB SAPIR: *Model-Based Design: What Is it, Why Use It and How to Use It*. 2019. – URL <https://realtechnologytools.com/model-based-design/>. – Zugriffsdatum: 09.07.2020
- [19] KIFFE, Axel: *Echtzeitsimulation leistungselektronischer Schaltungen für die Hardware-in-the-Loop-Simulation*. Berlin, Technische Universität Berlin, Dissertation, 2018. – URL <http://nbn-resolving.org/urn:nbn:de:101:1-2018112100581542559172>. – Zugriffsdatum: 15.06.2020
- [20] LARA-KRISTIN KREFT: *Simulationsbasierte Parameterstudie eines Virtual Inertia Controllers für einen Batteriespeicher im elektrischen Verbundnetz bei niedriger Systemträgheit*. Hamburg, Hochschule für Angewandte Wissenschaften Hamburg, Master of Engineering Thesis, 22.04.2020
- [21] LENTIJO, S. ; D'ARCO, S. ; MONTI, A.: Comparing the Dynamic Performances of Power Hardware-in-the-Loop Interfaces. In: *IEEE Transactions on Industrial Electronics* 57 (2010), Nr. 4, S. 1195–1207. – ISSN 0278-0046
- [22] MAHSEREDJIAN, J. ; DINAHAHI, V. ; MARTINEZ, J. A.: Simulation Tools for Electromagnetic Transients in Power Systems: Overview and Challenges. In: *IEEE Transactions on Power Delivery* 24 (2009), Nr. 3, S. 1657–1669. – ISSN 0885-8977
- [23] MATAR, Mahmoud ; IRAVANI, Reza: FPGA Implementation of the Power Electronic Converter Model for Real-Time Simulation of Electromagnetic Transients. In: *IEEE Transactions on Power Delivery* 25 (2010), Nr. 2, S. 852–860. – ISSN 0885-8977
- [24] MATHWORKS: *What is MIL, SIL, PIL, HIL and how do they integrate in Model Based Design approach? - MATLAB Answers*. 09.07.2020. –

- URL <https://de.mathworks.com/matlabcentral/answers/440277-what-is-mil-sil-pil-hil-and-how-do-they-integrate-in-model-based-design-approach>. – Zugriffsdatum: 09.07.2020
- [25] MCGRATH, B. P. ; HOLMES, D. G. ; GALLOWAY, J.: Improved power converter line synchronisation using an adaptive Discrete Fourier Transform (DFT). In: *2002 IEEE 33rd Annual IEEE Power Electronics Specialists Conference. Proceedings (Cat. No.02CH37289)*, IEEE, 23-27 June 2002, S. 821–826. – ISBN 0-7803-7262-X
- [26] NEW 4.0 - NORDDEUTSCHE ENERGIEWENDE: Speicherregelkraftwerk Hamburg-Curslack - Innovative Forschung zur Systemintegration erneuerbarer Energien. (2018). – URL [https://www.new4-0.de/wp-content/uploads/2019/07/Faktenblatt\\_Speicherregelkraftwerk.pdf](https://www.new4-0.de/wp-content/uploads/2019/07/Faktenblatt_Speicherregelkraftwerk.pdf). – Zugriffsdatum: 17.09.2020
- [27] OPCSA ; FRAUNHOFER ISIT (Hrsg.): *opcsa als Middleware für die Kommunikation über Modbus-TCP mit Simulink: Gespräch über Videokonferenz*. 16.06.2020
- [28] PÉREZ-NAVARRO, A. ; ALFONSO, D. ; ARIZA, H. E. ; CÁRCEL, J. ; CORRECHER, A. ; ESCRIVÁ-ESCRIVÁ, G. ; HURTADO, E. ; IBÁÑEZ, F. ; PEÑALVO, E. ; ROIG, R. ; ROLDÁN, C. ; SÁNCHEZ, C. ; SEGURA, I. ; VARGAS, C.: Experimental verification of hybrid renewable systems as feasible energy sources. In: *Renewable Energy* 86 (2016), S. 384–391. – ISSN 09601481
- [29] PLEXIM: *RT Box LaunchPad Interface User Manual - May 2020*. 2020. – URL <https://www.plexim.com/sites/default/files/launchpadinterfacemanual.pdf>. – Zugriffsdatum: 24.09.2020
- [30] PLEXIM: *RT Box mit LaunchPad Interface*. 2020. – URL [https://plexim.com/sites/default/files/pictures/rtbox1\\_side\\_breakout\\_1400px.jpg](https://plexim.com/sites/default/files/pictures/rtbox1_side_breakout_1400px.jpg). – Zugriffsdatum: 21.09.2020
- [31] PLEXIM GMBH: *Introduction to the RT Box using PLECS*. 2020. – URL [https://www.plexim.com/sites/default/files/tutorials/introduction\\_rtbox.pdf](https://www.plexim.com/sites/default/files/tutorials/introduction_rtbox.pdf). – Zugriffsdatum: 11.09.2020
- [32] PLEXIM GMBH: *RT Box | Plexim*. 2020. – URL [https://www.plexim.com/de/products/rt\\_box](https://www.plexim.com/de/products/rt_box). – Zugriffsdatum: 16.08.2020
- [33] REAL TIME AUTOMATION, INC.: *Modbus TCP/IP Overview*. 08.07.2020. – URL <https://www.rtautomation.com/technologies/modbus-tcpip/>. – Zugriffsdatum: 25.09.2020

