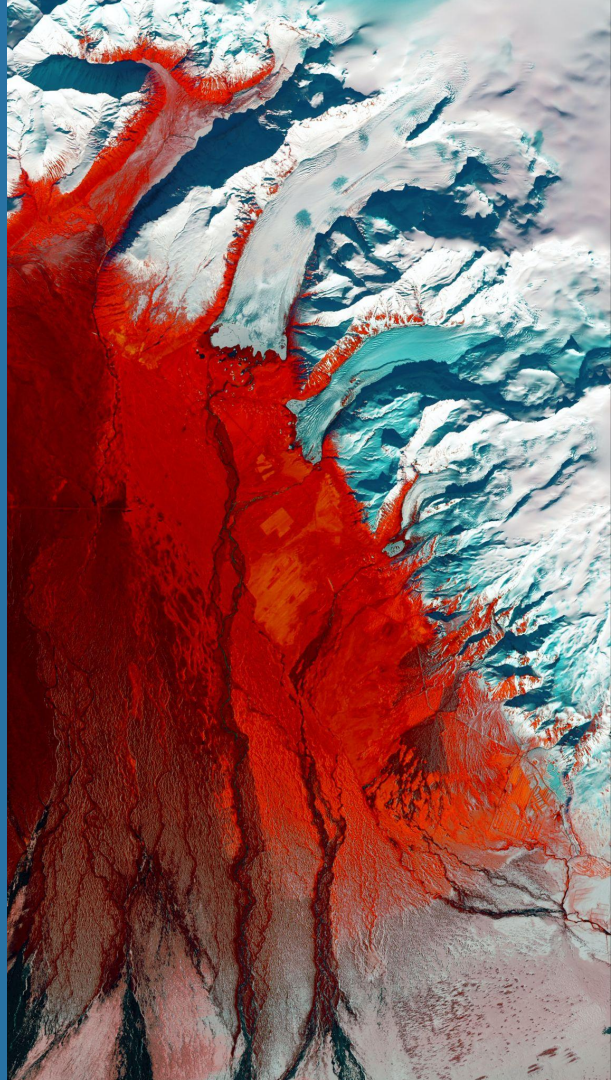
Development status of a schema for wind turbine power curves.

Looking at a candidate schema for IEC 61400-16, showing some usage examples and points for development/improvement.

Tom Clark
CEO Octue
16 May 2023



Octue

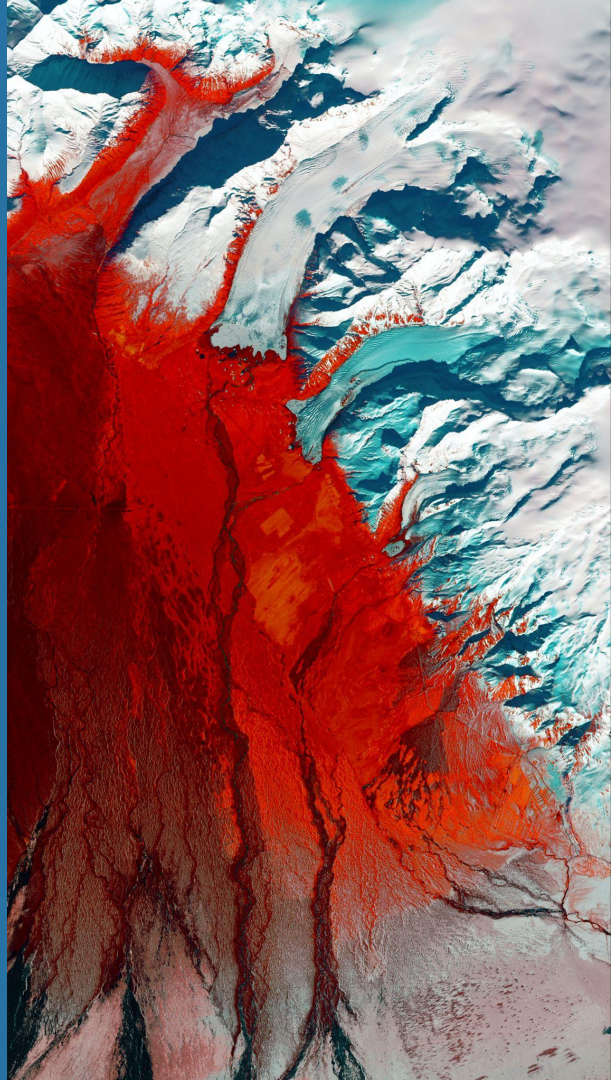
Our mission: Help scientists and engineers work more effectively with data

Octue is a solutions provider:

- we develop open-source tools
- and provide consultancy on data engineering and application development

Why are we here?

