

Modelling with Spicy: control loops for seismic isolation, characterization of sensor noise, and more

Artem Basalaev (Universität Hamburg)

On behalf of current Spicy developers: C. Darsow-Fromm (U Hamburg), O. Gerberding (U Hamburg), M. Hewitson (U Hannover), N. A. Holland (Nikhef), C. Mow-Lowry (VU Amsterdam), A. Patra (U Cardiff), P. Saffarieh (Nikhef), O. Vega (MIT), D. Voigt (U Hamburg)

ET-ISB workshop, L'Aquila

Simulation of ET-LF lower stage and angular controls

18 October 2022



Spicypy project



Combining several tools to facilitate **signal processing, control systems modelling, and the interface between the two.**

LVK

Einstein Telescope

LISA

Laser physics, material science, ...

- **Strong ties between communities & projects** (overlap in collaboration membership)
- Nice to **open source some of our algorithms** developed over the years
- And have them available in a **neat self-contained package** that allows to **build models across different applications** in GW instrument science
- Some **previous efforts** in this direction **exist** already: great! **base our code on them** and don't reinvent the wheel

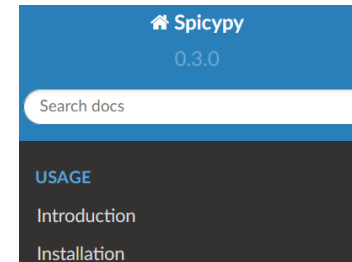
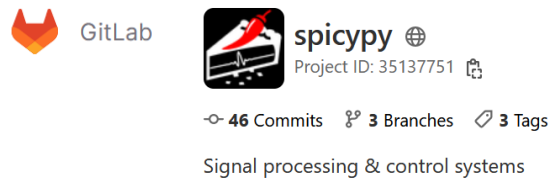
Project started early this year

- we did **a lot of planning** before actual coding

Spicyypy project

Since then:

We've got a nice name (registered in pypi.org), [gitlab project](#) & all machinery set up



Contributors

- Artem Basalaev <artem.basalaev@physikDOTuni-hamburg.de>
- Christian Darsow-Fromm <cdarsowf@physnetDOTuni-hamburg.de>
- Oliver Gerberding <oliver.gerberding@physikDOTuni-hamburg.de>
- Martin Hewitson <martin.hewitson@aeiDOTmpg.de>
- Nathan A. Holland <n holland@nikhefDOTnl>
- Conor Mow-Lowry <c.m.mow-lowry@vuDOTnl>
- Abhinav Patra <patraa1@cardiffDOTac.uk>
- Pooya Saffarieh <p.saffarieh@nikhefDOTnl>
- Octavio Vega <ovega84@mitDOTedu>
- Daniel Voigt <daniel.voigt@physikDOTuni-hamburg.de>

Concept is finalized:

- Base on [GWpy](#) (and [scipy](#)) for signal processing and on [python-control](#) for control systems
- **Development aim 1: interface**
 - seamlessly feed GWpy-like signals through control systems, object-oriented intuitive model-building
- **Development aim 2: enhance underlying packages**
 - expand and improve upon GWpy and python-control, and propagate these changes upstream (we are in contact with GWpy and python-control developers)

Developer team formed:

- researchers from 6 universities and 4 countries
- with ties to LIGO, LISA and ET already **directly contributed code/algorithms**;
- even larger community follows up on the project and joins our meetings

Spicypy project: examples

- We are maintaining documentation for both [users](#) and [developers](#), and make an example for each new feature; will discuss some of them today

- Example 1: Feed a signal through a simple control system

- Creating a control system
- Creating a test signal and feeding it through the control system

- Example 2: Time-match signals and estimate coherence

- Open STACIS data
- Create time series and visualize them, calculate PSD
- Matched filtering and coherence

- Example 3: Transfer function between time series and Daniell averaging method

- Estimating transfer function
- ASD, PSD, CSD and transfer function with Daniell averaging method

- Example 4: Plot Einstein Telescope alignment sensing noise and make bode plots

- Imports
- Useful Functions
- Data
- The Transfer Functions (kept in phase space format)
- Plot the Data

- Example 5: Model active seismic isolation ("VATI Grav" platform)

- Create model of the control system
- Examine closed-loop impulse and step response
- Feed seismic motion into System 1 and calculate response
- Feed sensor noise into System 2, calculate response and add it in quadrature

