

Raum-zeitliche Optimierung des bidirektionalen EV-Ladens durch Verkehrsflussanalyse und stochastische Energiemodellierung

Paul Bauer, Benedikt Herold,
Stefan Wilker, Thilo Sauter

19. Symposium Energieinnovation

12. Februar 2026

ICT Energy&IT Group
introducing smart to the electricity grid



TECHNISCHE
UNIVERSITÄT
WIEN

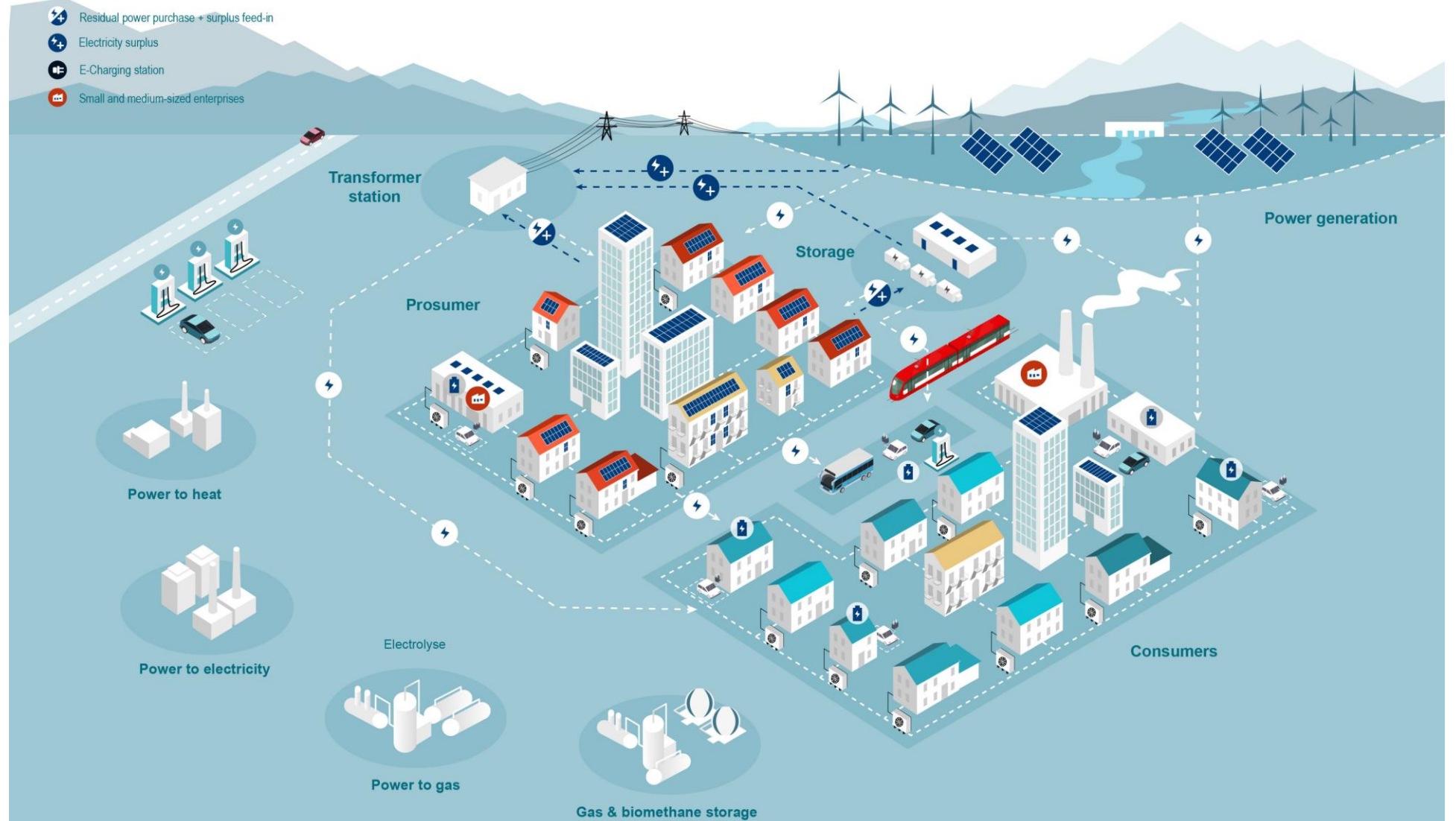
ICT

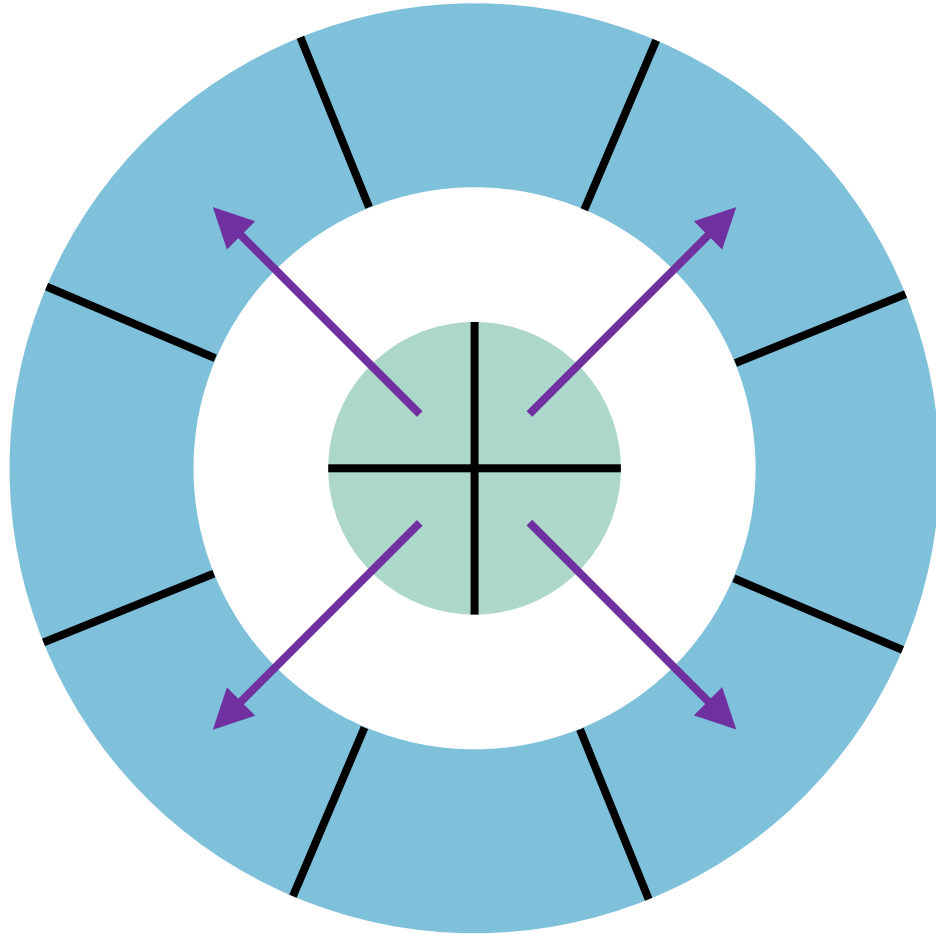
Institute of
Computer Technology





- Prosumer
- Consumer
- Electricity consumption
- Residual power purchase + surplus feed-in
- Electricity surplus
- E-Charging station
- Small and medium-sized enterprises