- [34] SOCCI, Vince: Implementing a model-based design and test workflow. In: *First IEEE International Symposium on Systems Engineering*. Piscataway, NJ : IEEE, 2015, S. 130–134. – ISBN 978-1-4799-1920-8
- [35] SPEKTRUM ; SPEKTRUM AKADEMISCHER VERLAG (Hrsg.): *Trägheit*. 1998. – URL <https://www.spektrum.de/lexikon/physik/traegheit/14662>. – Zugriffsdatum: 30.09.2020
- [36] TEXAS INSTRUMENTS ; INCORPORATED [SPRS945 ; E ]: TMS320F28004x Microcontrollers datasheet (Rev. E). (2020). – URL [https://www.ti.com/lit/ds/symlink/tms320f280049.pdf?ts=1597589960592&ref\\_url=https%253A%252F%252F](https://www.ti.com/lit/ds/symlink/tms320f280049.pdf?ts=1597589960592&ref_url=https%253A%252F%252F). – Zugriffsdatum: 16.08.2020
- [37] TIELENS, Pieter ; VAN HERTEM, Dirk: The relevance of inertia in power systems. In: *Renewable and Sustainable Energy Reviews* 55 (2016), S. 999–1009. – ISSN 13640321

# A Anhang

## A.1 PLECS-Schaltbilder des Modells für Szenario 1 & 2

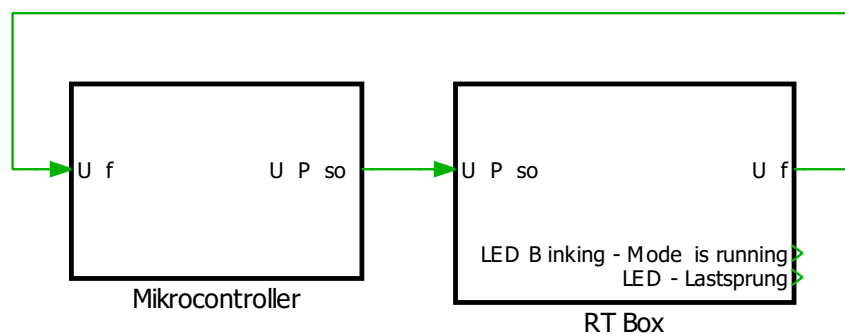


Abbildung A.1: Szenario 1 & 2 - Gezeigt sind die beiden Subsysteme der Modelle des Reglermodells (Mikrocontroller) und des Netzmodells (RT Box)

