

# Power Module

The power module contains a set of functions to calculate quantities of interest for power production and power quality.

Power Module

## Characteristics

This module contains functions for calculating electrical power metrics from measured voltage and current data. It supports both direct current (DC) and alternating current (AC) calculations, including instantaneous frequency analysis for AC signals and power calculations for three-phase AC systems. The calculations can accommodate both line-to-neutral and line-to-line voltage measurements and offer flexibility in output formats, allowing results to be saved as either pandas DataFrames or xarray Datasets.

### Functions:

`instantaneous_frequency`: Calculates the instantaneous frequency of a measured voltage signal over time.

`dc_power`: Computes the DC power from voltage and current measurements, providing both individual channel outputs and a gross power calculation.

`ac_power_three_phase`: Calculates the magnitude of active AC power for three-phase systems, considering the power factor and voltage measurement configuration (line-to-neutral or line-to-line).

<code>instantaneous_frequency</code>	Calculates instantaneous frequency of measured voltage
<code>dc_power</code>	Calculates DC power from voltage and current
<code>ac_power_three_phase</code>	Calculates magnitude of active AC power from line to neutral voltage and current

`mhkit.power.characteristics.instantaneous_frequency`(*measured\_voltage*: [Series](#) | [DataFrame](#) | [DataArray](#) | [Dataset](#), *time\_dimension*: *str* = "", *to\_pandas*: *bool* = *True*)→ [DataFrame](#) | [Dataset](#) [\[source\]](#)

Calculates instantaneous frequency of measured voltage

- Parameters:
- `measured_voltage` (*pandas Series*, *pandas DataFrame*, *xarray DataArray*,) – or *xarray Dataset* Measured voltage (V) indexed by time
  - `time_dimension` (*string (optional)*) – Name of the *xarray* dimension corresponding to time. If not supplied, defaults to the first dimension. Does not affect pandas input.
  - `to_pandas` (*bool (Optional)*) – Flag to save output to pandas instead of *xarray*. Default = *True*.

Returns:

**frequency** (*pandas DataFrame* or *xarray Dataset*) – Frequency of the measured voltage (Hz) indexed by time with signal name columns

`mhkit.power.characteristics.dc_power`(*voltage*: [Series](#) | [DataFrame](#) | [DataArray](#) | [Dataset](#), *current*: [Series](#) | [DataFrame](#) | [DataArray](#) | [Dataset](#), *to\_pandas*: *bool* = *True*)→ [DataFrame](#) | [Dataset](#) [\[source\]](#)

Calculates DC power from voltage and current

- Parameters:**
- **voltage** (*pandas Series, pandas DataFrame, xarray DataArray, or xarray Dataset*) – Measured DC voltage [V] indexed by time
  - **current** (*pandas Series, pandas DataFrame, xarray DataArray, or xarray Dataset*) – Measured three phase current [A] indexed by time
  - **to\_pandas** (*bool (Optional)*) – Flag to save output to pandas instead of xarray. Default = True.
- Returns:** **power\_dc** (*pandas DataFrame or xarray Dataset*) – DC power [W] from each channel and gross power indexed by time

---

**mhkit.power.characteristics.ac\_power\_three\_phase**(*voltage: Series | DataFrame | DataArray | Dataset, current: Series | DataFrame | DataArray | Dataset, power\_factor: float, line\_to\_line: bool = False, to\_pandas: bool = True*) → *DataFrame | Dataset* [\[source\]](#)

Calculates magnitude of active AC power from line to neutral voltage and current

- Parameters:**
- **voltage** (*pandas Series, pandas DataFrame, xarray DataArray, or xarray Dataset*) – Measured DC voltage [V] indexed by time
  - **current** (*pandas Series, pandas DataFrame, xarray DataArray, or xarray Dataset*) – Measured three phase current [A] indexed by time
  - **power\_factor** (*float*) – Power factor for the efficiency of the system
  - **line\_to\_line** (*bool (Optional)*) – Set to true if the given voltage measurements are line\_to\_line
  - **to\_pandas** (*bool (Optional)*) – Flag to save output to pandas instead of xarray. Default = True.
- Returns:** **power\_ac** (*pandas DataFrame or xarray Dataset*) – Magnitude of active AC power [W] indexed by time with Power column

## Quality

This module contains functions for calculating various aspects of power quality, particularly focusing on the analysis of harmonics, interharmonics and distortion in electrical power systems. These functions are designed to assist in power quality assessments by providing tools to analyze voltage and current signals for their harmonic and interharmonic components based on the guidelines and methodologies outlined in IEC 61000-4-7:2008 ED2 and in IEC 62600-30:2018 ED1.