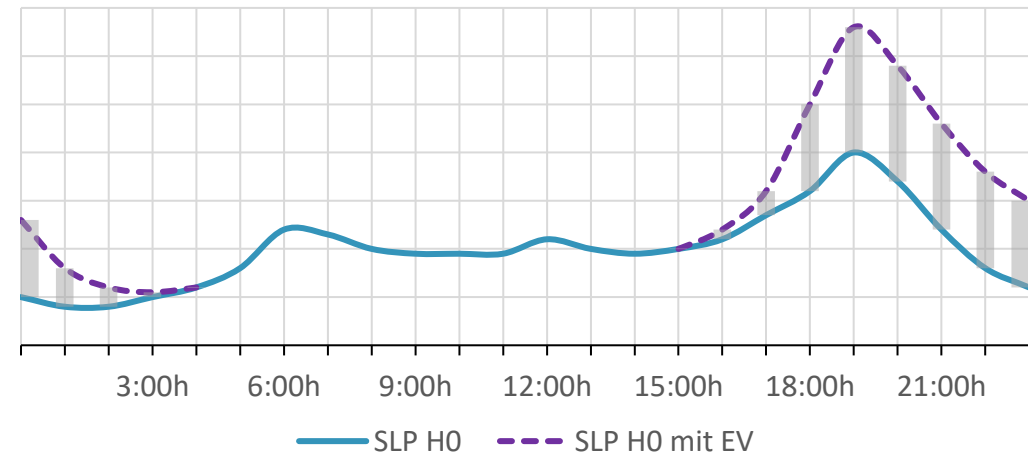


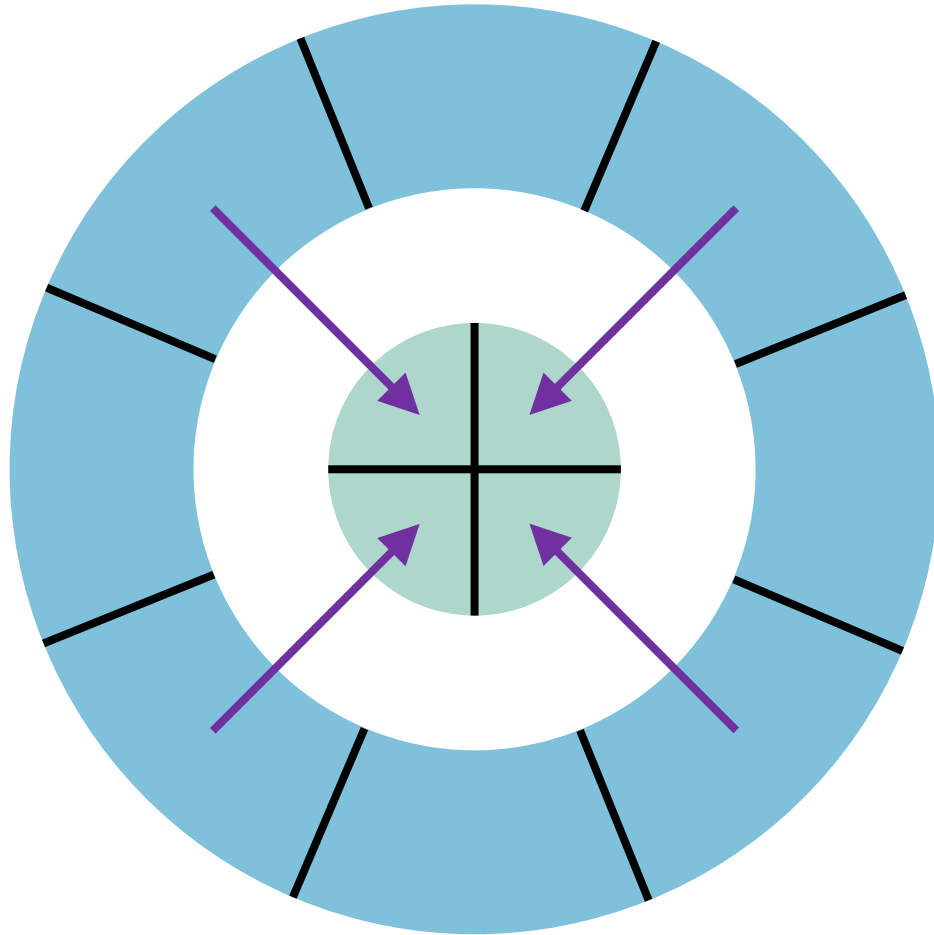


- Laden wenn Produktion durch Erneuerbare gering
- mehrere Engpässe
- Vollladestrategie

3

Lastprofil - Wohngebiet

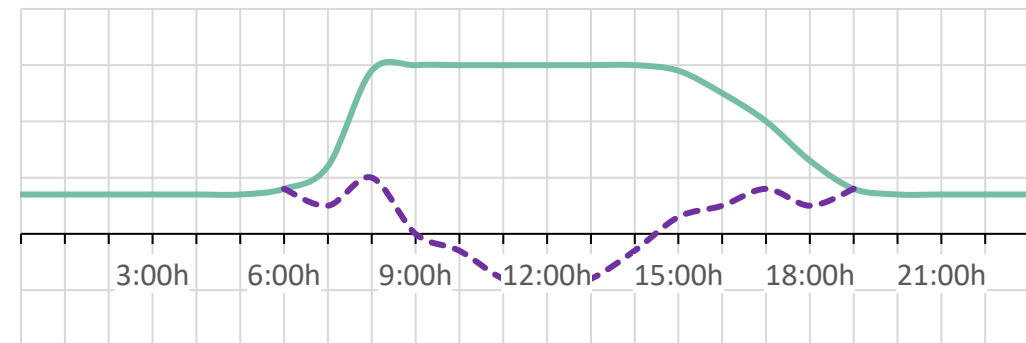




- Batterie State-of-Charge: fast voll
- PV-Überschuss
 - belastet das Stromnetz
 - Einspeisung durch Netzanschluss limitiert