Modelling active seismic isolation

“VATIGrav” platform

- Project in our lab: actively isolated vacuum chamber
- Focus of [this example](#): first simple 1D-model of isolation control with sensor noise

Modelling seismically isolated vacuum chamber “VATIGrav”

Authors: Octavio Vega <ovega84@mitDOTedu>, Artem Basalaev artem.basalaev@physikDOTuni-hamburg.de

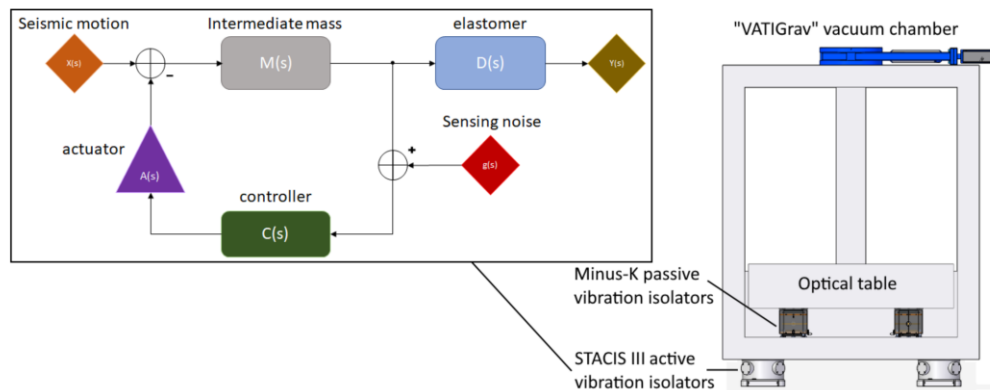
If not done already, install spicypy (uncomment the line below):

```
! pip install spicypy
```

Note: on Windows the above might not work, you may see a compilation error in `ligo-segments` package. If this happens, please install `gpy` separately first, by running `conda install -c conda-forge gpy` in without Anaconda. We are working on that!

Create model of the control system

VATIGrav chamber and corresponding schematic of the control system for active isolation:



System 1 (feedback system taking in seismic motion)

```
[3]: # create a feedback loop
from spicypy.control.system import System
sys_plant = System(M)
sys_ctrl = System(C)*System(A)

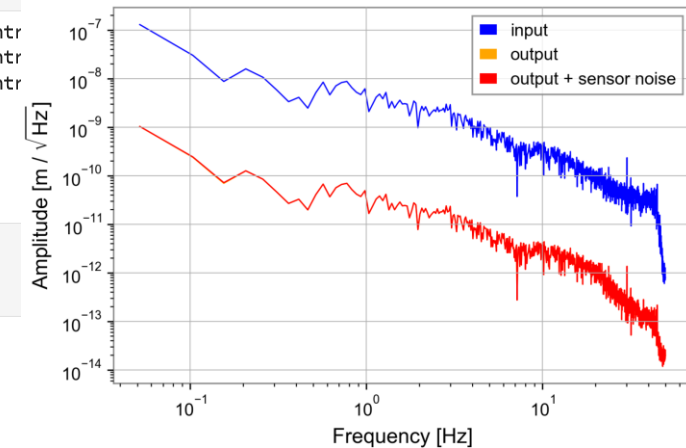
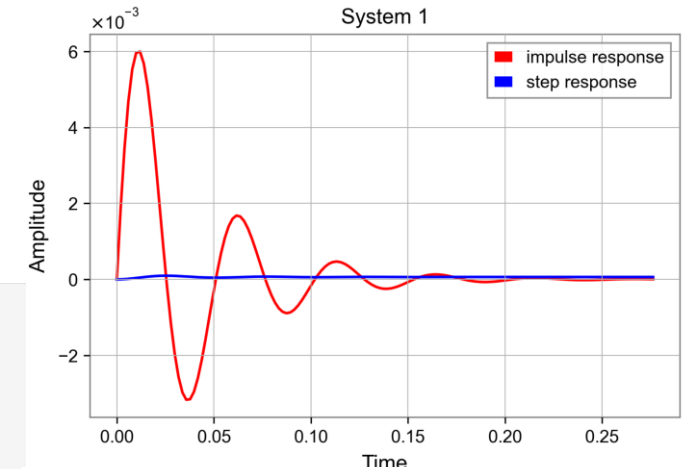
sys1 = sys_plant.feedback(sys_ctrl)
sys1 = System(sys1*D)
```

Info: supplied transfer function was converted to contr
 Info: supplied transfer function was converted to contr
 Info: supplied transfer function was converted to contr
 Info: supplied control.StateSpace
 Info: supplied control.StateSpace

System 2 (sensing noise open loop)

```
[4]: # create a series connection
olg = sys_ctrl*sys_plant
sys2 = System(olg)
```

Info: supplied control.StateSpace



Modelling active seismic isolation

“VATIGrav” platform

- Where Spicypy comes in?