Functions in this module include:

- **harmonics**: Calculates the harmonics from time series of voltage or current. This function returns the amplitude of the time-series data harmonics indexed by the harmonic frequency, aiding in the identification of harmonic distortions within the power system.
- **harmonic\_subgroups**: Computes the harmonic subgroups as per IEC 61000-4-7 standards. Harmonic subgroups provide insights into the distribution of power across different harmonic frequencies, which is crucial for understanding the behavior of non-linear loads and their impact on the power quality.
- **total\_harmonic\_current\_distortion (THCD)**: Determines the total harmonic current distortion, offering a summary metric that quantifies the overall level of harmonic distortion present in the current waveform. This metric is essential for assessing compliance with power quality standards and guidelines.
- **interharmonics**: Identifies and calculates the interharmonics present in the power system. Interharmonics, which are frequencies that occur between the fundamental and harmonic frequencies, can arise from various sources and potentially lead to power quality issues.

<code>harmonics</code>	Calculates the harmonics from time series of voltage or current based on IEC 61000-4-7.
<code>harmonic_subgroups</code>	Calculates the harmonic subgroups based on IEC 61000-4-7
<code>total_harmonic_current_distortion</code>	Calculates the total harmonic current distortion (THC) based on IEC/TS 62600-30
<code>interharmonics</code>	Calculates the interharmonics from the harmonic_amplitudes of current

---

**mhkit.power.quality.harmonics**(*signal\_data: Series | DataFrame | DataArray | Dataset, freq: float | int, grid\_freq: int, to\_pandas: bool = True*)→ DataFrame | Dataset [\[source\]](#)

Calculates the harmonics from time series of voltage or current based on IEC 61000-4-7.

- Parameters:**
- **signal\_data** (*pandas Series, pandas DataFrame, xarray DataArray, or xarray Dataset*) – Time-series of voltage [V] or current [A]
  - **freq** (*float or int*) – Frequency of the time-series data [Hz]
  - **grid\_freq** (*int*) – Value indicating if the power supply is 50 or 60 Hz. Options = 50 or 60
  - **to\_pandas** (*bool (Optional)*) – Flag to save output to pandas instead of xarray. Default = True.
- Returns:** **harmonic\_amplitudes** (*pandas DataFrame or xarray Dataset*) – Amplitude of the time-series data harmonics indexed by the harmonic frequency with signal name columns

---

**mhkit.power.quality.harmonic\_subgroups**(*harmonic\_amplitudes: Series | DataFrame | DataArray | Dataset, grid\_freq: int, frequency\_dimension: str = "", to\_pandas: bool = True*)→ DataFrame | Dataset [\[source\]](#)

Calculates the harmonic subgroups based on IEC 61000-4-7

- Parameters:**
- **harmonic\_amplitudes** (*pandas Series, pandas DataFrame, xarray DataArray, or xarray Dataset*) – Harmonic amplitude indexed by the harmonic frequency
  - **grid\_freq** (*int*) – Value indicating if the power supply is 50 or 60 Hz. Options = 50 or 60
  - **frequency\_dimension** (*string (optional)*) – Name of the xarray dimension corresponding to frequency. If not supplied, defaults to the first dimension. Does not affect pandas input.
  - **to\_pandas** (*bool (Optional)*) – Flag to save output to pandas instead of xarray. Default = True.
- Returns:** **subgroup\_results** (*pandas DataFrame or xarray Dataset*) – Harmonic subgroups indexed by harmonic frequency with signal name columns

---

**mhkit.power.quality.total\_harmonic\_current\_distortion**(*harmonics\_subgroup: Series | DataFrame | DataArray | Dataset, frequency\_dimension: str = "", to\_pandas: bool = True*)→ DataFrame | Dataset [\[source\]](#)

Calculates the total harmonic current distortion (THC) based on IEC/TS 62600-30

- Parameters:**
- **harmonics\_subgroup** (*pandas Series, pandas DataFrame, xarray DataArray, or xarray Dataset*) – Subgrouped current harmonics indexed by harmonic frequency
  - **frequency\_dimension** (*string (optional)*) – Name of the xarray dimension corresponding to frequency. If not supplied, defaults to the first dimension. Does not affect pandas input.
  - **to\_pandas** (*bool (optional)*) – Flag to save output to pandas instead of xarray. Default = True.

**Returns:** **thcd\_result** (*pd.DataFrame or xarray Dataset*) – Total harmonic current distortion indexed by signal name with THCD column

---

**mhkit.power.quality.interharmonics**(*harmonic\_amplitudes: Series | DataFrame | DataArray | Dataset, grid\_freq: int, frequency\_dimension: str = "", to\_pandas: bool = True*)→ *DataFrame | Dataset* [\[source\]](#)

Calculates the interharmonics from the harmonic\_amplitudes of current

**Parameters:**

- **harmonic\_amplitudes** (*pandas Series, pandas DataFrame, xarray DataArray, or xarray Dataset*) – Harmonic amplitude indexed by the harmonic frequency
- **grid\_freq** (*int*) – Value indicating if the power supply is 50 or 60 Hz. Options = 50 or 60
- **frequency\_dimension** (*string (optional)*) – Name of the xarray dimension corresponding to frequency. If not supplied, defaults to the first dimension. Does not affect pandas input.
- **to\_pandas** (*bool (Optional)*) – Flag to save output to pandas instead of xarray. Default = True.

**Returns:** **interharmonic\_groups** (*pandas DataFrame or xarray Dataset*) – Interharmonics groups