4

Lastprofil - Industriegebiet



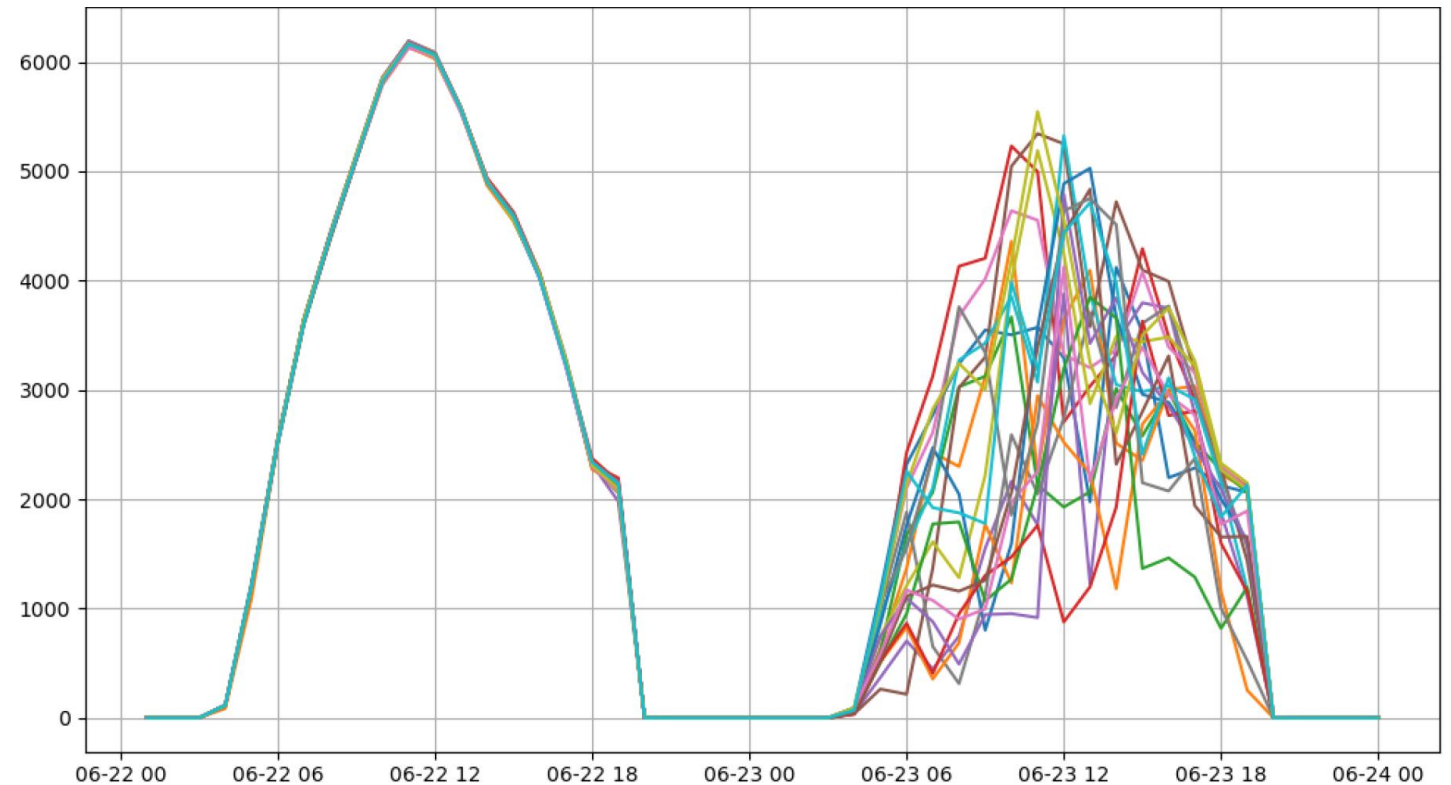
— SLP C2 - - - SLP C2 mit PV

– PV-Produktion als Beispiel

