On-line tool to calculate the expected heat load from impedance (and synchrotron radiation)

Benoit, Giovanni and Gianni
with the help of

Elias Metral, Michael Schenck, Carlo Zannini
and BE-CO

Available options for impedance

- Implement existing full scripts in a dedicated display

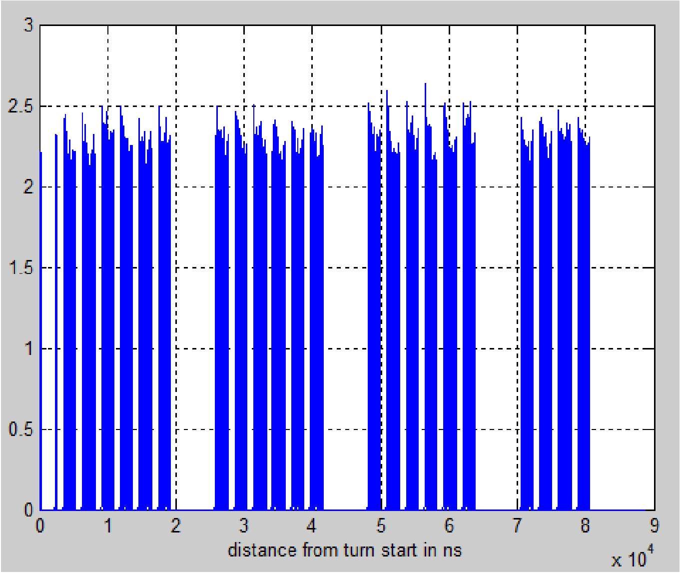
  → existing Matlab and python scripts already used by Carlo and Michael for scrubbing and SPS MKP heating

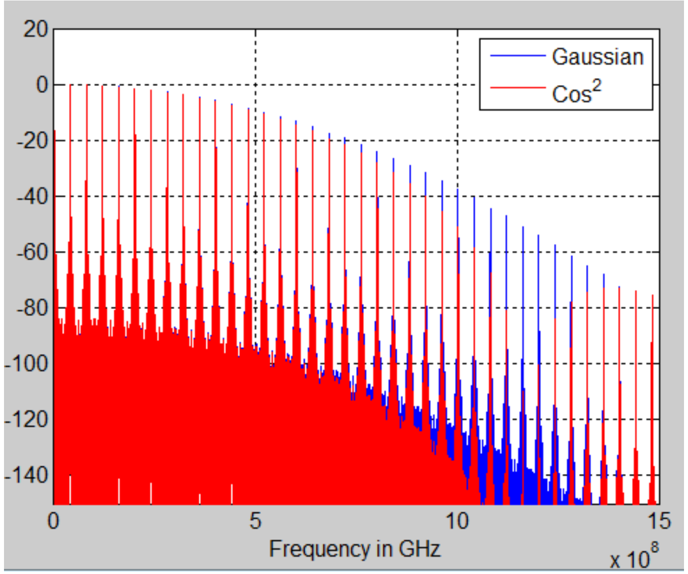
  →Matlab script used for LHC beam induced heating

- Implement simple formula in Timber and use the existing framework to display it in the fixed display.

Full scripts

- reconstruct the beam intensity profile *(from bunch by bunch intensities and bunch lengths stored in Timber by the FBCT and BQM)*
- Compute the FFT to obtain the beam spectrum
- Interpolate the impedance and multiply in frequency domain to obtain the power loss
- Need to assume a distribution (e.g. Gaussian, cos^2)





Full scripts

- An alternative could be to use directly the beam spectrum that is also stored in Timber (but has not been working well in 2015 due to issues with the spectrum analyzers)
- Interpolate the impedance and multiply in frequency domain to obtain the power loss

→ No need to assume a distribution, but could be also more affected by noise

→ Proposal: implement both computations before high intensity in 2016, with priority to the first option

Full scripts

- Requires implementation to be available in the CCC
- Should use the framework that is being put in place by Riccardo and Lee for other tools
- Unclear whether this can be available for the scrubbing run

