

Locust “Fake Event/Track” Generator Tutorial

Penny Slocum & Luis Saldaña

11/01/18

Generating pre-defined tracks with Locust

- Purpose: Generate custom Project 8 CRES-like events/tracks within Locust to be used for further analysis (e.g track finding with deep learning)
- A “fake” track Locust generator has been added to Locust (LMCFakeTrackSignalGenerator) with register name “fake-track”
- Track characteristics can be specified from a config file
 - Template config: https://github.com/project8/locust_mc/Config/Tutorial/LocustFakeTrack.json
- This will produce an egg file which may be processed with Katydid
 - Template config: https://github.com/project8/locust_mc/Config/Tutorial/katydid_faketrack.json

Parameters in Locust config file

```
{  
  "generators":  
  [  
    "fake-track",  
    "lpf-fft",  
    "decimate-signal",  
    "gaussian-noise",  
    "digitizer"  
  ],  
  "fake-track":  
  {  
    "signal-power": 1.0e-15,  
    "start-frequency-max": 20.053e9,  
    "start-frequency-min": 20.049e9,  
    "start-vphase": 0.0,  
    "slope-mean": 0.6,  
    "slope-std": 0.025,  
    "start-time-min": 0.001,  
    "start-time-max": 0.003,  
    "lo-frequency": 20.0e9,  
    "track-length-mean": 0.001,  
    "ntracks-mean": 2.0,  
    "random-seed": 0  
  },  
}
```

- Track parameters are randomly drawn from probability distribution functions PDF (see README)
- A given 'random-seed' corresponds to unique track parameter set, i.e unique event
- Note: An event with multiple scattered tracks may be built by setting 'ntracks-mean' > 1

← Signal power (W)

← Starting voltage phase (rad)

← LO oscillator frequency (Hz)

Parameters in Locust config file

```
"gaussian-noise":  
{  
  "noise-floor": 2.7e-21,  
  "domain": "time"  
},
```

← Noise floor (W/Hz)

```
"digitizer":  
{  
  "v-range": 1.0e-4,  
  "v-offset": -0.5e-4  
}
```

← Digitizer range (V)

← Digitizer offset (V)

Note: Don't saturate digitizer!

Generating fake event/track

- Two methods:
 1. Using Locust directly with single event/track config:
https://github.com/project8/locust_mc/Config/Tutorial/LocustFakeTrack.json
 2. Using multi-simulation python script:
https://github.com/project8/locust_mc/blob/develop/Config/Tutorial/LocustFakeTrack.py
- Let's go through each one :)

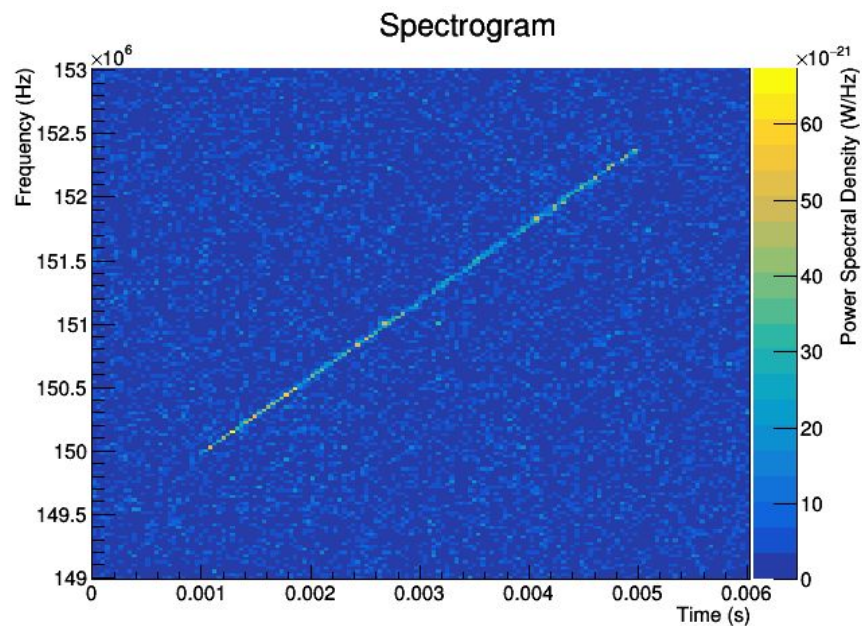
1. Single fake event/track

- Edit the Locust config file to specify the egg file output

```
"simulation":  
{  
    "egg-filename": "/path/to/output/locust_faketrack.egg";  
    ...  
}
```

