Energy Analysis Methodology

Grid Savings Simulation for a PV + Battery System

Home Assistant Data Science Project

April 16, 2026

Contents

1 Overview

The script `heating_pv/energy_analysis.py` answers two questions:

1. How much electricity did the HVAC loads (BWWP heat pump, Splitklima Wohnzimmer, Splitklima Arbeitszimmer) consume over a given time period?
2. How much *more* electricity would have been required from the public grid if the photovoltaic (PV) generators and the battery storage system had not been installed?

The analysis reads from a SQLite database populated by `heating_pv/extract_energy_5min.py`. Two database variants are supported (see Section 9): the default `energy_hourly.db` (1-hour

resolution, permanent history) and the optional `energy_5min.db` (5-minute resolution, ~74-day retention on this installation).

2 System Architecture

The installation consists of the following components measured in the database:

Symbol	Description	Database column
$E_{\text{from}}[t]$	Energy drawn from grid	`grid_from_grid_kwh`
$E_{\text{to}}[t]$	Energy fed into grid	`grid_to_grid_kwh`
$E_{\text{chg}}[t]$	Energy charged into battery	`battery_charge_kwh`
$E_{\text{dis}}[t]$	Energy discharged from batt.	`battery_discharge_kwh`
$E_{\text{pv1}}[t]$	Solar production, Fronius Symo 6 kW	`solar_kwh`
$E_{\text{pv2}}[t]$	Solar production, second generator	`solar2_kwh`
$E_{\text{bwwp}}[t]$	BWWP heat pump consumption	`bwwp_kwh`
$E_{\text{wz}}[t]$	Splitklima Wohnzimmer	`splitklima_wz_kwh`
$E_{\text{az}}[t]$	Splitklima Arbeitszimmer	`splitklima_az_kwh`

All values are energies (kWh) *per time interval*. Index t runs over every interval (1 hour or 5 minutes, depending on the chosen data source) in the selected time range. The values are derived from the cumulative `change` statistic in Home Assistant's long-term statistics recorder, or from the first difference of raw cumulative counters in InfluxDB.

3 House Consumption Reconstruction

The total electricity consumption of the house $E_{\text{house}}[t]$ is not measured directly by a single meter. It is instead derived from the node-balance equation at the AC bus:

$$\underbrace{E_{\text{from}}[t] + E_{\text{pv1}}[t] + E_{\text{pv2}}[t] + E_{\text{dis}}[t]}_{\text{sources}} = \underbrace{E_{\text{house}}[t] + E_{\text{to}}[t] + E_{\text{chg}}[t]}_{\text{sinks}} \tag{1}$$

Solving for house consumption:

$$\boxed{E_{\text{house}}[t] = E_{\text{from}}[t] + E_{\text{pv1}}[t] + E_{\text{pv2}}[t] + E_{\text{dis}}[t] - E_{\text{to}}[t] - E_{\text{chg}}[t]} \tag{2}$$

Measurement noise. Because each quantity is measured by a different sensor (Victron ESS, Fronius inverter, Shelly energy meters) with different update intervals and quantisation steps, the result can occasionally be slightly negative. Such values are clipped to zero before further analysis.

4 HVAC Consumption

The electricity consumed by the three HVAC loads over the analysis period $[t_0, t_1]$ is:

$$E_{\text{HVAC}} = \sum_{t=t_0}^{t_1} \big(E_{\text{bwwp}}[t] + E_{\text{wz}}[t] + E_{\text{az}}[t]\big) \tag{3}$$

These meters (ShellyPro EM3) have sub-Wh resolution, so the sum is accurate to within a few Wh regardless of the time-interval length.

5 Simulation Scenarios

Four scenarios are evaluated; their results are printed side by side for easy comparison.

5.1 Scenario A — Baseline: No Solar, No Battery

Without any generation or storage, the house draws everything it needs from the public grid:

$$E_{\text{grid,A}}[t] = E_{\text{house}}[t] \tag{4}$$

Total grid draw: $G_A = \sum_t E_{\text{house}}[t]$. There is no feed-in. This is the worst-case reference scenario.

