

# ENTWICKLUNG EINER SIMULATIONSUMGEBUNG ZUR ANALYSE DEZENTRALER STEUERUNG ZEITLICH FLEXIBLER LASTEN IM VERTEILNETZ

Hakan SUSAR<sup>1(\*)</sup>, Veronika BARTA<sup>2</sup>, Stephanie UHRIG<sup>3</sup>

## Motivation

Angesichts der zunehmenden Elektrifizierung und der damit verbundenen Verringerung der Abhängigkeit von fossilen Energieträgern ist es von großer Bedeutung, die Netzstabilität zu gewährleisten. Der in diesem Jahr diskutierte und neu festgelegte § 14a des deutschen Energiewirtschaftsgesetzes (EnWG) fungiert als richtungweisende Leitlinie für die Steuerung steuerbarer Verbrauchseinrichtungen (SteuVE). Im Rahmen des von Verteilnetzbetreibern initiierten Forschungs- und Entwicklungsprojekts FLAIR<sup>2</sup> (Flexible Anlagen intelligent regeln) wurde ein dezentral arbeitender Algorithmus entwickelt, der anhand historischer Messdaten Spannungsspitzen und –senken erkennt und die netzorientierte Steuerung von Verbrauchern ermöglicht. Ein Feldversuch mit 72 FLAIR-Steuermodulen, der die Funktionsweise und Effektivität des Algorithmus bestätigte, lieferte wichtige Grundlagen für die anschließende Entwicklung der Simulationsumgebung [1], [2]. Da die Umgestaltung der elektrischen Verteilnetze den Endverbraucher, die Netzbetreiber, die Hersteller der SteuVE und den Gesetzgeber gleichermaßen betrifft, ist es erforderlich, die resultierenden Auswirkungen der dezentralen Steuerung z.B. bei wachsender Durchdringung von SteuVE auf das Stromnetz und den Kunden zu ermitteln. Die hier entwickelte Simulationsumgebung soll genutzt werden, um verschiedene Szenarien im Niederspannungsnetz und die Auswirkungen von Parameteränderungen des Algorithmus zu analysieren.

## Methodik

Für die Entwicklung der Simulationsumgebung wird Python als Programmiersprache ausgewählt. Vorteile von Python sind die schnelle Entwicklungszeit aufgrund der einfachen Syntax, eine große Auswahl an Bibliotheken wie `pandapower`, `pandas` und `numpy`, vorhandene Teamexpertise und eine starke Community-Unterstützung. Diese Faktoren sind ausschlaggebend für die Entscheidung zugunsten von Python gegenüber anderen Sprachen wie C++ oder Java.

Ein zentraler Bestandteil der Simulationsumgebung ist `pandapower` [3], eine umfangreiche Bibliothek speziell für Energiesystemsimulationen. `Pandapower` ermöglicht es, ein Stromnetz als übergeordnete Struktur zu erstellen, welche alle Einzelkomponenten wie Busse, Lasten und Controller umfasst. Ein Bus stellt in `pandapower` einen physischen Knoten im Netzwerk dar, an dem elektrische Elemente angeschlossen sind. Die Lasten repräsentieren die Verbraucher im Netz, während die Controller als algorithmische Elemente fungieren, die andere Netzelemente dynamisch steuern können.

Der FLAIR-Algorithmus wurde umgesetzt, indem das FLAIR-Steuermodul als eine eigene Controller-Klasse repräsentiert wird, die von der Basis-Controller-Klasse in `pandapower` abgeleitet ist. Diese Klasse nutzt eine `FlairAlgorithm`-Klasse und eine `AlgorithmParameter`-Klasse, worin der Flair-Algorithmus implementiert ist. Diese modulare Herangehensweise bietet mehrere Vorteile: Sie verbessert die Flexibilität, vereinfacht die Wartung und fördert die Skalierbarkeit des Projekts.

Zur Simulation werden eigens erstellte, sowie Open Source Last- und Einspeiseprofile verwendet, die eine sorgfältige Verwaltung und Organisation dieser Datenquellen erfordern. Im Rahmen dieser Organisation wurde ein spezielles Modul, der `DataProcessor`, entwickelt. Der `DataProcessor` beinhaltet Methoden zur Vorverarbeitung, zum Laden und zur Interpolation von Daten, die eine standardisierte und gleichzeitig flexible Handhabung verschiedener Datenformate gewährleisten.

---

<sup>1</sup> HM Hochschule München, Lothstr. 64, 80335 München, hakan.susar@hm.edu

<sup>2</sup> HM Hochschule München, Lothstr. 64, 80335 München, 089 / 1265 4481, veronika.barta@hm.edu, <https://sites.hm.edu/ises/>

<sup>3</sup> HM Hochschule München, Lothstr. 64, 80335 München, 089 / 1265 3416, stephanie.uhrig@hm.edu, <https://sites.hm.edu/ises/>