- we developed a Power Curve Schema for our customer WindPioneers, who have subsequently open-sourced it for the benefit of the community



Today

1. **An open source power curve schema**

   Used and evolved in production for 2 years to represent hundreds of different turbines in full-timeseries, bankable resource assessments

2. **The tools ecosystem**

   Generated webforms for data entry, QC processes, plotting tools, databases and report generators

3. **Areas of improvement**

   Generated webforms for data entry, QC processes, plotting tools, databases and report generators

The schema

Schema

- JSONSchema for power curves available at:

  strands.octue.com/windpioneers/power-curve

- ‘Strands’ is a repository application we’re developing for schema. It has a number of benefits I won’t cover here, but please ask me if you’re interested in JSONSchema.

DOI 10.5281/zenodo.7940068

Read the poster

- It describes the method we used to build it

  https://doi.org/10.5281/zenodo.7940068

- Broadly:
  - 2 years in production of which >18 months stable (no breaking changes to the schema)
  - Used for > 30 turbines from 9 manufacturers

Why JSONSchema?

Extensive review of the attributes and suitability of different schema languages undertaken in IEA Task 43. See:

Clark, Thomas “How to communicate and collaborate on data: easy-to-use tools and techniques for eliminating overwhelm, confusion and ambiguity.” June 2022
https://doi.org/10.5281/zenodo.7928816