5.2 Scenario B — Solar Only (No Battery)

Solar production reduces the net grid draw at each interval, but any surplus that exceeds instantaneous consumption is fed into the grid rather than being stored:

$$E_{\text{pv}}[t] = E_{\text{pv1}}[t] + E_{\text{pv2}}[t] \tag{5}$$

$$E_{\text{grid,B}}[t] = \max\bigl(0,\ E_{\text{house}}[t] - E_{\text{pv}}[t]\bigr) \tag{6}$$

$$E_{\text{feed,B}}[t] = \max\bigl(0,\ E_{\text{pv}}[t] - E_{\text{house}}[t]\bigr) \tag{7}$$

Total grid draw: $G_B = \sum_t E_{\text{grid,B}}[t]$.

5.3 Scenario C — Solar + Battery (Re-simulation)

A battery is added. At each interval the controller first satisfies the house deficit from the battery, or charges the battery with any solar surplus, before drawing from or exporting to the grid.

Battery model. Let $S[t]$ denote the state of charge (SOC) in kWh, C the usable capacity, and η the round-trip storage efficiency. The efficiency is applied at charge time: of every kWh that flows into the battery, only η kWh is recoverable. Discharge is lossless with respect to the stored energy.

Surplus step ($E_{\text{pv}}[t] \geq E_{\text{house}}[t]$):

$$\Delta_{\text{net}} = E_{\text{pv}}[t] - E_{\text{house}}[t] \tag{8}$$

$$\Delta_{\text{chg}} = \min\left(\Delta_{\text{net}},\ \frac{C - S[t-1]}{\eta}\right) \tag{9}$$

$$S[t] = \min\bigl(C,\ S[t-1] + \eta\,\Delta_{\text{chg}}\bigr) \tag{10}$$

$$E_{\text{feed,C}}[t] = \max\bigl(0,\ \Delta_{\text{net}} - \Delta_{\text{chg}}\bigr) \tag{11}$$

$$E_{\text{grid,C}}[t] = 0 \tag{12}$$

The division by η in the second line ensures that Δ_{chg} is the energy flowing *into the charger*; multiplying by η yields the energy actually stored.

Deficit step ($E_{\text{pv}}[t] < E_{\text{house}}[t]$):

$$\Delta_{\text{def}} = E_{\text{house}}[t] - E_{\text{pv}}[t] \tag{13}$$

$$\Delta_{\text{dis}} = \min\bigl(\Delta_{\text{def}},\ S[t-1]\bigr) \tag{14}$$

$$S[t] = S[t-1] - \Delta_{\text{dis}} \tag{15}$$

$$E_{\text{grid,C}}[t] = \max\bigl(0,\ \Delta_{\text{def}} - \Delta_{\text{dis}}\bigr) \tag{16}$$

$$E_{\text{feed,C}}[t] = 0 \tag{17}$$

Initial SOC. The simulation starts with $S[0] = 0.5\,C$ (half-charged). Over a period of several months the exact initial condition has negligible influence on the integrated totals.

Parameters.

- Default capacity: $C = 14\,\text{kWh}$
- Default round-trip efficiency: $\eta = 0.85$ (85 %)
- Both values are command-line arguments: `--battery-capacity` and `--battery-efficiency`.

5.4 Scenario D — Actual Measured

The actual grid draw $E_{\text{from}}[t]$ and feed-in $E_{\text{to}}[t]$ directly from the database. Scenario C should approximate Scenario D; any difference reflects real control-strategy effects (e.g. time-of-use optimisation by the Victron ESS) or sensor drift.

6 Self-Sufficiency Ratio

The self-sufficiency ratio (shown as a percentage in the output) is defined as the fraction of house consumption *not* supplied by the grid:

$$\text{SS} = \left(1 - \frac{G_X}{\sum_t E_{\text{house}}[t]}\right) \times 100\,\% \tag{18}$$

where G_X is the total grid draw for scenario $X \in \{A, B, C, D\}$.

7 Grid Savings

The benefit of the solar + battery system is reported at three levels:

$$\text{Solar savings} = G_A - G_B \tag{19}$$

$$\text{Battery savings} = G_B - G_C \quad \text{(on top of solar only)} \tag{20}$$

$$\text{Total simulated} = G_A - G_C \tag{21}$$

$$\text{Total actual} = G_A - G_D \tag{22}$$

A positive value means the technology reduced grid draw. The difference $G_C - G_D$ quantifies how well the simulation reproduces reality.