System 1 (feedback system taking in seismic motion)

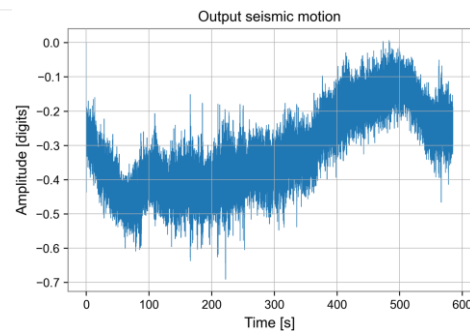
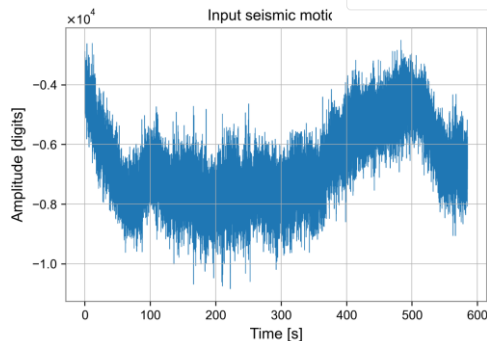
```

# create a feedback loop
from spicypy.control.system import System
sys_plant = System(M)
sys_ctrl = System(C)*System(A)

sys1 = sys_plant.feedback(sys_ctrl)
sys1 = System(sys1*D)
  
```

```

response_seismic = sys1.response(input_seismic)
plt.plot(response_seismic.times, response_seismic, lw=0.5)
plt.xlabel('Time [s]')
plt.ylabel('Amplitude [digits]') # we have time series in digits
plt.title('Output seismic motion')
  
```



System class encapsulates control.StateSpace response() to time series implemented

```

class System(control.StateSpace):
    """
    Class to model control systems and their response

    """

    def feedback(self, *args, **kwargs):
        """Add a feedback connection
        Parameters
        -----
        args : list
            positional arguments, all passed on to `control.feedback`
        kwargs : dict
            additional arguments, all passed on to `control.feedback`
        Returns
        -----
        System : System
            Control system with feedback connection
        """
        return System(control.StateSpace.feedback(self, *args, **kwargs))

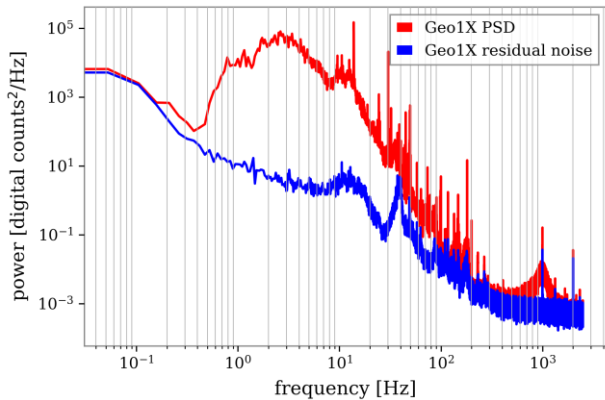
    def response(self, time_series):
        """Calculate system's response to an input signal
        Parameters
        -----
        time_series : TimeSeries
            input signal time series
        Returns
        -----
        time_series : TimeSeries
            output signal time series
        """
        t = np.array(time_series.times)
        v = np.array(time_series.value)
        resp = control.input_output_response(control.ss2io(self), t, v)
        return TimeSeries(resp.outputs, times=resp.time, unit=time_series.unit)
  
```

Future:

- Simulink-like schematic of control systems (static)
- Frequency response
- Multiple inputs...

