# Allgemeines

## Übersicht

Die Hinzunahme einzelner Anlagen und die Erweiterung der Funktionen wurden in Form von unterschiedlichen Directories realisiert.

### OptFlexEigen

Beinhaltet eine Optimierung hinsichtlich des größten Eigenverbrauchs. Kosten sind nicht enthalten. VDI-Profile und PV2013 Kassel. Nachregelung implementiert? Prognose enthalten?

* OptFlex\_PV\_Battery – Umstellung coopr-pyomo, verbesserte KPIs, „Dummy-Nachreglung“
* OptFlex\_PV\_Battery\_Gasboiler\_TES
* OptFlex\_PV\_Battery\_Gasboiler\_TES\_EHeater
* OptFlex\_CHP\_Battery\_TES\_Gasboiler

### OptFlexCost

Beinhaltet eine Optimierung hinsichtlich der geringsten Kosten. VDI-Profile und PV2013 Kassel. Prognose enthalten?

* OptFlex\_PV\_Battery\_Costs
* CHP\_PV\_Batt\_Gasbrenner\_TES\_Costs
* OptFlex\_CHP\_Batt\_Gasbrenner\_TES\_Costs
* OptFlex\_CHP\_PV\_Batt\_Gasbrenner\_TES\_EHeater\_Costs

### OptFlexAdhoc

Beinhaltet OptFlexCosts und eine Nachregeglung mit Tagespersistent-Prognose. VDI-Profile und PV2013 Kassel.

Stand 17.05.2016

* OptFlex\_PV\_Battery\_Costs\_Adhoc
* OptFlex\_CHP\_PV\_Batt\_Gasbrenner\_TES\_Costs\_AdHoc

### OptFlexAdhoc\_V

Beinhaltet OptFlexAdhoc mit der Option das die Optimierung nicht alle 10min, sondern einmal pro Tag durchgeführt wird.

* OptFlex\_CHP\_PV\_Batt\_Gasbrenner\_TES\_Costs\_AdhocV
* OptFlex\_PV\_Batt\_Costs\_AdhocV

### OptFlexCo2

Die Kostenfunktion bestimmt nicht mehr minimale Betriebskosten, sondern minimaler CO2 Produktion. Dazu wurden Grid\_Imp, PV2batt , PV2load und CHP2batt, CHP2load Emissionskoeffizienten zugewiesen und die Summe der Emission durch Stromerzeugung minimiert. Der Rest des optimierers ist gleich geblieben.

Eine neue Variable CO2Emis wird eingeführt.

