Generating non-white transverse noise in COMBIp

Juan Sebastián Pereira Cubillo

Universidad de Costa Rica

Supervisor: Sondre Vik Furuseth

Monday 19$^{\text{th}}$ August, 2019

Motivation

- Particle beams in hadron colliders are subject to several deteriorating mechanisms. These mechanisms lead to a reduction of the beam quality and can lead to beam instabilities.
- I have studied the impact of external noise sources on the beam, randomly kicking the particles transversely.
- I have focused on the stability of multi-bunch modes in a machine with noise of a given frequency spectrum.

Understanding the physics of the problem

Phase Advance

Bunches rotate in phase space due to the linear magnets. Octupole magnets an also be used to create a spread in the phase advance per turn of the individual particles, to avoid instabilities.

Damper

Active beam feedback system that attempts to reduced the measured centroid offset of individual bunches.

Noise

Could be either coherent or incoherent, correlated or uncorrelated.

Implementation of noise in COMBIp

1. Create action in COMBIp to generate non-white noise and apply it to every bunch (action 7, type 5).
2. Utilize previously implemented Fortran functions to generate the Fourier spectrum of white noise (flat and random)
3. Multiply the random spectrum by a weight function to get the desired power spectral density of the noise.
4. Utilize an inverse FFT to get the noise values in time domain
5. Separate noise and distribute it to each bunch
6. Apply the noise each turn

Algorithm for noise generation

Code outline

```
generate random x and y;
z = x + i*y;
NoiseBig = Inverse Fourier transform [z*weight(freq)];
MPI_Type_vector(NoiseBig, slots, &bunch_noise)
MPI_Isend(bunch_noise);
MPI_Receive(bunch_noise);
for every turn:
    for every bunch in the beam:
        apply bunch_noise to each particle in the bunch
```

Results

White noise

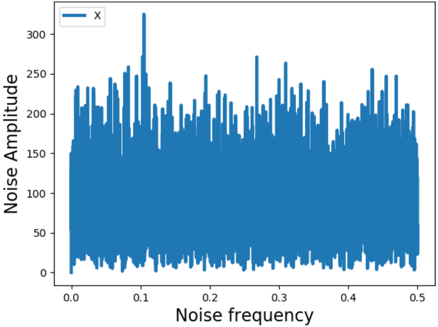

Figure: Frequency spectrum

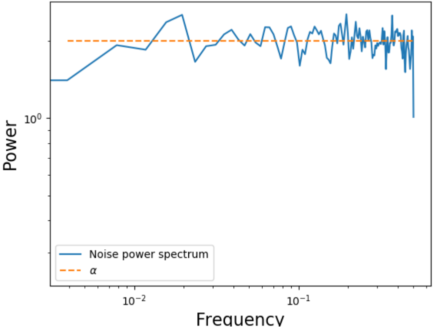

Figure: Power density spectrum

The algorithm can generate white noise (test case)

Results

Power density Spectrums

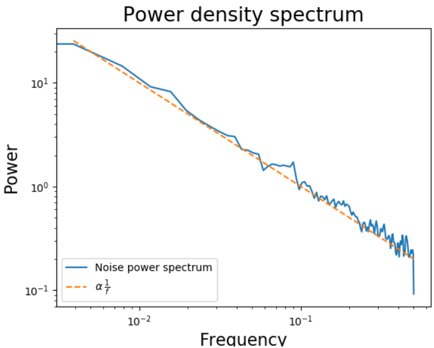

Figure: Pink noise

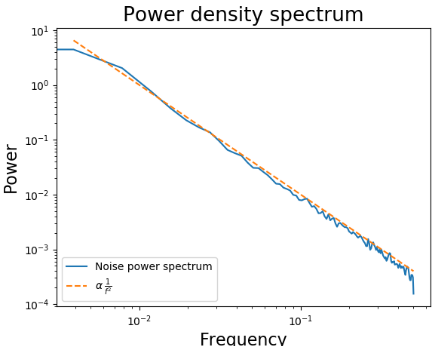

Figure: Brown noise

The algorithm is also capable to generate colored noise or any other weight function you ask it to use.