“Simple” power loss formula

- See for instance, A. Chao’s book or G. Rumolo and E. Metral at USPAS course in 2010
- Assumes Gaussian bunch and thick wall formula for the resistive impedance.

$$\frac{Z_{rw}^{||}(\omega)}{L} = \sqrt{\frac{2}{\sigma Z_0 c}}\frac{Z_0}{4\pi b}|\omega|^{\frac{1}{2}}[1 - i\,\mathrm{sgn}(\omega)] \Rightarrow \frac{\Delta E}{L} = -\frac{N_b^2 e^2 c}{4\pi^2 b \sigma_z^{3/2}}\sqrt{\frac{Z_0}{2\sigma}}\cdot\Gamma\left(\frac{3}{4}\right)$$

Giovanni (USPAS course)

- Assumes uniform filling.

- **Assuming a Gaussian bunch**

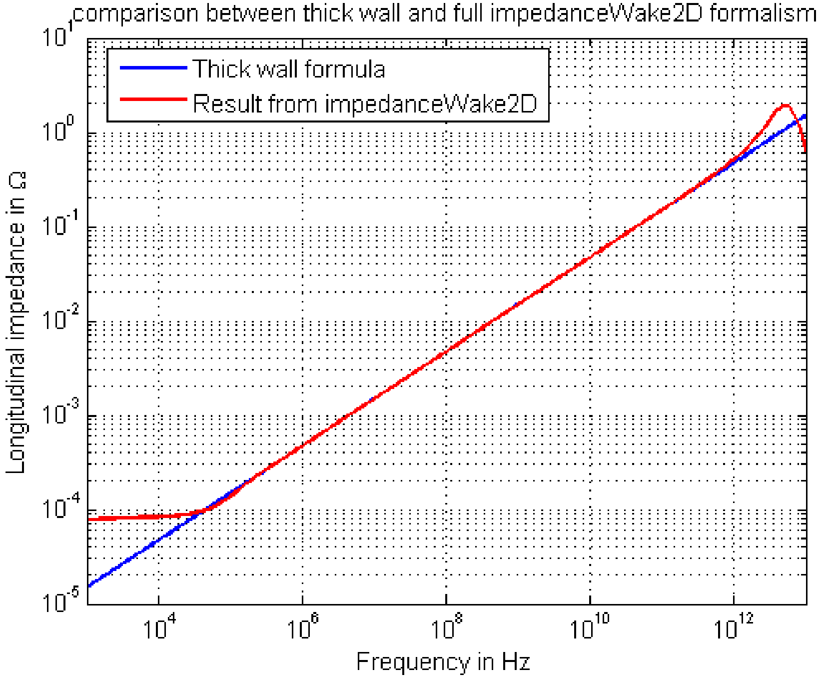
$$P_{loss/m}^{G,RW,1\,\mathrm{layer}} = \frac{1}{2\pi R}\Gamma\left(\frac{3}{4}\right)\frac{M}{b}\left(\frac{N_b\, e}{2\pi}\right)^2\sqrt{\frac{c\rho Z_0}{2}}\,\sigma_t^{-3/2} \approx 85\,\mathrm{mW/m}$$