8 Command-Line Reference

Run from the project root directory (`~/python_projects/home-assistant`):

```
# HA hourly data (default) --- full database range
uv run python heating_pv/energy_analysis.py

# HA hourly data --- specific date range (both dates inclusive)
uv run python heating_pv/energy_analysis.py \
    --start 2025-10-01 --end 2026-03-31

# InfluxDB 5-minute data --- recent data, higher resolution
uv run python heating_pv/energy_analysis.py --source influxdb

uv run python heating_pv/energy_analysis.py --source influxdb \
    --start 2026-03-15 --end 2026-04-14

# Custom battery parameters
uv run python heating_pv/energy_analysis.py \
    --battery-capacity 10 --battery-efficiency 0.90

# All options
uv run python heating_pv/energy_analysis.py --help
```

Options

Option	Default	Description
`--source {ha,influxdb}`	`ha`	Data source (see Section 9)
`--start YYYY-MM-DD`	earliest DB record	Analysis start date (inclusive)
`--end YYYY-MM-DD`	latest DB record	Analysis end date (inclusive)
`--battery-capacity KWH`	14.0	Usable battery capacity (kWh)
`--battery-efficiency FRAC`	0.85	Round-trip efficiency (0–1)
`--db PATH`	source default	Override SQLite database path

9 Data Sources

Two data sources are supported, selected with `--source`. Both are populated by `heating_pv/extract_energy_5` using the same `--source` flag.

9.1 HA Long-Term Statistics (default, `--source ha`)

Home Assistant's `recorder/statistics_during_period` WebSocket API returns hourly `change` statistics permanently for every entity that has a `state_class`. This is the recommended source for any analysis spanning more than a few weeks.

- Resolution: 1 hour per row
- Retention: permanent (full entity lifetime)
- Output: `data/energy_hourly.db`, table `energy_hourly`

Note: HA also maintains a `statistics_short_term` table at 5-minute resolution, but this is purged after approximately 10 days by default. It is therefore unsuitable for multi-month analysis.

9.2 InfluxDB 5-Minute Data (`--source influxdb`)

The InfluxDB add-on records every raw sensor state update continuously. The extraction script queries the raw cumulative counter values and reconstructs per-interval energy by:

1. Querying raw cumulative values in 30-day chunks to avoid timeouts.
2. Resampling to a regular 5-minute UTC grid (`resample("5min").last()`).
3. Forward-filling gaps caused by brief sensor outages.
4. Computing the first difference (`.diff()`) to obtain energy per interval.
5. Clipping negative values to zero (counter resets, sensor noise).

- Resolution: 5 minutes per row
- Retention: ~74 days on this installation (InfluxDB retention policy)
- Output: `data/energy_5min.db`, table `energy_5min`

9.3 Which Source to Use?

Time range	Recommended source
$>\sim$74 days	`--source ha` (only option with full history)
$\leq\sim$74 days, hourly precision sufficient	`--source ha` (simpler)
$\leq\sim$74 days, 5-minute resolution needed	`--source influxdb`

10 Limitations and Caveats

1. **Grid sensor resolution.** The Victron energy counters update in steps of 0.1 kWh (= 100 Wh), so the grid draw and feed-in figures have a quantisation uncertainty of roughly ± 0.05 kWh per interval.
2. **Reconstruction noise.** $E_{\text{house}}[t]$ is derived from four independent sensors. Occasional small negative values are evidence of sensor timing skew and are clipped to zero; the resulting error accumulates to at most a few kWh over a 6-month period.
3. **Battery simulation vs. reality.** The simulated battery uses a greedy solar-first dispatch: surplus solar charges the battery immediately. The actual Victron ESS may use a time-of-use or dynamic feedback strategy, which can produce a different (sometimes better) result. Scenario C is therefore an approximation of what a *simple* battery controller would achieve.
4. **Initial SOC.** The simulation starts at 50 % SOC. For analysis periods longer than one week, the impact on total grid draw is below 0.1 %.
5. **Hourly vs. 5-minute simulation accuracy.** The battery dispatch simulation is more accurate at finer time resolution. At 1-hour granularity, intra-hour solar surpluses and deficits cancel out in the same bin, so the simulation slightly overestimates the battery benefit compared to the 5-minute source.