Storage check with basic flexibility use case

Note of 9 July 2019 by Dr. Michael Stöhr, B.A.U.M. Consult GmbH

Setting

Let an area have an electricity demand d(t) which is met by a generation g(t) with zero marginal costs, e.g. a wind park or PV plants or a combination of them. Let d(t) and g(t) be step-like functions of t, i.e. constant within each time-interval dt. Let further the sum Σ d(t) equal the sum Σ g(t), but let d(t) and g(t) differ for at least two time-intervals. As an example, d(t) and g(t) let be expressed by the following time series:

g(t) = [1, 1, 1, 0, 0, 0, 0, 0, 0, 0]

d(t) = [0, 0, 0, 1, 1, 1, 0, 0, 0, 0]

where each value in the series represents the function value for a time-interval.

Flexibility options to match d(t) and g(t)

In order to match d(t) and g(t), combinations of the following flexibility options are taken into account:

1. the generation g(t) is curtailed if g(t) > d(t), described by a curtailment function curt(t)
2. a storage is charged if g(t) > d(t), described by a charging function sc(t)
3. a backup generation b(t) is activated if g(t) < d(t), described by a backup generation function b(t)
4. a storage is discharged if g(t) < d(t), described by a storage discharging function sd(t)

If all functions are defined such that they take only non-negative values, conservation of energy requires that

g(t) + b(t) + sd(t) = d(t) + curt(t) + sc(t)   for all t

Summing up over t and considering that Σ d(t) equals Σ g(t) it follows

Σ s(t) = Σ b(t) – Σ curt(t)

where the storage flow function s(t) is the difference sc(t) - sd(t). In case of no storage or a storage without any losses, Σ s(t) = 0 and the total backup generation E_{backup} = Σ b(t) equals the total curtailment E_{curt} = Σ curt(t). Given g(t) and d(t) as above, E_{curt} = 3. In case of a storage with losses Σ s(t) > 0.

Costs of flexibility

Curtailment does not change the costs as the marginal costs of g are assumed to be zero. For backup generation, additional costs c per unit of electricity have to be taken into account. In case of no storage or a storage without any losses, the additional costs of flexibility, C_{flex}, are the product of the maximum sum of backup generation, $E_{backup,\ max}$ = Σ b(t), and c:

$$C_{flex,without\ storage} = c \cdot E_{backup,max}$$

This is the maximum value for the costs of flexibility. If c = 1, it equals 3.

A storage is only activated if it reduces the cost of flexibility below that value. If this the case, the sum of backup generation equals the sum of curtailment + the losses in the storage, E_{loss}, and the costs of flexibility are:

$$C_{flex,with\ storage} = epc \cdot E_{max} + c \cdot E_{backup}$$

Where epc denotes the equivalent periodical costs (depreciation + fixed variable costs) of the storage for the considered period (here: 10 time units) and E_{max} = max (E(t)) the maximum energy in the storage in one or more of the considered time units. If the storage is designed such that its state of charge can vary between 0 and 1, E_{max} is identical to the nominal storage capacity.

A storage is only minimising the costs of flexibility if functions curt(t), b(t) and s(t) can be found such that

$$C_{flex,with\ storage} < C_{flex,without\ storage}$$

i.e. the following check function ctrl is not negative:

$$ctrl := E_{backup,max} - \frac{epc}{c} \cdot E_{max} - E_{backup} \geq 0$$

If c = 1:

$$ctrl = 3 - epc \cdot E_{max} - E_{backup} \geq 0$$

The maximum value ctrl can take is 3. This happens if a storage has no costs and backup generation can be totally avoided by using the storage. If no storage is activated ctrl = 0.

Optimum combination of flexibility options for storage without any losses

The optimum combination of the different flexibility options is searched with a python programme using oemof 0.3.1. The following constraints are set:

- The energy stored in the storage before the first timestep must equal the energy stored in the last timestep: E(-1) = E(10).
- Energy must be conserved in the entire system and in the storage subsystem.

The problem is implemented in the programme storage_check_11.py with the following parameter settings:

- storage charging efficiency = storage discharging efficiency = 1
- storage self-discharge = 0 per timestep
- c = 1 -> C_{flex} = 3
- epc is varied

The results are shown in the following figures.

There are two distinct ranges of epc with respectively different optimum solution:

1. For epc < 1, a storage is activated which completely balances the difference between generation and demand. No backup generation is needed.
2. For epc > 1 no storage is activated. A combination of curtailment and backup generation completely balances the difference between generation and demand.