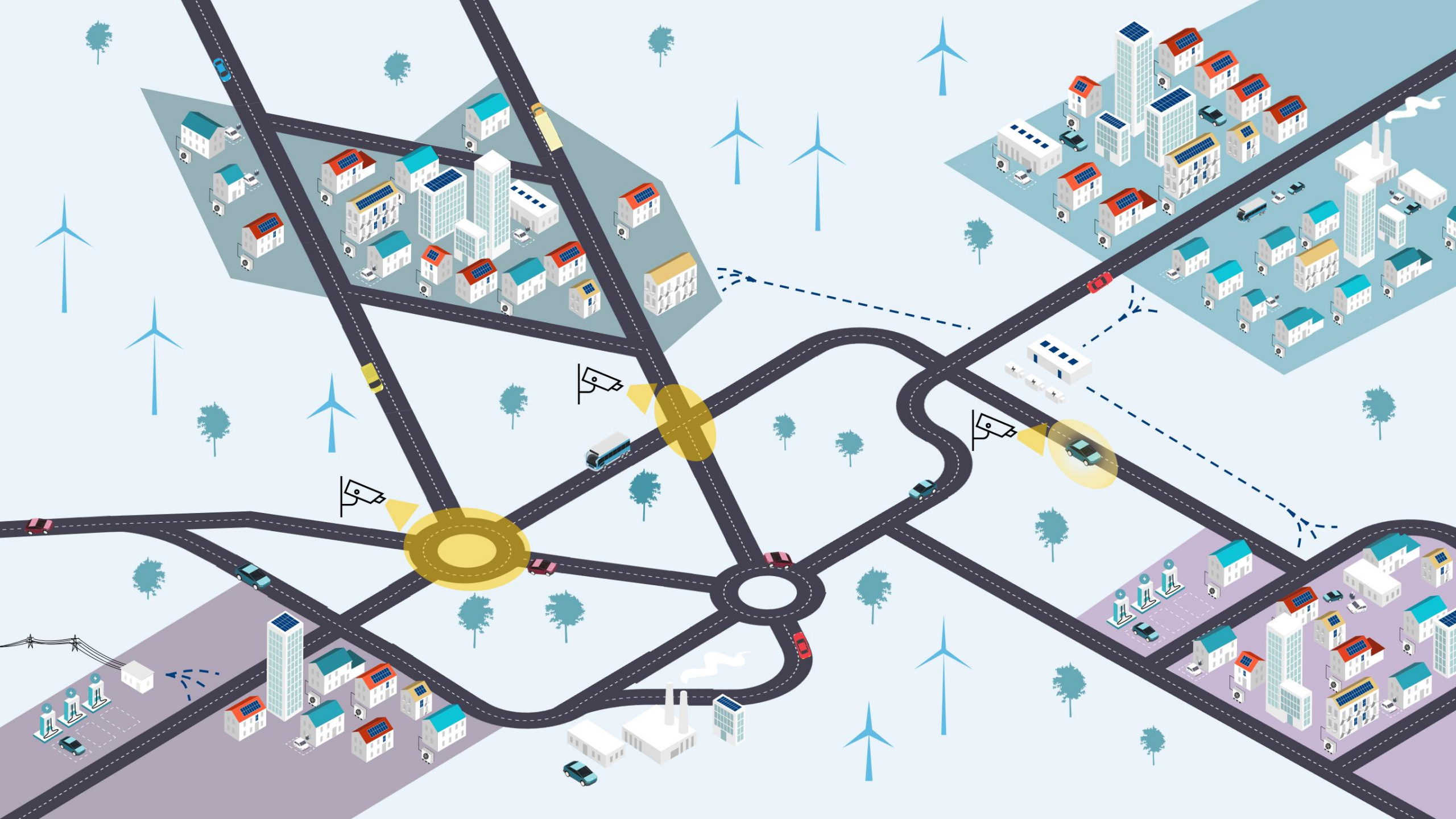
– Ähnliche Annahmen:

- Grundlast
- Wärmebedarf
- EV-Nutzungsprofile

AC Leistung [W]