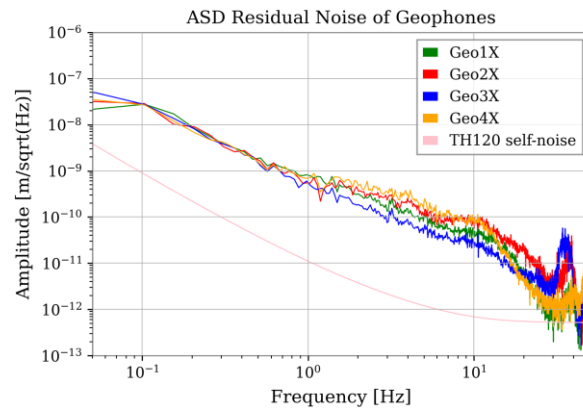
Estimating sensor self-noise with Huddle test

- Multi-channel coherent subtraction (Huddle test) algorithm [<https://doi.org/10.1063/1.5000592>], further developed by others*, [already open source](#) in Spicypy
- Easy to use interface in development



Coherently subtract Geo2-4X PSDs from Geo1X PSD

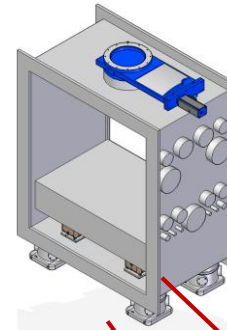
Going to be added to Spicypy "signal" module



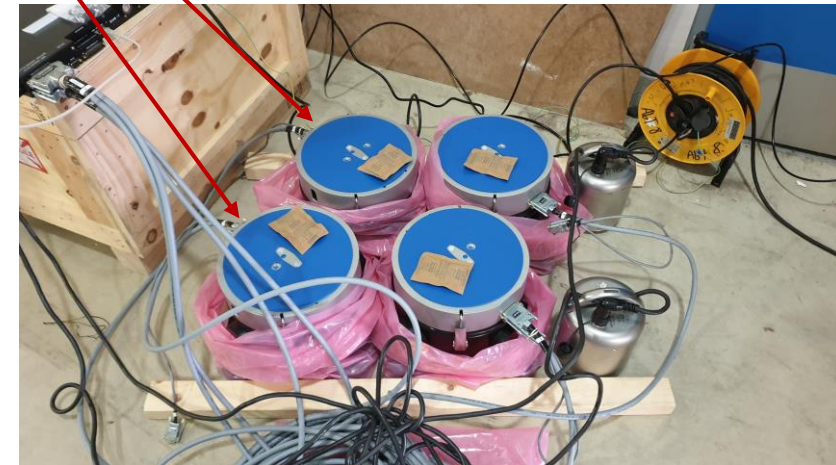
Convert to proper units using seismometers for calibration

```

spicypy/signal/cmccs.py 0 - 100644
1 +
2 +
3 + An attempt to make cmccs in Python. We generalise this to support cross spectral
4 + densities, something I assert is useful.
5 +
6 + Although I only, properly, understand an alternative method for doing this
7 + procedure that suffers from poor behaviour with non-linearly independent noise
8 + sources. Alternatively I know that cmccs works, and the community is happy with
9 + its performance - so I will copy its functionality.
10 +
11 + Authors: Nathan A. Holland, Pooya Saffarieh, Brian Lantz, Conor Mow-Lowry
12 + Contact: nholland@ikhef.nl
13 + Date: 2022-05-20
  
```



- Recorded data from geophones in STACIS III
- + Two seismometers for calibration



* Nathan A. Holland, Pooya Saffarieh, Brian Lantz, Conor Mow-Lowry

Modelling suspensions of ET

- Alignment sensing noise, [early quick example](#)
 - Adapted from Matlab script by C. Mow-Lowry