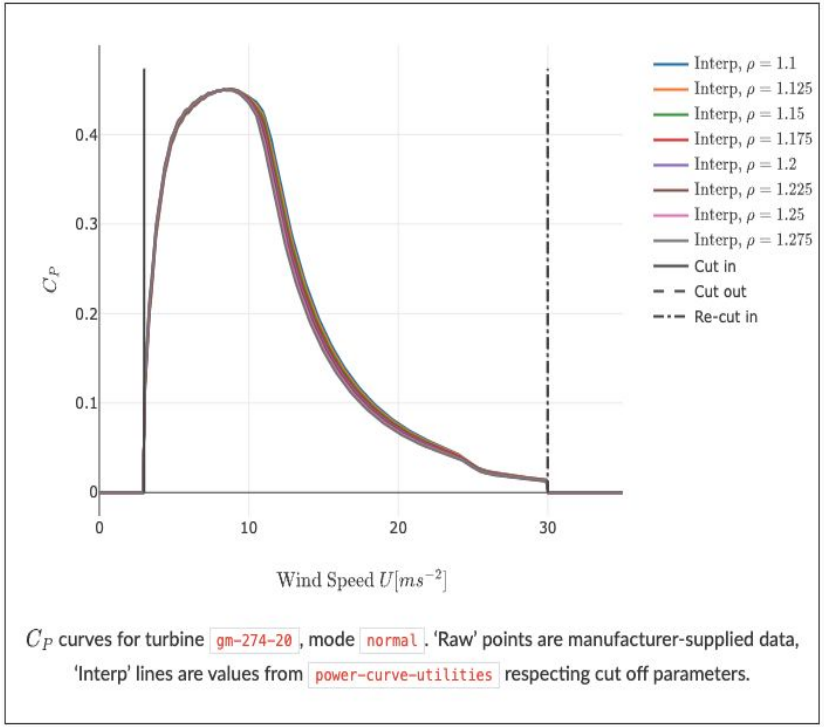
Schema and data

Tom to show the schema and matching example data

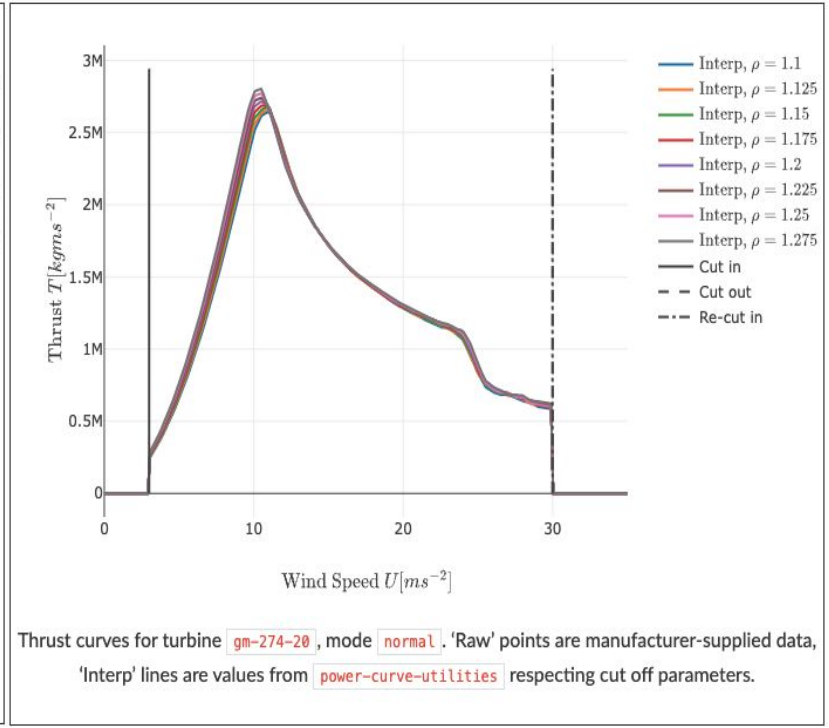
- Data at https://codesandbox.io/s/9pl493?file=/src/example.json

Tools and ecosystem

A JSONSchema allows quick development of a rich set of tools and integrations around the schema. Here are a few of the tools that have been built on top.



C_P curves for turbine `gm-274-20`, mode `normal`. 'Raw' points are manufacturer-supplied data, 'Interp' lines are values from `power-curve-utilities` respecting cut off parameters.



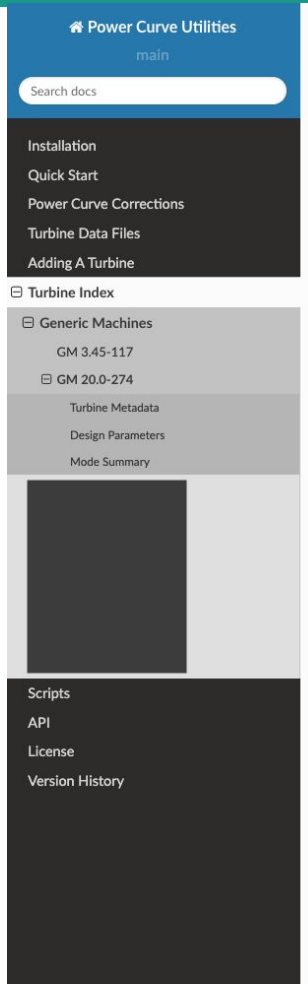
Thrust curves for turbine `gm-274-20`, mode `normal`. 'Raw' points are manufacturer-supplied data, 'Interp' lines are values from `power-curve-utilities` respecting cut off parameters.

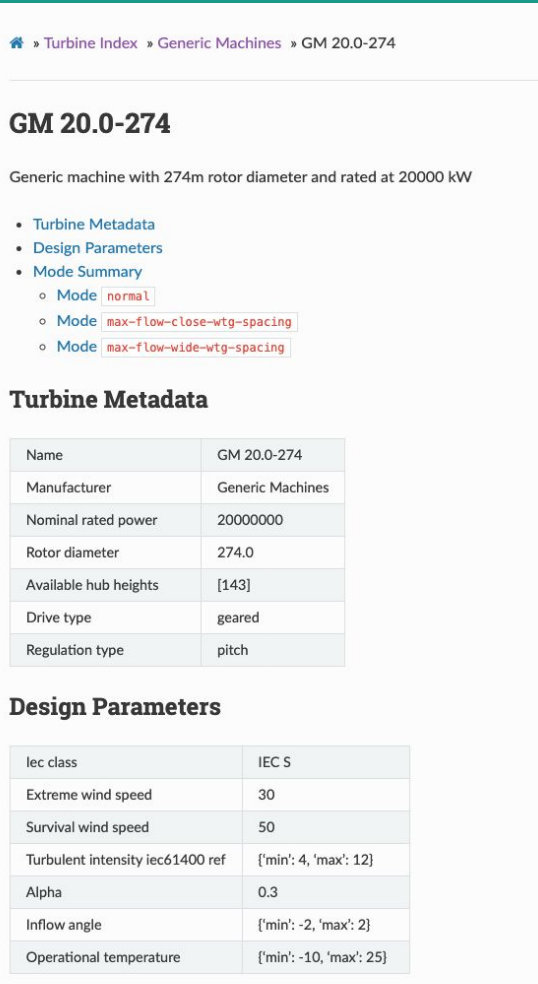
CP, T charts expressed in plotly-compliant JSON, interactively viewable in browser. This is very useful for QC checks as well as inspection of turbine performance characteristics.