## Zusammenfassung Eigenschaften (Stand 07.09.2016)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name (EFH)** | **OptZiel: Eigen E Costs C oder CO2** | **PV** | **Lasten** | **Analyse** | **Nachregelung** | **Prognose** | **OptZeitVariable** | **KPI** | **Plot** | **Jahressimulation** | **Preiszeitreihe** |
| OptFlex\_PV\_Battery | E | Kassel 2013 | VDI2009/ EFH | Nein | Nein (dummy) | Persistenz | nein | ja | ja | Nein | nein |
| OptFlex\_PV\_Battery\_Gasboiler\_TES | E | Kassel 2013 | VDI2009/ EFH | Nein | Nein | Perfekt | nein | nein | nein | Nein | nein |
| OptFlex\_PV\_Battery\_Gasboiler\_TES\_EHeater | E | Kassel 2013 | VDI2009/ EFH | nein | Nein | Perfekt | nein | Nein | nein | Nein | nein |
| OptFlex\_CHP\_Battery\_TES\_Gasboiler | E | Kassel 2013 | VDI2009/ EFH | nein | Nein | Perfekt | nein | nein | nein | Nein | nein |
| OptFlex\_CHP\_PV\_Batt\_Gasbrenner\_TES\_Costs | C  30min | Kassel 2013 | VDI2009/LPG/EFH/MFH(2013) | Ja | Nein (dummy) | Persistenz | nein | ja | ja | Ja | ja |
| OptFlex\_CHP\_PV\_Batt\_Gasbrenner\_TES\_EHeater\_Costs (EHeater, funktioniert nicht!) | C | Kassel 2013 | VDI2009/ EFH | nein | Nein (dummy) | Persistenz | nein | ja | ja | Nein | nein |
| OptFlex\_CHP\_Batt\_Gasbrenner\_TES\_Costs | C | Kassel 2013 | VDI2009/ EFH | nein | Nein | Perfekt | nein | Nein | nein | Nein | nein |
| OptFlex\_PV\_Battery\_Costs | C | Kassel 2013 | VDI2009/LPG EFH | nein | Nein | Persistenz | nein | Ja | ja | Ja | nein |
| OptFlex\_PV\_Battery\_Costs\_Adhoc | C | Kassel 2013 | VDI2009/ EFH | nein | Ja | Persistenz | nein | Ja | ja | Nein | nein |
| OptFlex\_CHP\_PV\_Batt\_Gasbrenner\_TES\_Costs\_AdHoc | C  30 min(Fehler siehe Sommer) | Kassel 2013 | VDI/LPG/EFH/MFH(2013) | Ja | Ja | Persistenz / 3T-Average | Nein | Ja | ja | Nein | ja |
| OptFlex\_CHP\_PV\_Batt\_Gasbrenner\_TES\_Costs\_AdhocV | C  30min | Kassel 2013 | VDI/LPG/EFH/MFH(2013) | Ja | Ja | Persistenz/3T-Average | Ja / 24h | Ja | ja | Nein | Ja (EPEX) |
| OptFlex\_PV\_Batt\_Costs\_AdhocV | C | Kassel 2013 | VDI2009/ EFH | nein | Ja | Persistenz | Ja / 24h | Ja | ja | Nein | nein |
| OptFlex\_CHP\_PV\_Batt\_Gasbrenner\_TES\_CO2 | CO2  30min | Kassel 2013 | VDI2009/LPG/EFH/MFH(2013) | Ja | Nein (dummy) | Persistenz | nein | ja | ja | Nein | ja |
| OptFlex\_CHP\_PV\_Batt\_Gasbrenner\_TES\_CO2\_AdHoc | CO2  30 min(Fehler siehe Sommer) | Kassel 2013 | VDI/LPG/EFH/MFH(2013) | Ja | Ja | Persistenz / 3T-Average | Nein | Ja | ja | Nein | ja |
| OptFlex\_CHP\_PV\_Batt\_Gasbrenner\_TES\_CO2\_AdhocV | C02 | Kassel 2013 | VDI/LPG/EFH/MFH(2013) | Ja | Ja | Persistenz/3T-Average | Ja / 24h | Ja | ja | Nein | Ja (EPEX) |
| OptFlexCo2/OptFlex\_CHP\_Batt\_Gasbrenner\_TES\_CO2 (PV=0) | CO2 | Kassel 2013 | VDI/LPG/EFH/MFH(2013) | Ja | Ja | Persistenz/3T-Average | Ja / 24h | Ja | ja | Nein | Ja (EPEX) |
| OptFlexTEST30min | Wie ”Costs”, zum testen der 30min on-off Regel | | | | | | | | | | |
| Optflex\_OGEMA | Version zur Einbindung in Ine-Ves Testhaushalt im Rahmen der Rest Schnittstelle und OGEMA | | | | | | | | | | |

# Idee Nachbar Einbindung in OptFlex

## Betrachte 3 Fälle

1. **Berechne Kosten konventionell:**

Hauhalt 1:

Wärmebedarf 🡪 Gaskosten durch Gasbrenner

Strombedarf 🡪 Netzbezug

Nachbar:

Wärmebedarf 🡪 Gaskosten durch Gasbrenner

Strombedarf 🡪 Netzbezug

1. **Berechne Kosten ohne Verbindung**

Haushalt 1

Wärmebedarf und Strombedarf von Prosumer durch MPC

Nachbar

Wärmebedarf 🡪Gaskosten durch Gasbrenner

Strombedarf 🡪 Netzbezug

1. **Berechne kombinierte Kosten**

Wärmebedarf (Haushalt 1+Nachbar) und Strombedarf (Haushalt 2)

Nachbar: Strombedarf 🡪 Netzbezug

## Vergleich

* Konventionell vs. MPC (Haushalt 1) 🡪 MPC führt zu Kostenvorteil (ja/nein)
* MPC (Haushalt 1 + Nachbar) vs. MPC (Haushalt 1) + Nachbar (Konventionell) 🡪 Können die Kosten des Nachbarn für Wärmebedarf gesenkt werden? Mehr Betriebstunden KWK?

# Filebeschreibung

Die einzelnen Ordner enthalten immer die gleichen Dateien (🡪Bessere Strukur des Programms möglich)

(„ORIG“-Files, sind File die nur als Vorlage dienten und von Diego Hidalgo stammen. Sie werden für die gesamte Berechnung nicht benötigt.)

## OptFlex\_MPC.py

Dieses File ist das “Mainfile”. Um das Programm zu starten ruft man dieses File auf und betätigt „run“. Hier wird festgelegt welchen Zeitraum man betrachten will, die MPC schleifen sind hier definiert und die meisten anderen Files werden hier aufgerufen. Hier wird auch noch der Forecast-Values vorbereitet (🡪 zukünftig auslagern!).

Änderung durch 30min commitment:

CHP['b\_CHP\_ini\_1'] = Result\_CHP\_End['CHP on\_off'][Result\_CHP\_End.index[timestep-2]]

CHP['b\_CHP\_ini\_2'] = Result\_CHP\_End['CHP on\_off'][Result\_CHP\_End.index[timestep-1]]

CHP['b\_CHP\_ini\_3'] = Result\_CHP\_End['CHP on\_off'][Result\_CHP\_End.index[timestep]]

## OptFlex\_inputvalues.py

Als erstes werden aus diesem File die Inputvalues eingelesen. Hier sind Parameter für die einzelnen Anlagen festgelegt und Kosten werden ebenfalls fixiert.

### **Funktionen**

*Inputvalues\*()*, wobei \*={Default\_MFH, Ohne\_Verg\_MFH, Default\_EFH, Ohne\_Verg\_EFH}, dieses Funktionen definieren Anlagenparameter und Kosten. Einfamilienhaus (EFH) und Mehrfamilienhaus (MFH), Momentane Kostensituation und Vergütung (Default), Ohne Zuschüsse und Vergütung (Ohne\_Verg)

*Input\_PV\_Load\_\*()*, wobei \*={VDE\_MFH, VDE\_EFH, LPG\_MFH, LPG\_EFH}, diese Funktionen lesen Files ein für PV, Wärmelast und Stromlast. Zwei Arten von Strom-Wärme-Lastprofilen sind implementiert. VDE4655(VDE) und Multi-LPG (siehe Masterarbeit Simon Drauz) (LPG).

*PreisProfil()* liegt aus einem File ein echtes variables Preisprofil des Spotmarkts ein.

*PreisProfilL0()* liest ein künstliches Preisprofil für Strompreise ein.

*PreisProfilER()* liest ein künstliches Profil für Einspeisevergütung ein.

Änderung durch 30min commitment

b\_CHP\_ini\_1 = 0

b\_CHP\_ini\_2 = 0

b\_CHP\_ini\_3 = 0

## OptFlex\_optimierer.py

Dann wird der Optimierer (Pyomo, CPlex) aufgerufen. Hier findet in jeden Schleifendurchlauf die Optimierung für den genannten Zeitraum statt.