- Generate single fake-track egg file:
 - /path/to/LocustSim config=/path/to/LocustFakeTrack.json
- Katydid processing to obtain 2D histogram of PSD values
 - /path/to/Katydid -c /path/to/katydid_faketrack.json -e /path/to/locust_faketrack.egg --waterfall-writer.output-file="/path/to/output/locust_faketrack_waterfall.root"
- Output root file may be processed with ROOT macro to plot waterfall picture:
 - https://github.com/project8/locust_mc/Config/Tutorial/PlotFakeTrackImages.c
 - Use PlotImages()

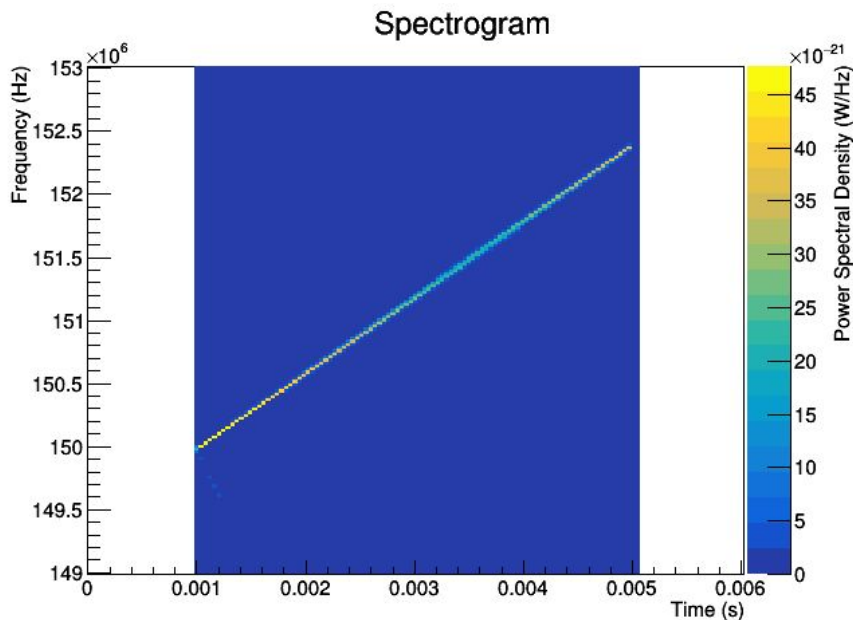
1. Single event/track: Example with Gaussian noise



- Signal PSD = $1\text{e-}15/(200\text{e}6/8192)$
 $= 40\text{e-}21 \text{ W/Hz}$
- Noise PSD = $2.7\text{e-}21 \text{ W/Hz}$
- SNR = 15
- Start frequency of 20.05 GHz - 20 GHz
= 50 MHz shifted by +100 MHz due to processing with RSA settings in Katydid config
- Slice size = 8192

1. Single event/track: Example without Gaussian noise

- We can remove the noise by editing the LocustFakeTrack.json config:
 - Remove "gaussian-noise" Locust generator



2. Multi-simulation script

- Must have Locust AND Katydid installed (and Python)
- Runs N simulations. For n^{th} iteration:
 - Creates two Locust config files: with and without "gaussian-noise"
 - Runs Locust for each creating two egg files:
 - `'locust_faketrack_*.egg'`
 - `'locust_faketrack_wnoise_*.egg'`
 - Each egg file contains same event. Made possible by fixing the random seed for the PDFs
 - Removes Locust config files
 - Process two Locust egg files with Katydid `katydid_faketrack.json` template config. Creates two root files contain 2D PSD histograms, i.e “waterfall” spectrograms
 - `'locust_faketrack_*.root'`
 - `'locust_faketrack_wnoise_*.root'`