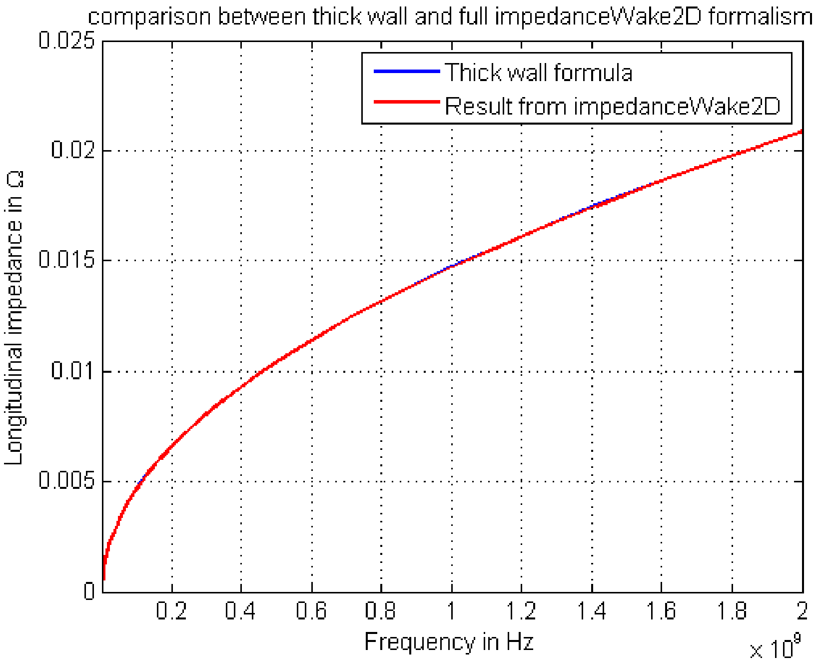
Elias (Chamonix 2012)

- Assumes a factor 1.4 for the weld (C. Zannini).

“Simple” power loss formula

- What error do we do in assuming the thick wall formula?



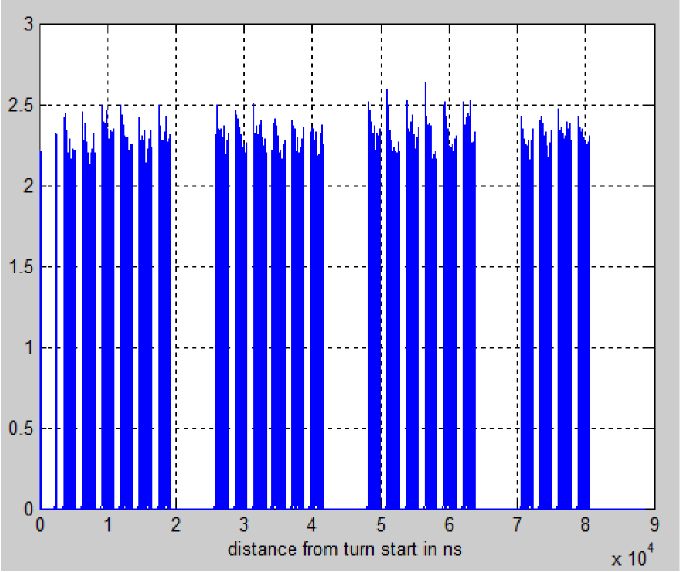


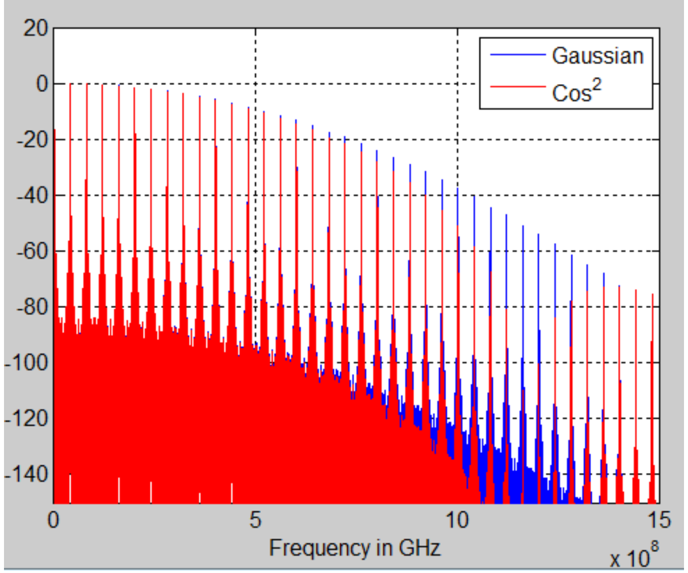
→ No visible difference in the frequency range of interest for the case of copper coated stainless steel beam screen

“Simple” power loss formula

- What error do we do with that simple formula?