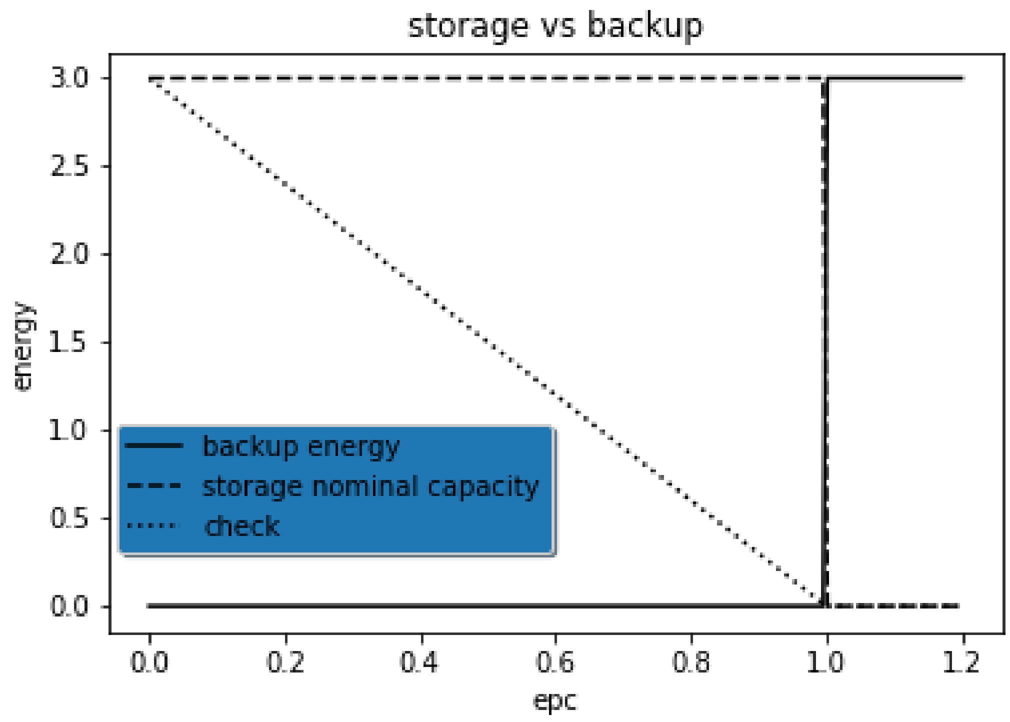

Figure 1: Storage vs backup for different storage costs in case of storage without any losses

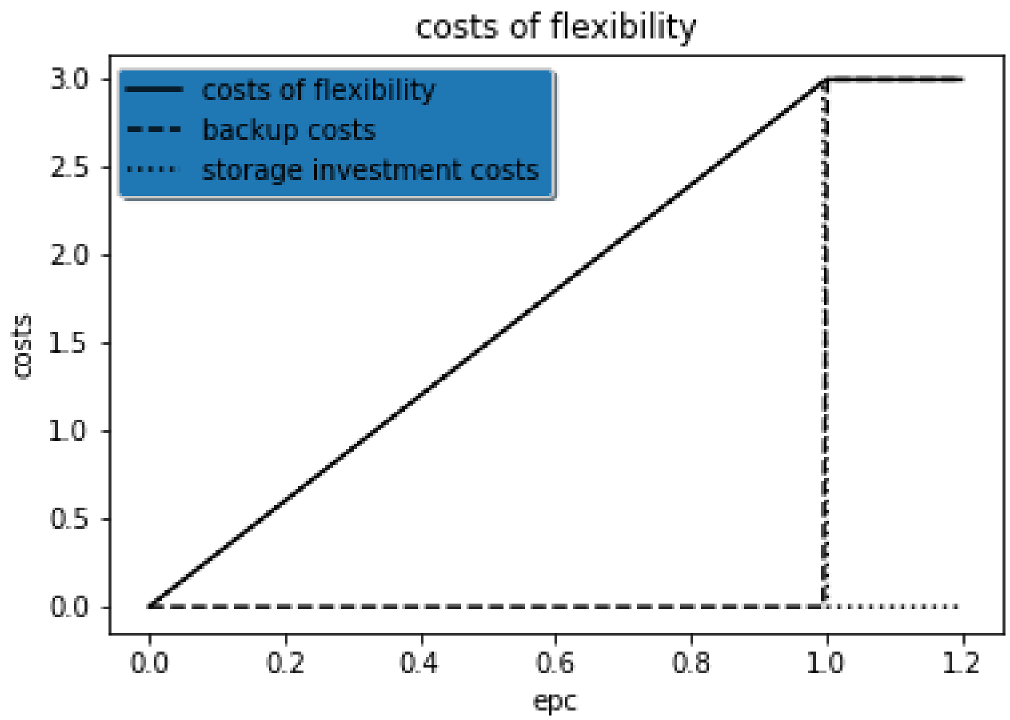

Figure 2: Costs of flexibility for different storage costs in case of storage without any losses

The cost of flexibility rise with epc until they reach the value of the maximum backup generation costs. The check function ctrl decreases with epc from its maximum possible value 3 to 0.

Optimum combination of flexibility options for storage with losses

Now the parameters are set as follows:

- storage charging efficiency = storage discharging efficiency = 0.95
- storage self-discharge = 0.1 per timestep
- c = 1 -> C_{flex} = 3
- epc is varied

The results are shown in the following figures.

There are now four distinct ranges of epc which lead to respectively different optimum solutions which differ in the size of the storage and the required backup generation. The backup generation is never zero because it needs to compensate at least the losses in the storage.