2. Multi-simulation script

- Number of simulations can be specified at command line as well as a number of other params:

```
les67@les67:~/locust_mc/Config/Tutorial$ python LocustFakeTrack.py --help
usage: LocustFakeTrack.py [-h] [-w WORKING_DIR] [-l LOCUST_BIN]
                        [-k KATYDID_BIN] [-c CONFIG]
                        n_sims

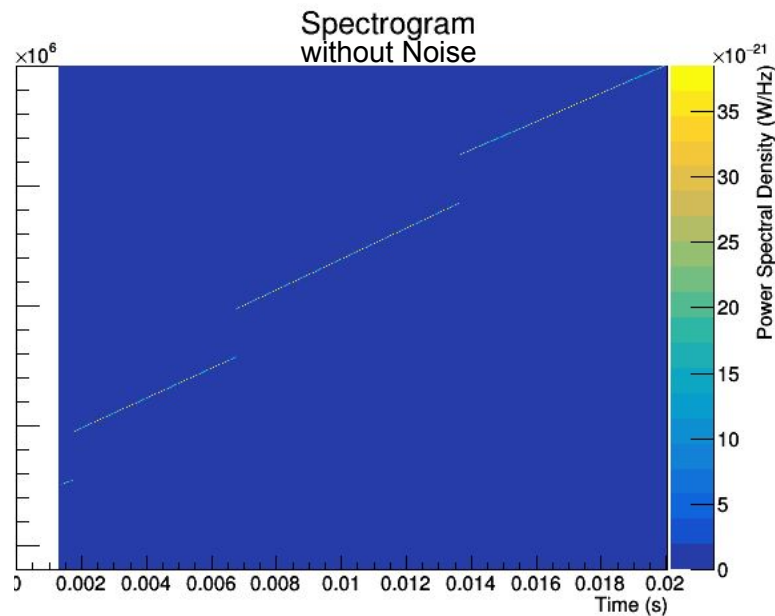
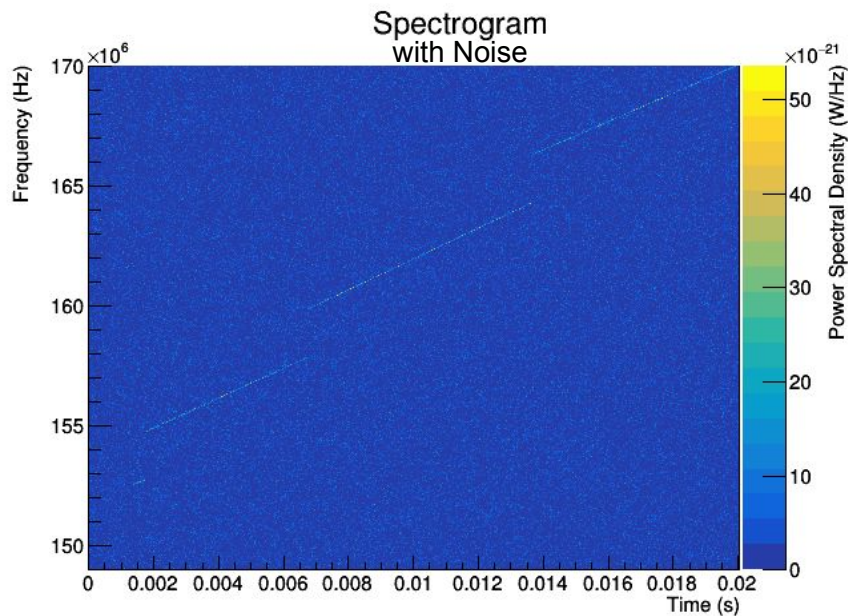
generate fake tracks in Locust and obtain Katydid waterfall spectrograms

positional arguments:
  n_sims                Number of simulations

optional arguments:
  -h, --help            show this help message and exit
  -w WORKING_DIR, --working_dir WORKING_DIR
                        Path to working directory to save egg and root files
  -l LOCUST_BIN, --locust_bin LOCUST_BIN
                        Path to Locust binary
  -k KATYDID_BIN, --katydid_bin KATYDID_BIN
                        Path to Katydid binary
  -c CONFIG, --config CONFIG
                        Path to Katydid config file
```

2. Multi-simulation script: Example Event

- Running 1 simulation and generating pictures using PlotFakeTrackImages.c
- The same event can be simulated as we fix the random seed for the PDFs in each iteration of the simulation



Generate away!