Änderung durch 30min commitment:

Add:

* b\_CHP\_ini\_1 = {1: CHP['b\_CHP\_ini\_1']}
* b\_CHP\_ini\_2 = {1: CHP['b\_CHP\_ini\_2']}
* b\_CHP\_ini\_3 = {1: CHP['b\_CHP\_ini\_3']}
* model.b\_CHP\_ini\_1 = Param(model.I, initialize = b\_CHP\_ini\_1)
* model.b\_CHP\_ini\_2 = Param(model.I, initialize = b\_CHP\_ini\_2)
* model.b\_CHP\_ini\_3 = Param(model.I, initialize = b\_CHP\_ini\_3)

def CHP\_min\_ON\_rule(model,k):# only implemented for three steps Delta\_T !!!TMK

if (k==0):

return model.b\_CHP\_ini\_1[1]+model.b\_CHP\_ini\_2[1]+model.b\_CHP\_ini\_3[1] + T\_CHP\_on\*model.b\_CHP\_up[k] <= 3

elif (k==1):

return model.b\_CHP\_ini\_2[1]+model.b\_CHP\_ini\_3[1]+model.b\_CHP\_on[k-1] + T\_CHP\_on\*model.b\_CHP\_up[k] <= 3

elif (k==2):

return model.b\_CHP\_ini\_3[1]+model.b\_CHP\_on[k-1]+model.b\_CHP\_on[k-2] + T\_CHP\_on\*model.b\_CHP\_up[k] <= 3

t\_on\_range = range(k+1,min(k+T\_CHP\_on+1,Np))

return sum(model.b\_CHP\_on[i] for i in t\_on\_range) - T\_CHP\_on\*model.b\_CHP\_up[k] >= 0

model.CHP\_min\_ON = \

Constraint(model.K, rule=CHP\_min\_ON\_rule)

def CHP\_min\_OFF\_rule(model,k): #only implemented for three steps Delta\_T !!!TMK

if (k==0):

return T\_CHP\_off - (model.b\_CHP\_ini\_1[1]+model.b\_CHP\_ini\_2[1]+model.b\_CHP\_ini\_3[1]) + \

T\_CHP\_off\*model.b\_CHP\_down[k] <= 3

elif (k==1):

return T\_CHP\_off - (model.b\_CHP\_ini\_2[1]+model.b\_CHP\_ini\_3[1]+model.b\_CHP\_on[0]) + \

T\_CHP\_off\*model.b\_CHP\_down[k] <= 3

elif (k==2):

return T\_CHP\_off - (model.b\_CHP\_ini\_3[1]+model.b\_CHP\_on[0]+model.b\_CHP\_on[1]) + \

T\_CHP\_off\*model.b\_CHP\_down[k] <= 3

t\_off\_range = range(k+1,min(k+T\_CHP\_off+1,Np))

return T\_CHP\_off - sum(model.b\_CHP\_on[i] for i in t\_off\_range) - T\_CHP\_off\*model.b\_CHP\_down[k] >= 0

model.CHP\_min\_OFF = \

Constraint(model.K, rule=CHP\_min\_OFF\_rule)

## OptFlex\_plotting.py

In “plotting” warden nach einer Vorlage von “OptIn-Tool (Jan vonAppen)” für den ausgewählten Zeitraum die Verläufe/Fahrpläne/Erzeugung und Verbrauch geplottet. Output sind die Bilder als \*.png in „RESULT“.

## OptFlex\_KPI.py

Die festgelegten KPI werden hier berechnet und sind jeweils die Summe über das ausgewählte Zeitintervall. Ouput ist ein File mit den KPIs als .csv in „RESULT“

## OptFlex\_Analyse.py

Liest die KPI.csv ein und plottet die Werte zusammen in Balkendiagramme. Output sind .png. in „OUTPUT“

## OptFlex\_nachregelung.py

Hier findet für die „Ad\_Hoc“ Rechnungen die Nachregelung statt. Die optimierten Werte werden auf Abweichungen kontrolliert und an die realen Werte angepasst. Die korrigierten Werte werden zurückgegeben, damit mit Ihnen weitergerechnet werden kann.