Ensemble-Kurven einer Prognose-Gruppe



- variable Anzahl von Energiegemeinschaften mit Speicher
- variierende Strompreise
- (Ent-)Ladeeffizienz unter 100%
- Entscheidungsvariablen:
 - externer Strombezug
 - EV-Ladeleistung

$$\arg \min_{\substack{pwr(c,n) \\ soc(car,n)}} \sum_{n=1}^{192} \left(w * \left(\sum_c I(c,n) \right)^2 + x * \sum_{car} SoC(car,n) + y * \sum_c Smooth(c,n) + z * Chrg(n) \right)$$

$$I(c,n) = pwr(c,n) * cost(c,n)$$

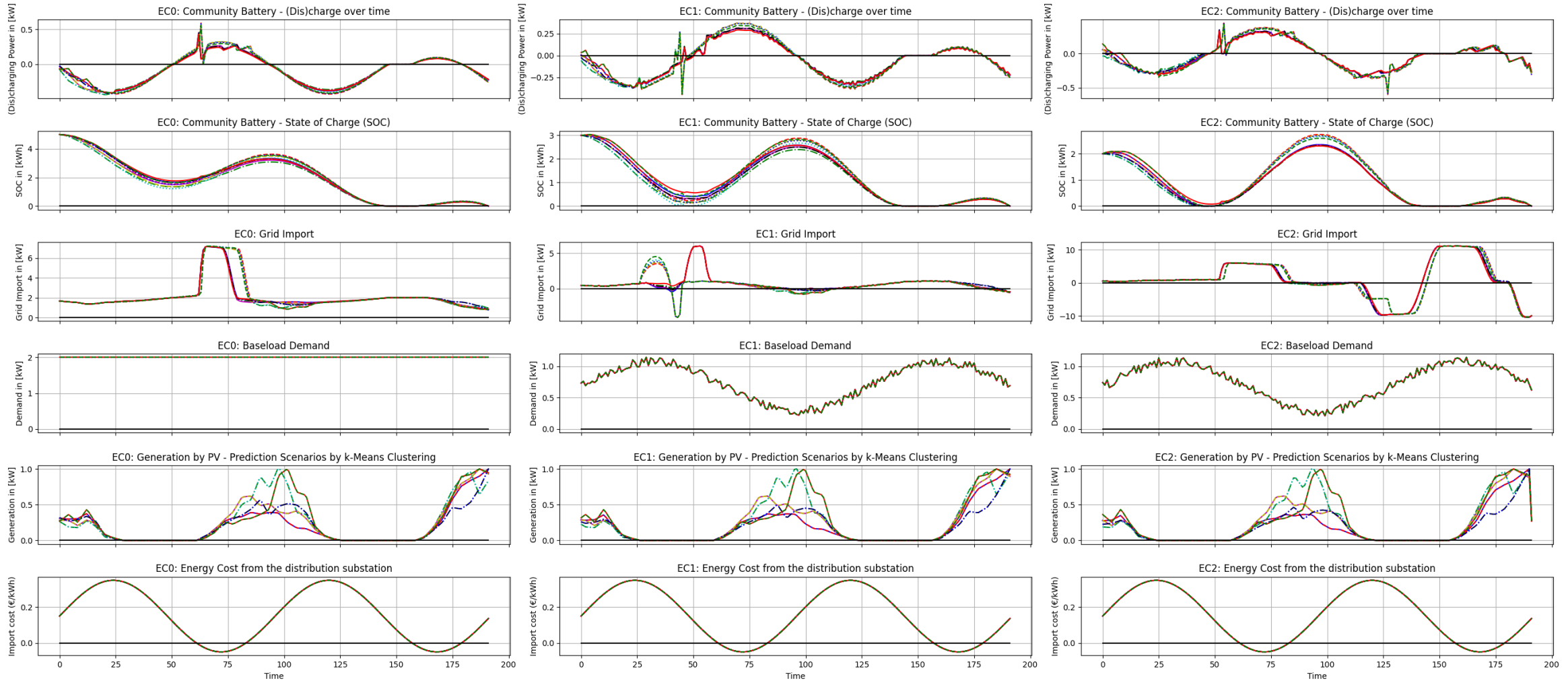
$$SoC(car,n) = soc_{max} - soc(car,n)$$

$$Smooth(c,n) = (pwr(c,n) - pwr(c,n-1))^2$$

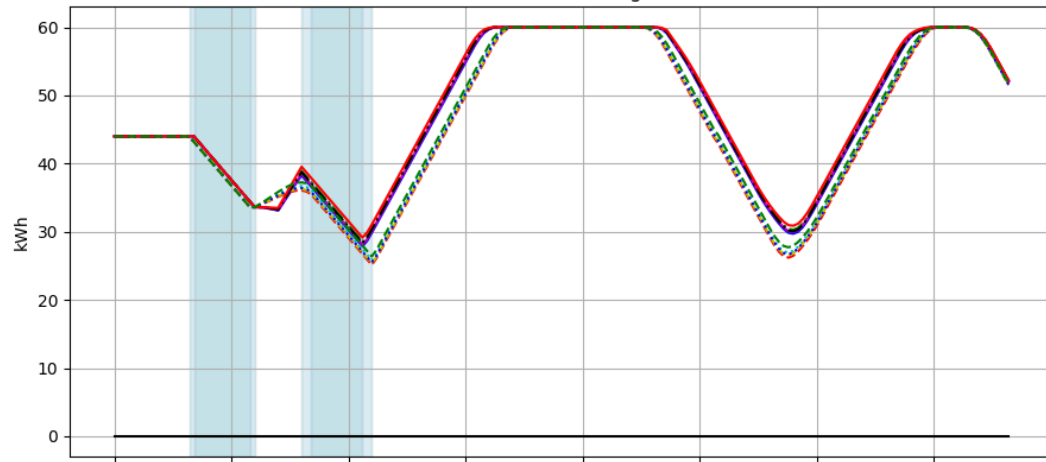


- Pyomo mit MPI-SPPY