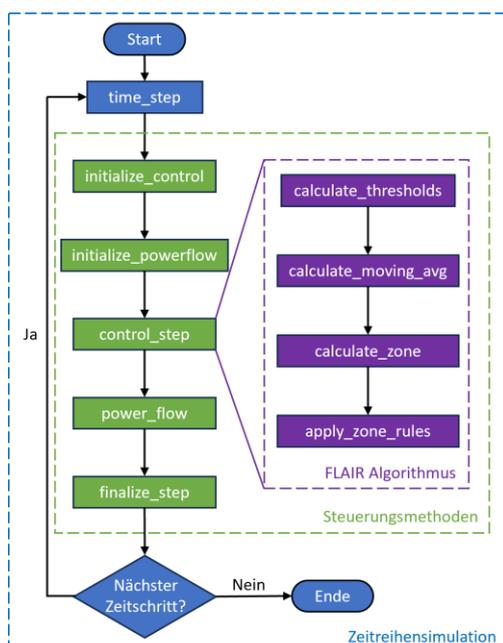


Abbildung 1: Darstellung der aufgerufenen Methoden in der Zeitreihensimulation. (basierend auf [4])

Pandapower ermöglicht es, mithilfe zugewiesener Lastprofile eine Zeitreihensimulation durchzuführen. Diese Simulation verarbeitet sequenziell die Leistungswerte und berechnet die resultierende Spannung an jedem Bus. Der Controller hat dabei die Möglichkeit, zwischen der Erfassung der Leistung und der Spannungsberechnung, die Leistung aus dem Lastprofil zu skalieren. Abbildung 1 veranschaulicht die in der Zeitreihensimulation aufgerufenen Methoden. Diese Methoden sind in der Controller-Klasse definiert und werden bei jedem Controller im Netz angewendet. Für jeden Zeitschritt liest die Methode `time_step` die entsprechende Leistung aus dem Lastprofil. Anschließend werden in den Methoden `initialize_control` und `initialize_powerflow` der Controller und die Lastflussberechnung initialisiert. Im `control_step` wird der vordefinierte FLAIR Algorithmus ausgeführt. Dabei werden zuerst in der Methode `calculate_thresholds` die Schwellwerte anhand der vergangenen Spannungsdaten berechnet. Nach der Ermittlung des gleitenden Mittelwerts in `calculate_moving_avg`, wird dieser in der Methode `calculate_zone` mit den berechneten Schwellwerten abgeglichen, um die zutreffende Zone und die damit verbundenen Skalierungswerte zu bestimmen. In der abschließenden Methode `apply_zone_rules` wird über-

prüft, ob die errechnete Zone mit den gesetzlichen Rahmenbedingungen und den festgelegten Regeln, wie beispielsweise spezifischen Einschränkungen für bestimmte Verbraucher, übereinstimmt. Anschließend werden die ausgelesenen Leistungswerte skaliert und für die Lastflussberechnung bereitgestellt. Nachfolgend berechnet die Methode `power_flow` die aktuelle Spannung. Im `finalize_step` speichert der Controller die berechnete Spannung, um sie in nachfolgenden Zeitschritten zu nutzen.

## Ausblick

Die Simulationsumgebung erfordert einen einheitlichen Ansatz zur Analyse und Visualisierung von Spannungs- und Leistungskennzahlen. Diese Standardisierung ist entscheidend, um zu untersuchen, wie verschiedene Parameter die Ergebnisse beeinflussen. Ein direkter Vergleich der Bewertungskriterien des Netzes, wie Leitungs- und Transformatorbelastung sowie das Spannungsband, zwischen Szenarien ohne Steuerung und solchen mit FLAIR-Controller-Einsatz ermöglicht die Analyse in die Auswirkungen der dezentralen Steuerung SteuVE auf das Netz. Die bisherigen Ergebnisse zeigen deutlich den Einfluss und die Effektivität des FLAIR-Controllers mit dem dezentralen Algorithmus auf das Netz.

## Danksagung

Diese Publikation ist im Rahmen des Kopernikus-Forschungsprojekts ENSURE – ‚Neue EnergieNetz-StruktURen für die Energiewende‘ – entstanden, gefördert vom Bundesministerium für Bildung und Forschung (BMBF).

## Referenzen

- [1] V. Barta, S. Baumgartner, S. Uhrig und R. Witzmann, „Algorithmus zur autarken netzdienlichen Steuerung von zeitlich flexiblen Lasten,“ in 17. Symposium Energieinnovation, Graz, 2022.
- [2] S. Baumgartner, V. Barta, S. Uhrig und R. Witzmann, „Praktische Umsetzung eines Reallabors für ein dezentrales Lastmanagement-Konzept,“ 17. Symposium Energieinnovation 2022, 2022.
- [3] L. Thurner, A. Scheidler, F. Schäfer et al, „pandapower - an Open Source Python Tool for Convenient Modeling, Analysis and Optimization of Electric Power Systems,“ IEEE Transactions on Power Systems, Bd. 6, pp. 6510-6521, Nov. 2018.
- [4] Fraunhofer IEE, University of Kassel, „Timeseries Module Overview — pandapower 2.13.1 documentation,“ 2023. [Online]. Available: [https://pandapower.readthedocs.io/en/latest/timeseries/timeseries\\_loop.html](https://pandapower.readthedocs.io/en/latest/timeseries/timeseries_loop.html). [Zugriff am 30 November 2023].