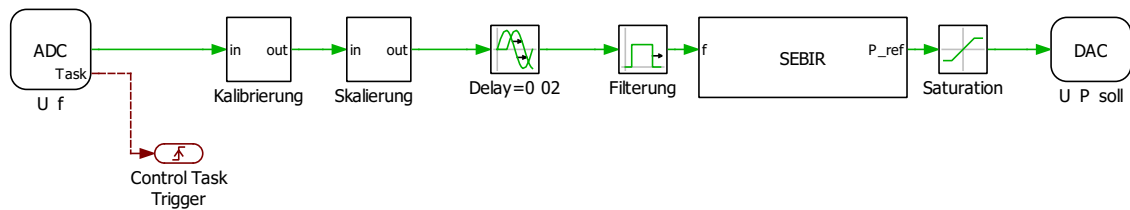


Abbildung A.2: Szenario 1 & 2 - Gezeigt ist das Reglermodell (Mikrocontroller)

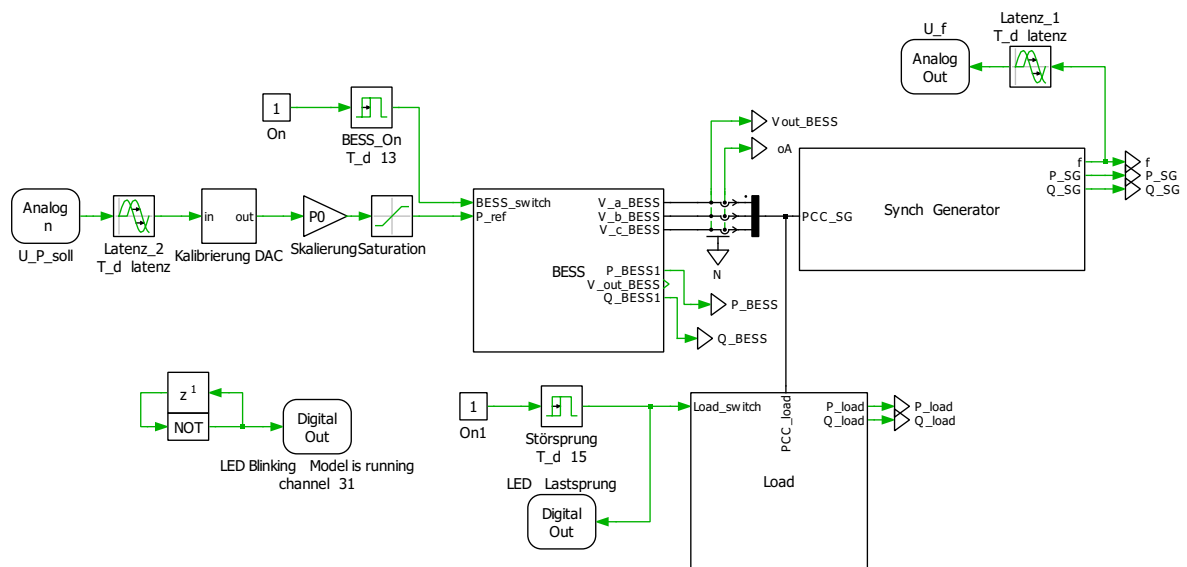


Abbildung A.3: Szenario 1 & 2 - Gezeigt ist das Netzmodell (RT Box)

## A.2 PLECS-Schaltbilder des Modells für Szenario 3

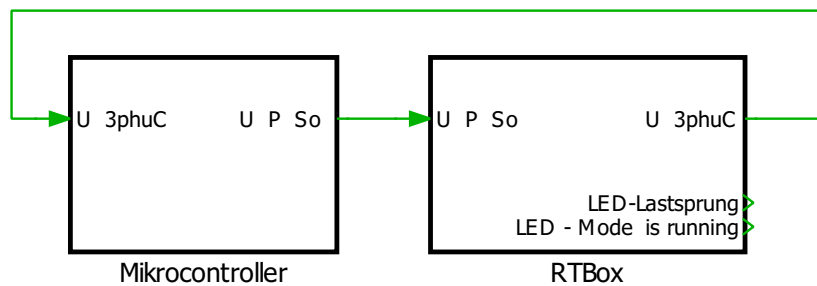


Abbildung A.4: Szenario 3 - Gezeigt sind die beiden Subsysteme der Modelle des Reglermodells (Mikrocontroller) und des Netzmodells (RT Box)