→Example of fill 4467: 1608 bunches, 1.02e11 p/b





→full formula with constant bunch length and intensity: Ploss=329.5 mW/m

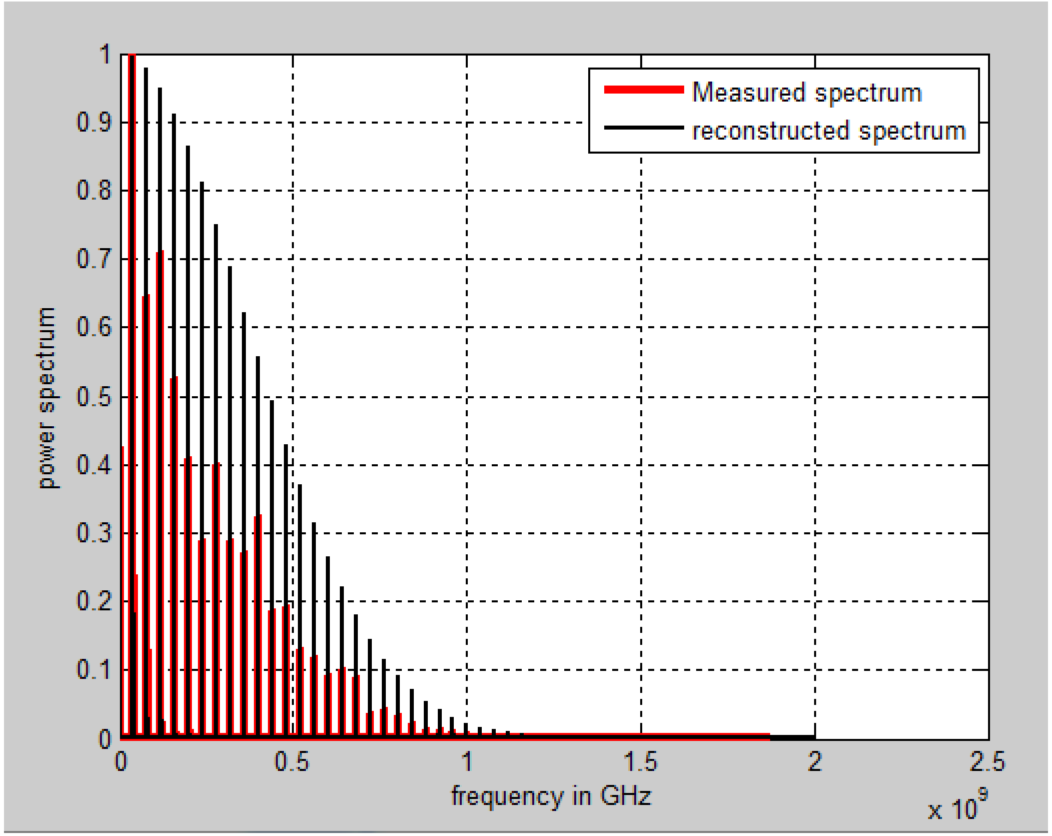
→full formula with constant bunch length and meas. intensity: Ploss=330 mW/m