The plant:

```

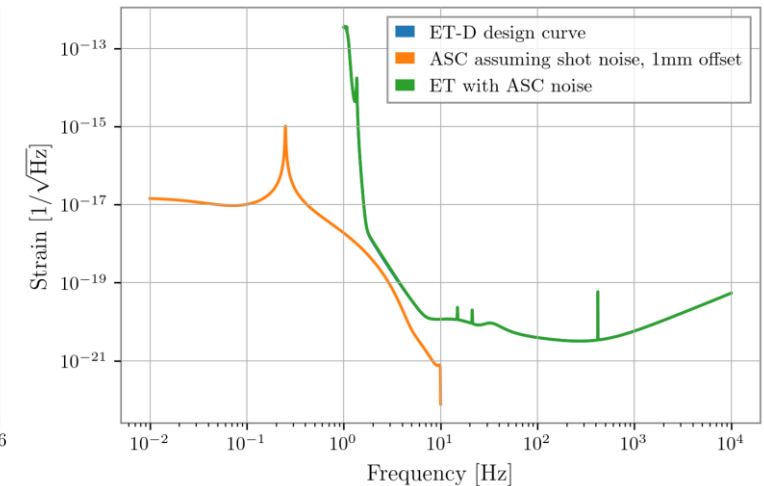
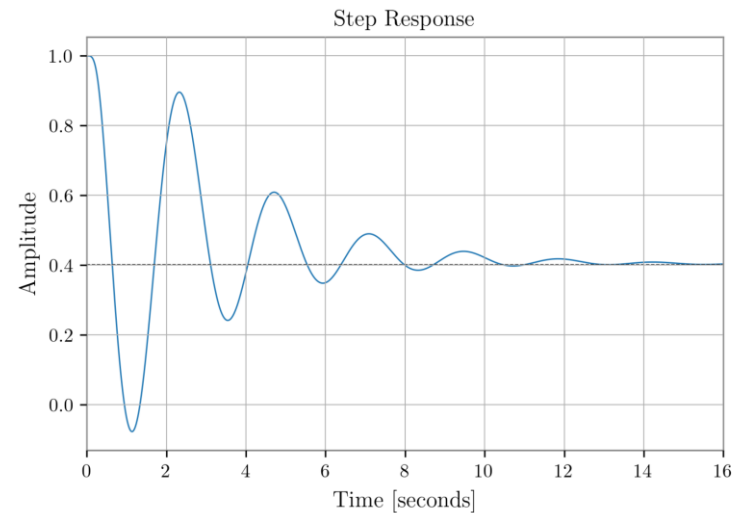
# For plant, want to assume a pole at 60mHz, Virgo-Like, but stiffened to
# 250mHz by Sigg-Sidles second pole at differential mode, assuming
# 'irrelevantly high' at ~10 Hz.
p = System(zpkgf([], pairQ(250e-3,100) + pairQ(10,100), 1e-1, 1))
  
```

The controller:

```

# Controller should have UGF >250mHz and noise less than ET-D at 3 Hz.
# Will use several cut-off sections
cut1 = System(zpkgf(pairQ(5,2), [1.2, 1.5], 1, 0))
cut2 = System(zpkgf(pairQ(10,1000) + pairQ(10,5), pairQ(10,10) + pairQ(5,1), 1, 0))
cont1 = System(zpkgf([0.1,0.1], [0.03] + pairQ(3,0.7), 1, 0.3))

c = cont1*cut1*cut2
  
```



Aspiration: make the model more realistic based on **Virgo ASC noise experience**;
potential integration with Finesse 3 - in talks with Andreas Freise, Riccardo Maggiore

Other small things...

- Transfer function class
 - Now in frequency domain, extend to time domain...

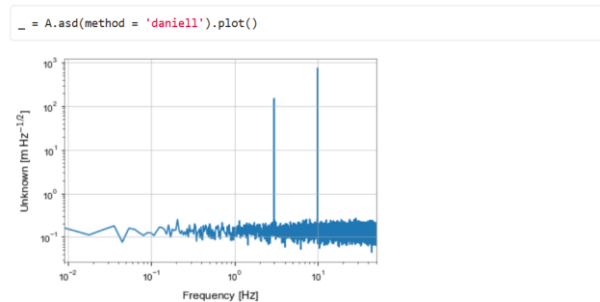
```

class TransferFunction: # pylint: disable=too-many-instance-attributes
    """
    Class calculating transfer function between two time series (result is FrequencySeries)
    """

    def __init__(
        self,
        ts1: TimeSeries = None,
        ts2: TimeSeries = None,
        tf: FrequencySeries = None,
        name="Generic transfer function",
        **kwargs,
    ):
        """Constructor takes either two time series_or_ another TransferFunction object
  
```

- Alternative spectrum calculation (averaging) methods
 - “Daniell”
 - “LPSD”
 - ...

ASD, PSD, CSD and transfer function with Daniell averaging method



... and rest of [open Issues](#)

pyda group > spicypy > Issues

Open 15 Closed 19 All 34

Search or filter results...