Parameters of simulation

Frequency and slots

For LHC, the number of slots is 3564 and the bunch spacing is 25 ns, thus sampling frequency is $\frac{1}{25\ \mathrm{ns}}$ and the number of noise samples is $turns * slots$. For clearer graphs, slots were utilized for the simulation, with a bunch spacing of 8910 ns.

Parameter	**Value**
Q_x	0.31
Q_y	0.32
g_x	0.02
g_y	0.02
Macroparticles per bunch	1E6
Turns	1000
Noise amplitudes	0.002 σ_p
Bunches beam 1	10
Bunches beam 2	0

Results

Multibunch PSD for a simulation with 10 equally spaced bunches in the LHC

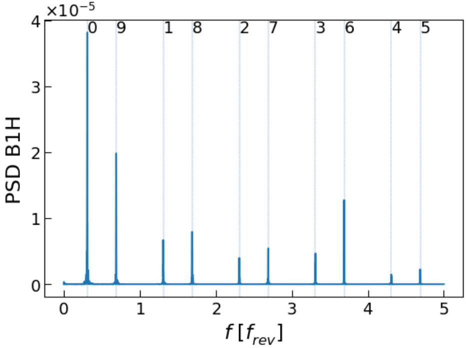

Figure: Pink noise

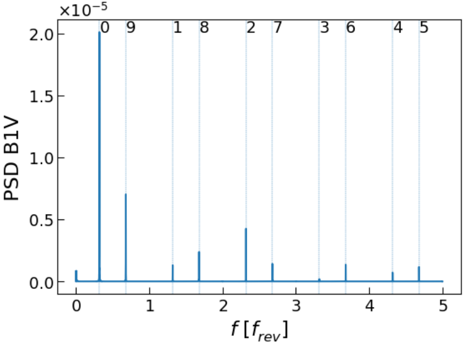

Figure: Brown noise

The excitation of the modes by the noise decrease for higher frequencies.

Future work

- As seen on the previous slide, the impact of pink and brown noise on the beam is smaller for higher frequencies. The current model of the noise in the LHC is approximately like pink noise, being weaker at higher frequencies.

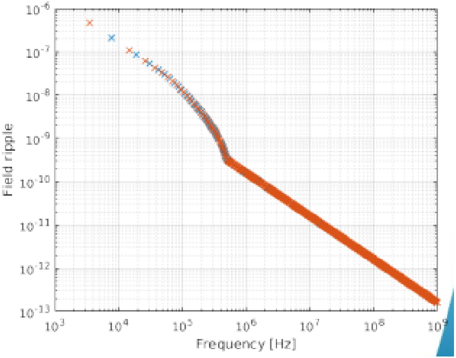

Figure: LHC noise spectrum from quadrupole field ripples

Future work

1. Develop another COMBIp module that simulates the effects of a transverse damper with limited bandwidth.
2. Simulate the evolution of the beam for given noise spectrum and certain damper bandwidth. Include more physical effects and study the efficiency of the code.

References

X. Buffat, S.V. Furuseth, D. Gamba (2018)
Emittance growth due to decoherence of external excitations in the LHC and HL-LHC
WP2 meeting – 10.04.2018

X. Buffat, D. Amorim, S. Antipov, L. Carver, N. Biancacci, S.V. Furuseth et al. (2018)
The impact of noise on beam stability
8th HL-LHC collaboration meeting – 11.10.2018

S.V. Furuseth, X. Buffat. (2019)
Parallel high-performance multi-beam multi-bunch simulations
Computer Physics Communications 0010-4655

H. Schmickler. (2018)
Multi-Bunch Feedback Systems
CERN, CAS 2018

Equations to model different physical effects

Phase Advance

$$\begin{bmatrix} x \\ Px \end{bmatrix}_{out} = R_{2D}(\Delta U) \begin{bmatrix} x \\ Px \end{bmatrix}_{in} \textit{ where } \Delta U = \begin{bmatrix} 2\pi Q_x \\ 2\pi Q_y \end{bmatrix} \textit{ and } \begin{bmatrix} Q_x \\ Q_y \end{bmatrix} = \begin{bmatrix} 0.31 \\ 0.32 \end{bmatrix}$$