Automated reports

Reports can be generated from the data (we use the sphinx rendering engine to generate static HTML and PDFs).

An overview of the metadata and design parameters sections for a generic turbine.



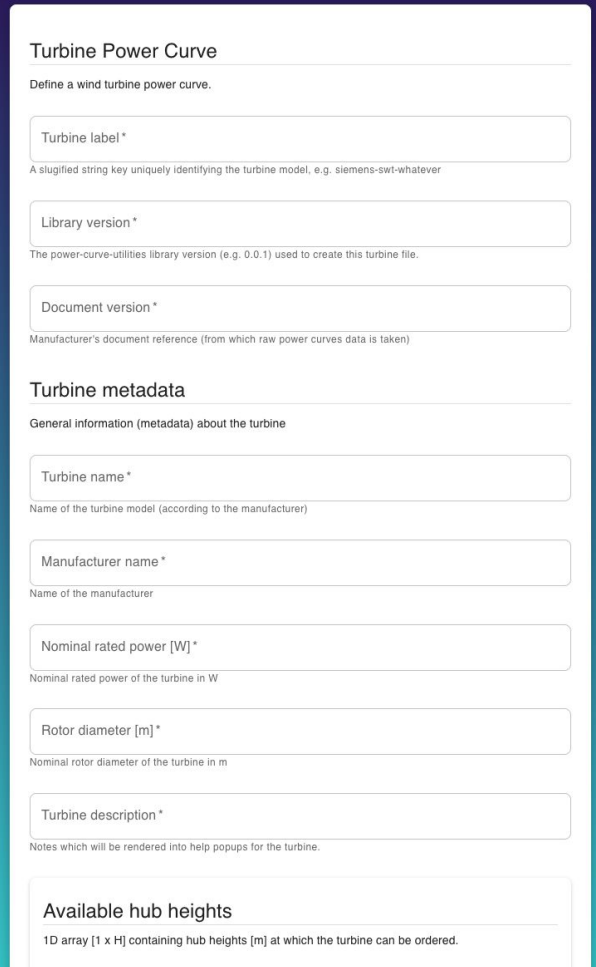


Input Forms

JSONSchema can be used to quickly generate web forms for data entry. The default table fields are not really suitable for multidimensional data but could be made fit for purpose.

https://9pl493.csb.app/

SCAN ME



Areas for improvement

There are many improvements that could be made to the tooling ecosystem, and a client library to work with power curves that matches the standard would be extremely useful to have as an officially supported outcome. HOWEVER, here, I want to focus solely on the schema itself.

Remove label and library version

- ‘label’ isn’t related to the turbine itself but how we refer to it in an external system. We could remove that, although should encourage the use of a unique ID.
- ‘library_version’ is deprecated.


```
{
  "label": "gm-274-20",
  "library_version": "0.0.2",
  "document_version": "Made Up Performance
  "turbine_metadata": {…
  },
  "design_parameters": {…
  },
  "default_mode": "normal",
  "modes": […
  ]
}
```

Remove document_version?

- ‘document_version’ refers to the PDF from which we extracted the data. Maybe this is no longer relevant


```
{
  "label": "gm-274-20",
  "library_version": "0.0.2",
  "document_version": "Made Up Performance
  "turbine_metadata": {…
  },
  "design_parameters": {…
  },
  "default_mode": "normal",
  "modes": […
  ]
}
```

Design Parameters

We've basically created this field from what we've seen in PDF specification documents. This is the weakest area of the schema because we capture that information but don't really use it ourselves.