Version 0.3

- #34 - created 2 hours ago by Christian
- Add *args and **kwargs to response method of System class #32 - created 3 weeks ago by Octavio Vega
- Make daniell averaging method more modular #31 - created 1 month ago by Artem BasalaeV [feature](#) [LVK conference talk about Spicypy](#)
- Add dynamic averaging to daniell method #29 - created 1 month ago by Artem BasalaeV [feature](#) [LVK conference talk about Spicypy](#)
- Increase performance with Numba and/or numexpr? #27 - created 1 month ago by Christian [feature](#)
- Dependencies (GWpy and python-control) may return objects of their classes, which are restricted comparing to Spicypy classes. #23 - created 2 months ago by Artem BasalaeV [bug](#)
- Daniell method CSD fails with number_averages > time series length AND complex input data #22 - created 2 months ago by Artem BasalaeV [bug](#)
- Add multi-channel coherence estimation (to perform Huddle test) #19 - created 2 months ago by Artem BasalaeV [feature](#)
- Handle multiple inputs/outputs in System.response #17 - created 3 months ago by Artem BasalaeV [feature](#)
- Add history tracing in control systems creation #16 - created 3 months ago by Artem BasalaeV [feature](#)
- Nicer-looking units? #13 - created 3 months ago by Artem BasalaeV [review](#)
- Check/implement uncertainties for data values #11 - created 3 months ago by Artem BasalaeV [review](#)
- Implement LPSD in TimeSeries #10 - created 3 months ago by Artem BasalaeV [feature](#) [LVK conference talk about Spicypy](#)
- Convert spectrum to different units/type (lossless) #7 - created 3 months ago by Artem BasalaeV [feature](#) [Implement most requested features in Spicypy](#)
- Control system transfer function estimate from the noise #5 - created 3 months ago by Artem BasalaeV [feature](#) [Implement most requested features in Spicypy](#)



Questions that came up in this workshop

- Can I optimize my controller using Spicypy?
- Some optimization already available through python-control: H-infinity control synthesis. But we definitely want to look into/expand this functionality!
- I have some code that works (Matlab, of course) and I'm willing to open source it but I don't have time/knowledge to translate it into python
- Let us look into it! If what code is doing is generally useful, we will help translate it into python. Conor's ET suspensions example above (admittedly simple) was quite easy to translate, most functionality already there

`control.hinfosyn(P, nmeas, ncon)`

H_{∞} control synthesis for plant P.

Parameters

P (partitioned lti plant) -

nmeas (number of measurements (input to controller)) -

ncon (number of control inputs (output from controller)) -

Returns

K (controller to stabilize P (State-space sys))

Future direction

- **Finalize implementing planned functionality**
 - Example: Simulink-like (but static) graphical schematics for control systems created with Spicypy
- **Propagate upstream**
 - open Issues on GWpy gitlab with suggested changes to default behavior, new spectrum calculation methods
 - implement already suggested changes to default behavior of python-control
- **Encourage wider usage**
 - Spicypy likely to be used for ET design
- **Open for new suggestions** in Issues section from everyone
- **Need more person-power:** consider joining as a developer!



spicypy

Signal processing & control systems. Combining several tools to facilitate between the two.

More details are provided in [Documentation](#).

Development roadmap

Spicypy is based on two main packages: [GWpy](#) for signal-processing and much of existing functionality of these packages as possible. Novel functi

Development process is steered by regular [meetings](#) with participants ac

The project is open for contributions, and feedback on usage is welcome [creating an Issue](#). For other enquiries contact details are provided below.

Contact

- Artem Basalaev (artem.basalaev@physik DOT uni-hamburg.de)
- Christian Darsow-Fromm (cdarsow@physnet DOT uni-hamburg.de)
- Oliver Gerberding (oliver.gerberding@physik DOT uni-hamburg.de)
- Martin Hewitson (martin.hewitson@taei DOT mpg.de)

Visit us at <https://gitlab.com/pyda-group/spicypy>!

Tutorial

Is in the documentation, entry point is here: <https://pyda-group.gitlab.io/spicypy/install.html>

1. Install

Installation

To install, simply do

```
pip install spicypy
```

Note: on Windows the above might not work, you may see a compilation error in `ligo-segments` package. If this happens, please install `GWpy` separately first, by running

```
conda install -c conda-forge gwpy
```

in your terminal. (It must be an Anaconda environment, unfortunately currently there's no easy way to install on Windows without Anaconda. We are working on that)

← Previous

Next →

2. Try examples

Introduction

Installation

Example 1: Feed a signal through a simple control system

Example 2: Time-match signals and estimate coherence

Example 3: Transfer function between time series and Daniell averaging method

Example 4: Plot Einstein Telescope alignment sensing noise and make bode plots

Example 5: Model active seismic isolation ("VATIGrav" platform)

API REFERENCE

Spicypy API

3. Look into code documentation to see what's possible; apply to your problem

Thank you!