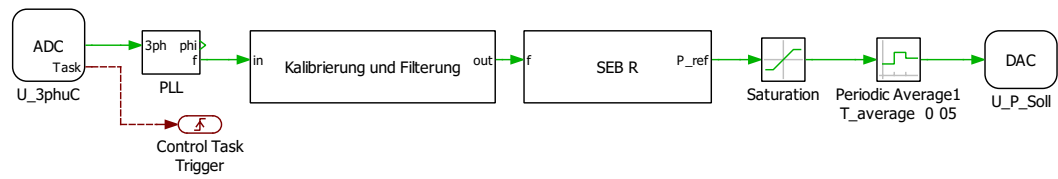


Abbildung A.5: Szenario 3 - Gezeigt ist das Reglermodell (Mikrocontroller)

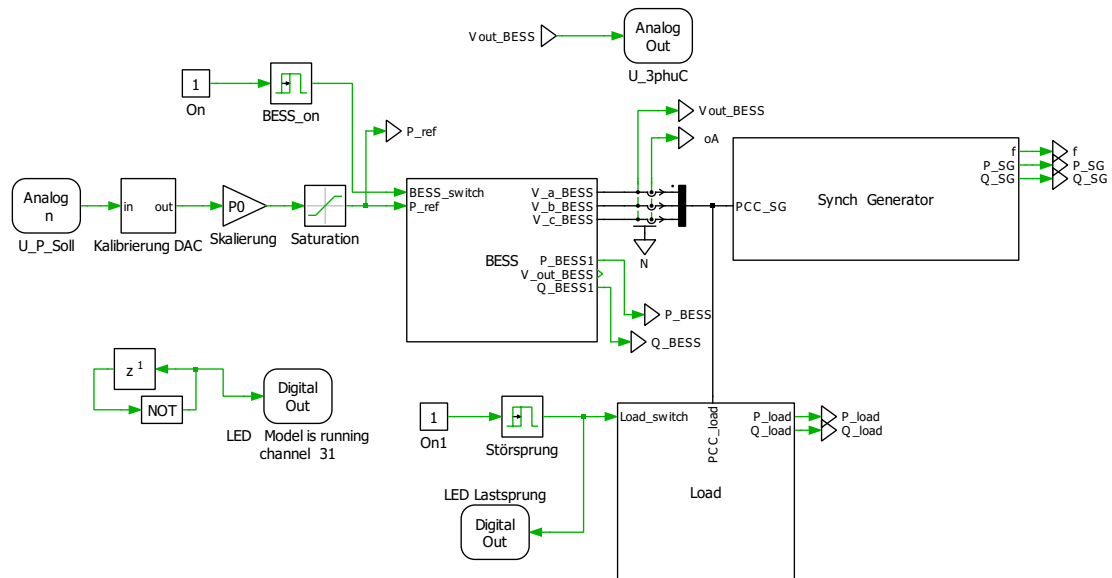


Abbildung A.6: Szenario 3 - Gezeigt ist das Netzmodell (RT Box)

# Glossar

**Controller-Modell** Repräsentation der Software des eingebetteten Systems [9].

**Plant-Modell** Repräsentation des physischen Systems, welches geregelt, gesteuert oder überwacht werden soll [9].

**PLECS** Simulationsplattform für leistungselektronische Systeme, entwickelt von der Firma Plexim.



## **Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit**

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

\_\_\_\_\_

Ort

Datum

Unterschrift im Original