Die Ad\_HocV Nachreglung kann variable optimieren und nachregeln. Dabei gilt, es wird auf jeden nach PrognoseHorizont optimiert und wenn einer der SOCs mehr als „procent“, z.B. 10% von Prognose abweicht. (31.5.2016)

## GesamtAnalyse.py

Liegt „über“ den anderen Directories und analysiert alle drei Regelungsvarianten gemeinsam.

Pfade zu Input-Daten beachten!

## OptFlex\_forecast

Zwei Methoden sind abgebildet:

* Tagespersistenz für alle Eingangszeitreihen (heute ist es wie gestern.9

# Tage Mittelwert, die Methode mindert die extremen Peaks ab.

# Sensitivitätsanalyse (Paper)

## Horizont (2, 15, 36, 72, 144) 🡪 (20min, 2.5h, 6h, 12h, 24h)

* „Übergang“, 144 funktioniert Optimierung gar nicht.
* **Ergebnis:** Favorit 36 also alle 6 Stunden (AdhocV), da Laufzeit vs. Gesamtkostenminimierung am besten.

## Optimierungszeitschritte bzw. off-Set SOC optimiert vs planung

* Schritte werden festgelegt, nach Abstand zu SOCs aus Optimierung.

## Feed-In Tarif

* + CHP (8 c/kWh: eingespeist, 4 c/kWh igenverbrauch) (8/4; 0/0; 0/8; 8/0)
  + PV (2016: 12,31 c/kWh, 2015: 12,40, 2014, 12,60) 12,31; 6; 0

## Gaspreis 5c/kWh, 10 c/kWh, 2.5 c/kWh

## Strompreis 2000 13,94 c/kWh, 2015, 28,69 c/kWh, 35 c/kWh, 40 c/kWh

## Forecast (Persistenz, Running average, letzten drei Tage)

# Dauert länger (Sommer, 144 470 vs 280 s)

* Übergang sind Kosten viel höher, CHP wird öfter angeschaltet im Sommer und Übergang
* Batterie wird im Winter und im Übergang kaum benutzt

# **Ergebnis:** Tagespersistenz ist besser bei adhocV, da Gesamtenergiebedarf besser abgeschätzt wird, die einzelnen Peaks sind nicht so relevant. Vor allem sichtbar bei „Übergangstagen“.

## Mit und ohne Nachregelung

* 100% Eigenverbrach in der Optimierung verliert ?% durch Prognosefehler

## CHP (kontinuierlich vs. On-off)

## (Objective: Kosten, Eigenverbrauch, Strompreise(?)) 🡪 eventuell schon abgebildet

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Größe** |  |  |  |  |  |  |  |  |  |
| Horizon | 2 | 15 | 36 | 72 | 144 (nicht für Übergang möglich) |  |  |  |  |
| Optimierungs Schrittweite | 10 min | Horizont + wenn 10% SOC Differenz |  |  |  |  |  |  |  |
| CHP Feed-In | +  +  + | 0  0  0 |  |  |  |  |  |  |  |
| CHP Eigenverbrauch |  |  |  |  |  |  |  |
| PV Feed-In | Var\_beliebig |  |  |  |  |  |  |
| Gaspreis |  |  |  |  |  |  |  |  |  |
| Strompreis | fix | Var\_Börse | Var\_beliebig |  |  |  |  |  |  |
| Forecast | Persistenz | 3 Tage Average |  |  |  |  |  |  |  |
| Nachregelung | Ohne | Mit |  |  |  |  |  |  |  |
| Objective | Kosten | CO2 |  |  |  |  |  |  |  |
| CHP | On/off | Cont. |  |  |  |  |  |  |  |

