docs Documentation

Release 1.0

Will Rogers, Razvan Vasile

Jun 20, 2019

Contents

Python Toolkit for Accelerator Controls (Pytac) is a Python library for working with elements of particle accelerators.

It is hosted on Github at https://github.com/willrogers/pytac.

Two pieces of software influenced its design:

- Matlab Middlelayer, used widely by accelerator physicists
- APHLA, high-level applications written in Python by the NSLS-II accelerator physics group

CHAPTER 1

Software Overview

Pytac provides a Python library `pytac` that makes it easier to communicate with machine hardware for online applications. Although it currently works with EPICS, the *cs.ControlSystem* class may be sub-classed to allow other control systems to be used.

The design is based around a `Lattice` object that contains a sequence of `Element` s. Each element may have zero or more 'fields' (examples x, y, a1 or b2), associated with which are `Device` objects. These devices contain the necessary information to request live data from the control system.

Data may be requested in `ENG` or `PHYS` units and will be converted as appropriate. Two types of unit conversion are available: Polynomial (often used for linear conversion) and Piecewise Cubic Hermite Interpolating Polynomial (Pchip; often used for magnet data where field may not be linear with current). In the case that measurement data (used to set up the conversion objects) is not in the same units as the physical models, further functions may be given to these objects to complete the conversion correctly.

Elements may be grouped into families (an element may be in more than one family) and requested from the lattice object in those families.

Machines are defined using a set of `.csv` files, located by default in the `pytac/data` directory.

CHAPTER 2

Contents:

2.1 Examples

2.1.1 Initialisation

- Install pytac and cothread for EPICS support, and start Python:

```
$ pip install pytac
$ pip install cothread
$ python
Python 2.7.3 (default, Nov  9 2013, 21:59:00)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-3)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- Import 'pytac' and initialise the VMX ring mode:

```
>>> import pytac.load_csv
>>> lattice = pytac.load_csv.load('VMX')
```

The `lattice` object is used for interacting with elements of the accelerator.

2.1.2 Print BPM pvs along with s position

- Get elements representing BPMs:

```
>>> bpms = lattice.get_elements('BPM')
```

- Print the device names and s position of each BPM:

```
>>> for bpm in bpms:
>>>     print('BPM {} at position {}'.format(bpm.get_device('x').name, bpm.s))
```

(continues on next page)

(continued from previous page)

```
BPM SR01C-DI-EBPM-01 at position 4.38
BPM SR01C-DI-EBPM-02 at position 8.8065
BPM SR01C-DI-EBPM-03 at position 11.374
BPM SR01C-DI-EBPM-04 at position 12.559
BPM SR01C-DI-EBPM-05 at position 14.9425
...
```

- Get PV names and positions for BPMs directly from the lattice object:

```
>>> lattice.get_pv_names('BPM', 'x', pytac.RB)
['SR01C-DI-EBPM-01:SA:X',
'SR01C-DI-EBPM-02:SA:X',
'SR01C-DI-EBPM-03:SA:X'
...
>>> lattice.get_pv_names('BPM', 'y', pytac.RB)
['SR01C-DI-EBPM-01:SA:Y',
'SR01C-DI-EBPM-02:SA:Y',
'SR01C-DI-EBPM-03:SA:Y',
...
>>> lattice.get_family_s('BPM')
[4.38,
 8.806500000000002,
 11.374000000000002,
 ...
```

2.1.3 Get the pv value from the quad elements

- Get the Quad elements and print their readback values on the b1 field:

```
>>> quads = lattice.get_elements('QUAD')
>>> for quad in quads:
>>>     print(quad.get_pv_value('b1', pytac.RB))
71.3240509033
129.351394653
98.2537231445
...
```

- Print the quad pv values on the b1 field using the lattice. This is more efficient since it uses only one request to the control system:

```
>>> lattice.get_pv_values('QUAD', 'b1', pytac.RB)
[71.32496643066406,
 129.35191345214844,
 98.25287628173828,
...
```

2.2 API documentation

2.2.1 pytac.cs module

2.2.2 pytac.device module

2.2.3 pytac.element module

2.2.4 pytac.epics module

2.2.5 pytac.exceptions module

2.2.6 pytac.lattice module

2.2.7 pytac.load_csv module

2.2.8 pytac.units module

CHAPTER 3

Indices and tables

- genindex
- modindex
- search