→full formula with measured bunch length and intensity: Ploss=331 mW/m

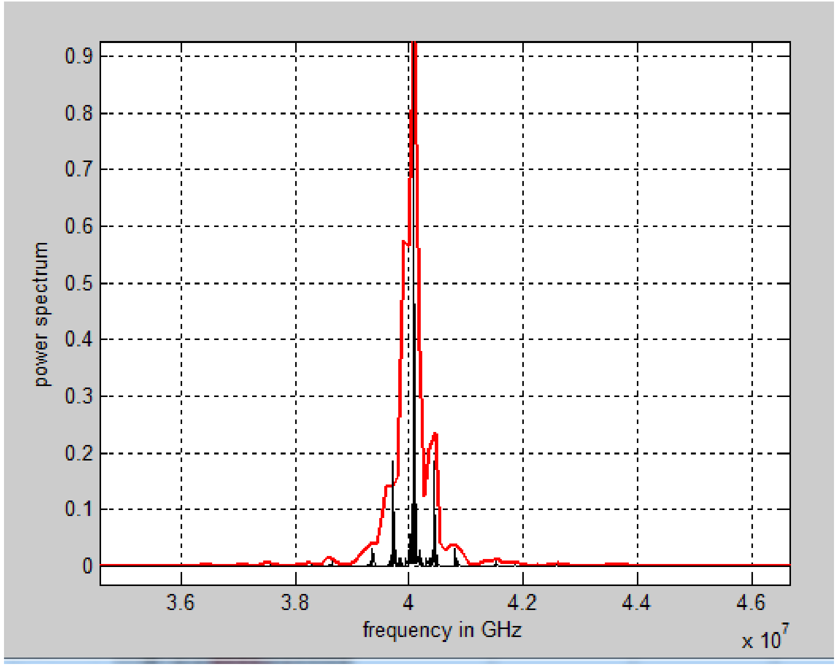
→Simple formula: Ploss=329 mW/m

→Formula using the measured spectrum: Ploss= 400 mW

Comparison between measured and reconstructed spectra



Zoom of the first line



→measured peaks are smaller and wider. One should try and increase the resolution next year

Simple formula

- An alternative could be to use directly the beam spectrum that is also stored in Timber (but has not been working well in 2015 due to issues with the spectrum analyzers)
- Interpolate the impedance and multiply in frequency domain to obtain the power loss

→ No need to assume a distribution, but could be also more affected by noise

→ Proposal: implement simple formula fast in Timber and use existing fixed display framework used for heatload

Implementation in Timber

From the following existing variables:

- LHC.BCTDC.A6R4.B1:BEAM_INTENSITY [Int1]
- LHC.BCTDC.A6R4.B2:BEAM_INTENSITY [Int2]
- LHC.BSRA.US45.B1:ABORT_GAP_ENERGY [E]
- LHC.BQM.B1:NO_BUNCHES [N1]
- LHC.BQM.B2:NO_BUNCHES [N2]
- LHC.BQM.B1:BUNCH_LENGTH_MEAN [len1]
- LHC.BQM.B1:BUNCH_LENGTH_MEAN [len2]

We would like to compute following virtual variables

LHC.QBS_CALCULATED_ARC_IMPED.B1 [Pimp1] in W/half cell
LHC.QBS_CALCULATED_ARC_IMPED.B2 [Pimp2] in W/half cell
LHC.QBS_CALCULATED_ARC_SYNCH_RAD.B1 [Psr1] in W/half cell
LHC.QBS_CALCULATED_ARC_SYNCH_RAD.B2 [Psr2] in W/half cell
LHC.QBS_CALCULATED_ARC.TOTAL [Ptot] in W/half cell

Timescaled in a fixed interval (average over 1 minute).

- The formulas are (in “Matlab” style):

```
Pimp1 =  0.4601e-39 * (len1/4)^-1.5 * Int1^2 * N1^-1 *sqrt(1.0048+ 3.1540e-04 * E)
Pimp2 =  0.4601e-39 * (len2/4)^-1.5 * Int2^2 * N2^-1 *sqrt(1.0048+ 3.1540e-04 * E)
Psr1 = Int1* 3.2292e+14^-1 * 0.33 * 53.4 * (E/7000)^4
Psr2 = Int2* 3.2292e+14^-1 * 0.33 * 53.4 * (E/7000)^4
Ptot= Pimp1 + Pimp2 + Psr1 + Psr2
```

Summary

- Two options for getting beam screen heat load:
  - Full formula implementation will require development
  - Simple formula should profit from the framework of Timber variables used by Fixed displays, which is already available since 2015
    - The implementation should be ready before Easter, thanks to the help of BE-CO (Nikolay Tsvetkov)