# Berechnung KPIs

## Beschreibung der KPIs

|  |  |  |  |
| --- | --- | --- | --- |
| **Key Performance Indicators** | **Beschreibung** | |  |
| **Gleichung** | **Einheit** | **Diego check** |
| Total grid export (kWh) | P\_exp\_k x dt = E\_exp\_k | |  |
| Total grid import (kWh) | P\_exp\_k x dt = E\_imp\_k | |  |
| Autarkie elektrisch (%) | (P\_PV\_sc\_k + P\_CHP\_el\_sc\_k) : E\_Load\_k | |  |
| Bil. Autarkie elektrisch (%) | (P\_PV\_k + P\_CHP\_el\_k) : E\_Load\_k | |  |
| Eigenverbrauch elektrisch (%) | (P\_PV\_sc\_k + P\_CHP\_el\_sc\_k) : (P\_PV\_k + P\_CHP\_el\_k) | |  |
| Bil. Eigenverbrauch elektrisch (%) | (P\_PV\_k + P\_CHP\_el\_k) : (P\_PV\_k + P\_CHP\_el\_k) | |  |
| Total efficiency (%) | ? | |  |
| Mean time (seg) | ? | |  |
| Mean CHPon time [hours] | b\_CHP\_k x dt | |  |
| CHP export (kWh) | P\_CHP\_exp\_k x dt = E\_CHP\_exp\_k | |  |
| PV export (kWh) | P\_PV\_exp\_k x dt = E\_PV\_exp\_k | |  |
| CHP electric self-consumption (kWh) | P\_CHP\_el\_sc\_k / P\_CHP\_el\_k  With P\_CHP\_el\_SC = CHP2batt+CHP2load+CHP2EH | |  |
| PV electric self-consumption (kWh) | P\_PV\_el\_sc\_k / P\_PV\_k  With P\_PV\_el\_sc = PV2batt+PV2load+PV2EH | |  |
| Total electrical consumption (kWh) | P\_el\_Load\_k x dt = E\_el\_Load\_k | |  |
| Total thermal consumption (kWh) | P\_th\_Load\_k x dt = E\_th\_Load\_k | |  |
| Total CHP electrical production (kWh) | P\_CHP\_el\_k x dt = E\_CHP\_el\_k | |  |
| Total CHP thermal production (kWh) | P\_CHP\_th\_k x dt = E\_CHP\_th\_k | |  |
| Total gas boiler thermal production (kWh) | P\_aux\_th\_k x dt = E\_aux\_th\_k | |  |
| Total e-heater thermal production (kWh) | P\_EH\_th\_k x dt = E\_EH\_th\_k | |  |
| Battery cycles | P\_bat\_dis\_k dt + P\_bat\_ch\_k dt )/ 2/Cap\_batt | |  |
| CHP on/off |  | |  |
| CHP mean on time [min] |  | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Key Performance Indicators -Costs** | **Beschreibung** | |  |
| **Gleichung** | **Einheit** | **Diego check** |
| Total costs |  | |  |
| Total costs Gas |  | |  |
| Total costs Strom |  | |  |
| Total costs PV feed-in |  | |  |
| Total costs PV   * Battery |  | |  |
| Total costs PV   * Load |  | |  |
| Total costs CHP feed-in |  | |  |
| Total costs CHP on/off |  | |  |
| Total costs CHP   * Battery |  | |  |
| Total costs CHP   * Load |  | |  |
| Total costs CHP eigen |  | |  |

***Abbreviations***

PV Photovoltaik

CHP Combined heat and power

TES Thermal Energy Storage

Batt Battery

Aux Auxiliary gas boiler

DHW Domestic Hot Water

ZX Electricity meter

GZ Gas Meter

WZ Heat Meter

T Temperatur Sensor

PCC Point of Common Coupling

MPC Model Predictive Control