Is there scope for richer data? And/or for tighter validation (based on rules I don't understand?).

```
"design_parameters": {
  "iec_class": "IEC S",
  "extreme_wind_speed": 30,
  "survival_wind_speed": 50,
  "turbulent_intensity_iec61400_ref": {
    "min": 4,
    "max": 12
  },
  "alpha": 0.3,
  "inflow_angle": {
    "min": -2,
    "max": 2
  },
  "operational_temperature": {
    "min": -10,
    "max": 25
  }
},
```

Improved metadata

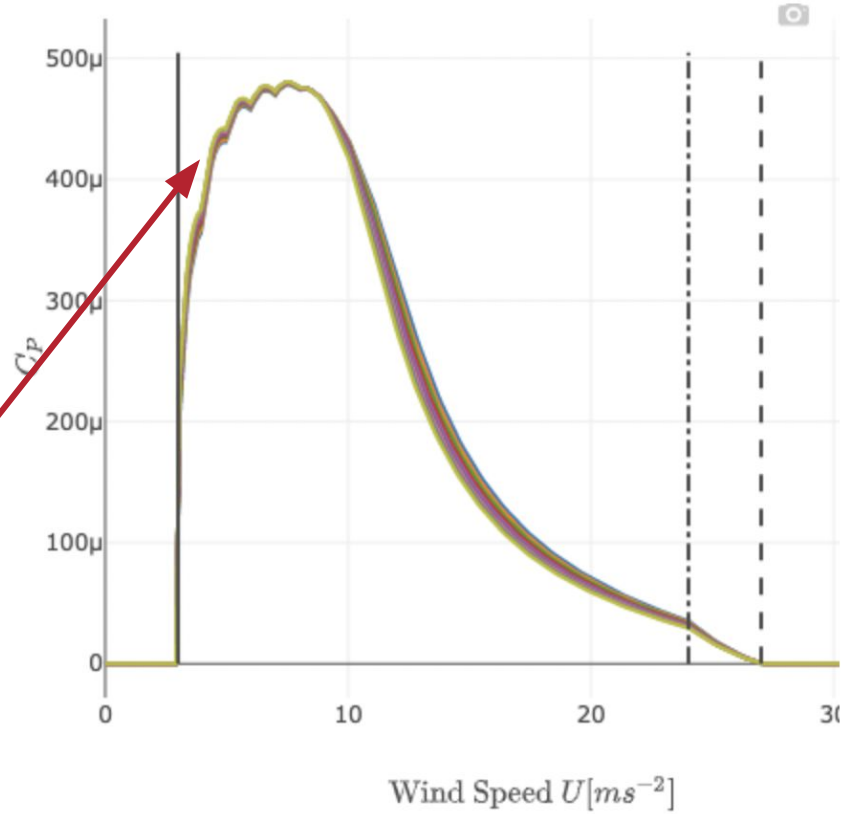
- Should require a superset of dublin core.
- Turbine ‘label’ isn’t related to the turbine itself but how we refer to it in an external system. We could remove that, although should encourage the use of a unique ID

Coefficient / SI Conversion

Presently, the schema allows P, T, curves to be specified in coefficient form or in SI units.

I think this is important to maintain, to facilitate different use cases - and that a client library should be used to convert as required.

But, to avoid ringing in the resulting curve, robust polynomial division is required in the client library: which we're not providing right now. So we need to take care of alerting users to the flexibility/complexity tradeoff.



Pre-cone angle

Not currently present, addition would be required for simulation tool setup and potentially other use cases

Pre-defined parameter list

Presently, parameters for variation can be specified in any way you wish, by creating labeled parameters and defining their ranges. For example:

- air-density, wind-speed
- air-density, wind-speed, turbulent-intensity
- air-density, wind-speed, turbulent-intensity, monin-obukhov-stability
- whatever, whatever-else

It's not a strict approach, because this required that the end client knows what to do with those labels. We need to decide whether to constrain these. If we do, the schema will be simpler but less future-proof.

Mode-specific overrides

Currently, hub heights are specified for the turbine model as a whole, however, some OEMs provide available hub heights that vary based on the operational modes listed.

How should we be managing this? Adding an optional override on a per-mode basis?

This also applies to the nominal rated power of the turbine, which we overcame by applying a ‘default mode’.

Octue

Thank you to WindPioneers for willingly open-sourcing!

Please follow Octue on LinkedIn!

There will be many more resources coming in the following months...