- kompatibel mit Solvern wie GLPK, IPOPT, CBC für (MI)LP und NLP

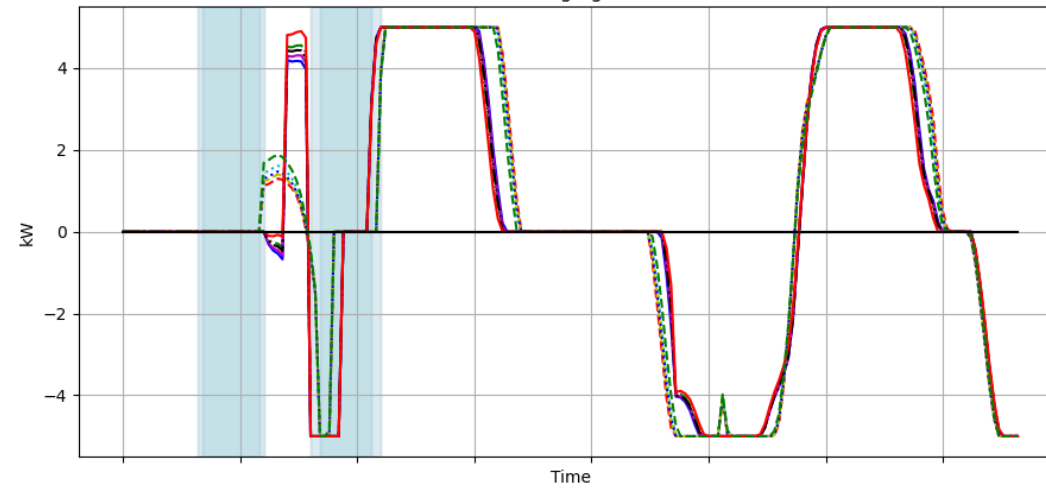
```
def scenario_creator(scenario_name):  
    car_map = {"0": 0, "1": 1, "2": 2, "3": 3}  
    prob_car_map = {"0": 0.1, "1": 0.7, "2": 0.1, "3": 0.1}  
    pv_id, car_id = scenario_name.split("_")  
  
    pv_profile, pv_probability = load_pv_data_from_csv(f"Scenario_{pv_id}")  
    car_flag = car_map[car_id]  
  
    model = build_model(pv_profile, car_flag)  
    sputils.attach_root_node(model, model.obj, id_first_stage_vars(model, 0, 25))  
    model._mpisppy_probability = pv_probability * prob_car_map[car_id]  
    return model
```