Damper

$$\begin{bmatrix} x \\ y \end{bmatrix}_{out} = \begin{bmatrix} x \\ y \end{bmatrix}_{in} - \begin{bmatrix} g_x & 0 \\ 0 & g_y \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}_{in} \textit{ where } g_x \textit{ and } g_y \textit{ are damping parameters}$$

Noise

$$P_{out} = Px_{in} + \delta \ (\textit{same } \delta \textit{ for each } P \textit{ if coherent, different if incoherent})$$

First python simulations

$$2000\ particles,\ \ g_x = g_y = 0.02,\ \ no\ octupole\ tune,\ \ 1000\ turns$$

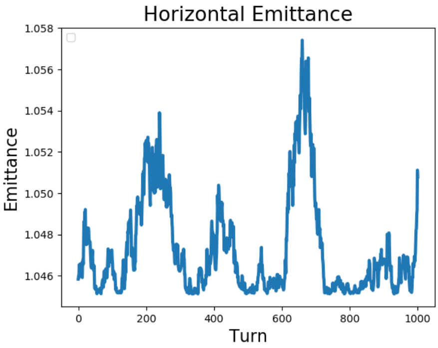

Figure: emittance coherent noise

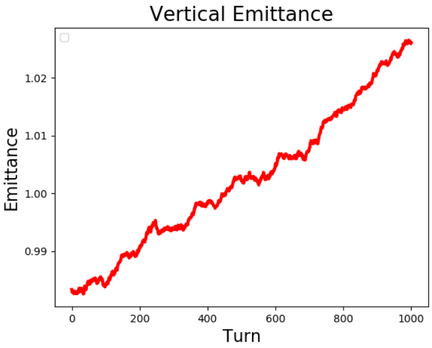

Figure: emittance incoherent noise

Emittance doesn't vary that much for coherent noise, but increases for incoherent noise.

First python simulations

$$2000 \; particles, \;\; g_x = g_y = 0.02, \;\; no \; octupole \; tune, \;\; 1000 \; turns$$

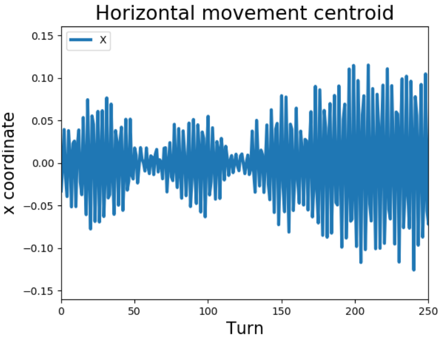

Figure: coherent noise

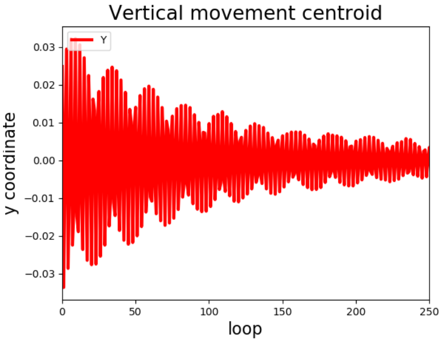

Figure: incoherent noise

Transverse movement of the centroid gets better damped for incoherent noise.

Action for noise generation

Action for noise generation

5	1	100	0	0.002	0.002	0.002	0.002
type	correlation	turns	type	beam and plane amplitudes			

- If uncorrelated noise (0), each bunch generates its own noise independently.
- Action tells for how many turns noise is created. If not specified, noise is created for entire run.
- New noise is created after the specified amount of turns
- Type indicates the kind of noise. Specific functions can also be implemented
- If some of the amplitudes are not defined, the program doesn't generate noise for them
- All generated noise is stored in the output directory