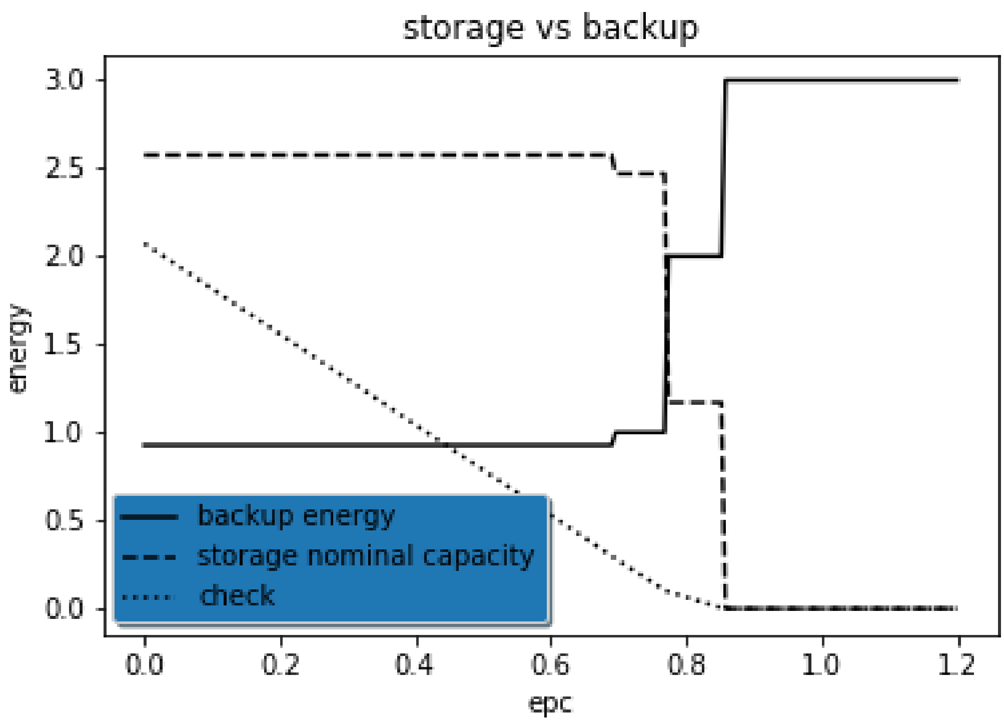

Figure 3: Storage vs backup for different storage costs in case of storage with losses

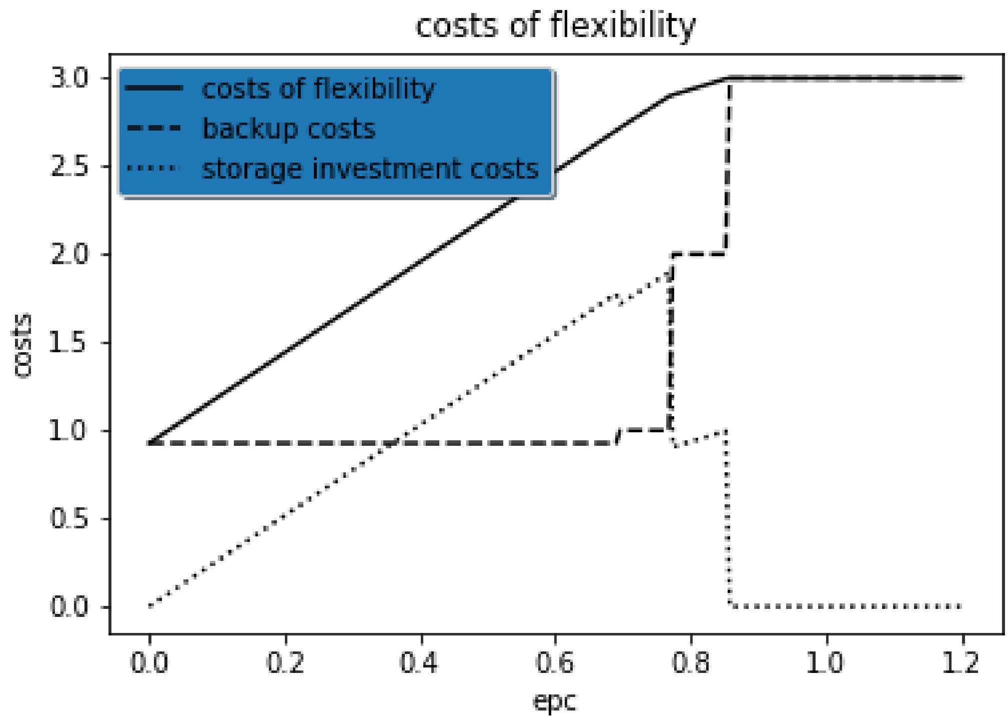

Figure 4: Costs of flexibility for different storage costs in case of storage with losses

Acknowledgements

This note has been written within the project Generalized Operational FLEXibility for Integrating Renewables in the Distribution Grid (GOFLEX) which receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731232.