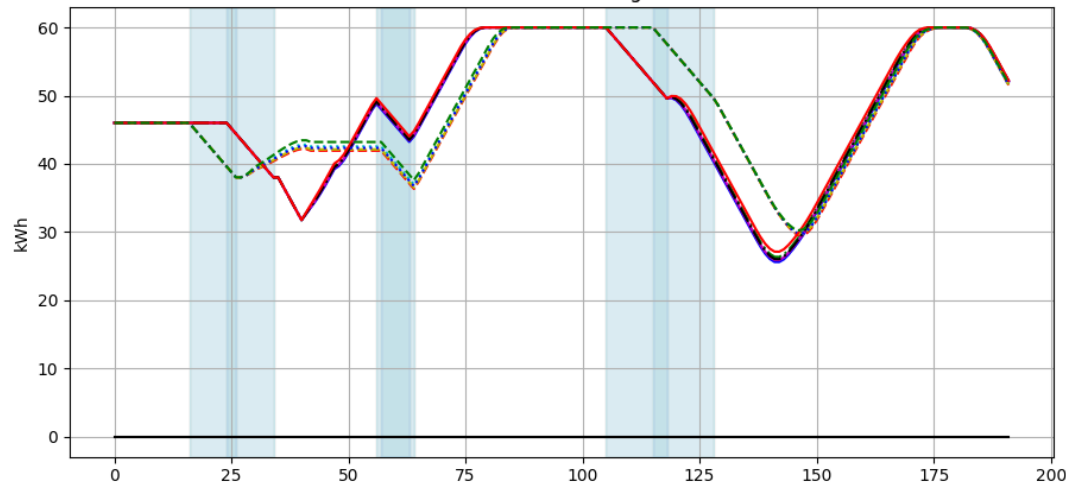
Car No. 0: State of Charge in kWh



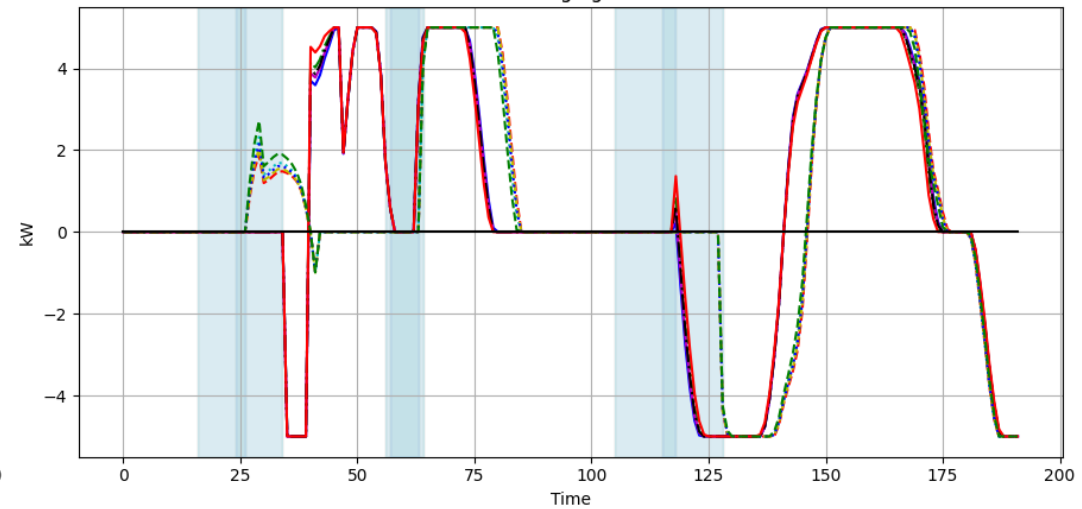
Car No. 0: Charging Power in kW



Car No. 1: State of Charge in kWh



Car No. 1: Charging Power in kW



- scenario 0_0
- scenario 0_1
- scenario 1_0
- scenario 1_1
- scenario 2_0
- scenario 2_1
- scenario 3_0
- scenario 3_1
- scenario 4_0
- scenario 4_1

– CAR 0: E1 => E2 => E1

– CAR 1: E1 => E0 => E2 => E1

Stefan Wilker
 Thilo Sauter
 Thomas Leopold
 Thomas Reisinger
 Paul Bauer
 Valentin Bauer
 Jonathan Kapfinger
 Dominic Hoffmann
 Claudia Benesch
 Carina Schöfl
 Verena Gartner
 Benedikt Herold
 Jakob Buchmeier
 Johannes Blaha
 Fiona Layes
 Dlovan Salim



ALUMNI IN RESEARCH

Peter Palensky, TU Delft
 Friederich Kupzog, AIT
 Albert Treytl, UWK
 Gerald Franzl, UWK

Aleksey Bratukhin, UWK
 Gerhard Zucker, AIT

Industriepartner (Exzerpt)



Wissenschaftliche Partner



Organisationen



12

Städtische und regionale Partnerschaften



Excited for more !

Paul Bauer

paul.bauer@tuwien.ac.at

13

This research was funded by CETPartnership, the European Partnership under Joint Call 2022 for research proposals, co-funded by the European Commission (GA N° 101069750) and the Austrian Research Promotion Agency (FFG).



TECHNISCHE
UNIVERSITÄT
WIEN