Implements of CPF and CRD Algorithm in Python

Li R [1,2] , Li Z [1,2] , He L [1] , Xiong Y [1]
1 Yunnan Observatories, CAS, Kunming Yunnan, China; 2 Key Laboratory of Space Object and Debris Observation, CAS

Abstract

It is a good try to implement CPF and CRD algorithm of ILRS in python, as the python is so powerful, popular, widely used in science research, and there are so many powerful packages, such as NumPy, pandas, matplotlib, SciPy, Astropy, etc. After that, it may be more easy to use, and much more extendible. It will be the first release in this workshop. The provisional name is 'LrPy'. According to the original sample code, for CPF, it mainly implements readin, files managing, check, interpolating for sched/slr/llr, and for CRD, it mainly implements readin, write, check, split, merg, conversion between CRD and older SLR/LLR formats.

Motivation

Currently the algorithm of CPF and CRD are implemented in C and fortran. It is very useful and powerful. But it is required to be compiled first, and a little complicate. This python package is developed to simplify the most commonly used procedures, sched and pred for CPF, and read/conversion for CRD.

Key Features

- Auto downloading CPF file from ftp if not exist locally
- Satellite Query by id, name, norad, ext
- Reading and interpolating for CPF
- Pass finding in seconds resolution (sched)
- Generating tracking file (pred)
- Configuration based
- Custom trackdata formatter
- Conversions for CRD

Example Usage

The using of lrpy is simple.

```
>>>import lrpy as lp
>>>cpf_filenames = lp.util.list_cpf_filenames('ajisai')
>>>cpf = lp.cpf.CPF(cpf_filenames[0])
>>>passes = lp.sched(cpf)

>>>npass = passes[0]
>>>start_datetime = npass.open_trackdata.epoch
>>>end_datetime = npass.close_trackdata.epoch
>>>track_filename = '{}.{}'.format(start_datetime.strftime('%Y%m%dT%H%M%S'),
                                  lp.satellite_by_name('ajisai')['ext'])
>>>trackdatas = lp.predict_all(cpf, lp.default_station,
                                  start_datetime, end_datetime,
                                  interval='1s', filename=track_filename,
                                  formatter=lp.core.default_trackdata_formatter)
```

Specifications

Modules

`satellite`	ILRS satellites manager
`station`	Station related
`cpf`	CPF file
`core`	Core lib (slr and llr)
`util`	Util for file handling
`constants`	Constants related
`lagrange`	lagrange interpolation
`math3d`	3D math

Key Interfaces

`list_cpf_filenames`	List CPF file names
`sched`	Find pass schedule
`predict_all`	Generate tracking data
`predict`	Compute a TrackData for SLR
`predict_llr`	Compute a TrackData for LLR
`default_trackdata_formatter`	Custom formatter
`satellite_by_name`	Query satellite by official name

Tests

Passed sched and pred tests including in original ILRS sample code.

- sched: gps36_cpf_051129_0083.cod, (2005-12-01T00:00:00~2005-12-04T23:00:00), MLRS01(7080)
- pred_slr: gps36_cpf_051129_0083.cod, (2005-11-30T01:12:00~2005-12-01T00:00:00, 1s) , ZIML(7810)
- pred_llr: apollo15_cpf_061220_8551.utx, (2006-12-20T14:15:00~2006-12-20T22:15:00, 15m), MLRS01(7080)

contract me
lirw@ynao.ac.cn