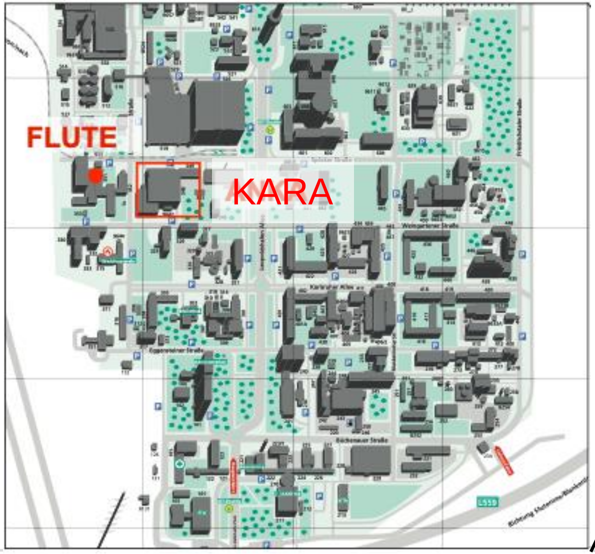
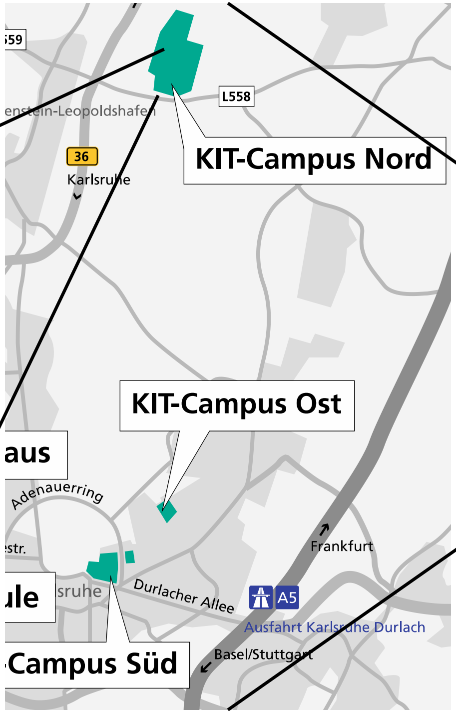
Induced Voltage with uneven Sampling

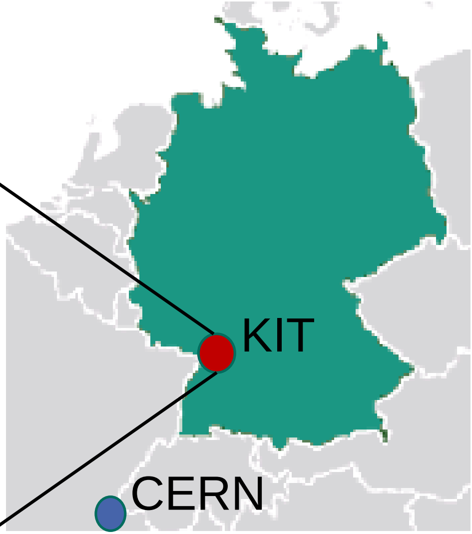
Markus Schwarz

Karlsruhe Institute of Technology

- University (23,000 Students)
- National Research Center
  - 9600 Employees







Motivation

- In many machines bunch-to-bunch distance exceeds bunch length
  - LHC beam in SPS: one bunch every 5 RF-buckets, 25 ns > 3 ns
  - KARA: 2 ps bunches in 2 ns RF-buckets
- Simulation of multi-bunch instabilities become challenging when uniform sampling is used because of the many empty bins

Basics of Induced Voltage Calculation

- To compute $V_{ind}(\Delta t) = -2\, q\, N_p \Re \int_0^\infty Z(f)\Lambda(f)e^{2\pi i f \Delta t}$ need
  - Line density $\lambda(\Delta t)$ and its Fourier transform $\Lambda(f)$
  - Compute inverse Fourier transform

Object	Impedance type	Method	Sampling	Complexity
InducedVoltageFreq	Any	Circular convolution	Uniform	N log N
InducedVoltageTime	Any	Linear convolution	Uniform	N log N
InducedVoltageResonator	Resonator	Matrix multiplication	Non-uniform	N^2
InducedVoltageSparse	Any	Matrix multiplication	Non-uniform	N^2

Fourier Transform of non-uniform Data Points

- N data points (x_j, y_j) not necessarily equidistant
- Continuous function $y(x)$ from (x_j, y_j) by linear interpolation
- Fourier transform of $y(x)$ at arbitrary wave number k given by

$$Y(k) = \frac{1}{k^2} \sum_{j=0}^{N-2} \frac{y_{j+1} - y_j}{x_{j+1} - x_j} \left(e^{-ikx_{j+1}} - e^{-ikx_j}\right)$$

slope of line segments

- Note: Interpolation does not have to be carried out

Algorithm to compute V_{ind}

- Given N data points (t_j, λ_j) from bunch profile and impedance $\mathrm{Z}(\omega)$
- $V_{ind}(t_j) = -2\, q\, N_p \Re \int_0^\infty Z(f)\Lambda(f)e^{2\pi i f t_j}$
  - Compute Fourier transform $\Lambda(\omega)$ from linear interpolation
  - Evaluate integrand $\mathrm{Y}(\omega) = \mathrm{Z}(\omega)\Lambda(\omega)$ at M arbitrary, convenient frequencies ω_m to obtain data points in frequency domain (ω_m, Y_m)
  - Inverse Fourier transform of (ω_m, Y_m) from linear interpolation gives $\mathrm{V}_{\mathrm{ind}}(t_j)$
- Swap summations over m (frequency) and n (time)
  $\Rightarrow N \times (N-1)$ “wake” matrix $\mathcal{W}_{j,n}$

Expression for V_{ind}

- $\mathrm{V}_{\mathrm{ind}}(t_j)$: matrix multiplication of "wake" matrix $\mathcal{W}_{j,n}$ and profile difference vector $\lambda_{n+1} - \lambda_n$

$$V_{ind}(t_j) = \sum_{n=0}^{N-2} \mathcal{W}_{j,n} \left(\lambda_{n+1} - \lambda_n\right)$$

constant

changes each turn

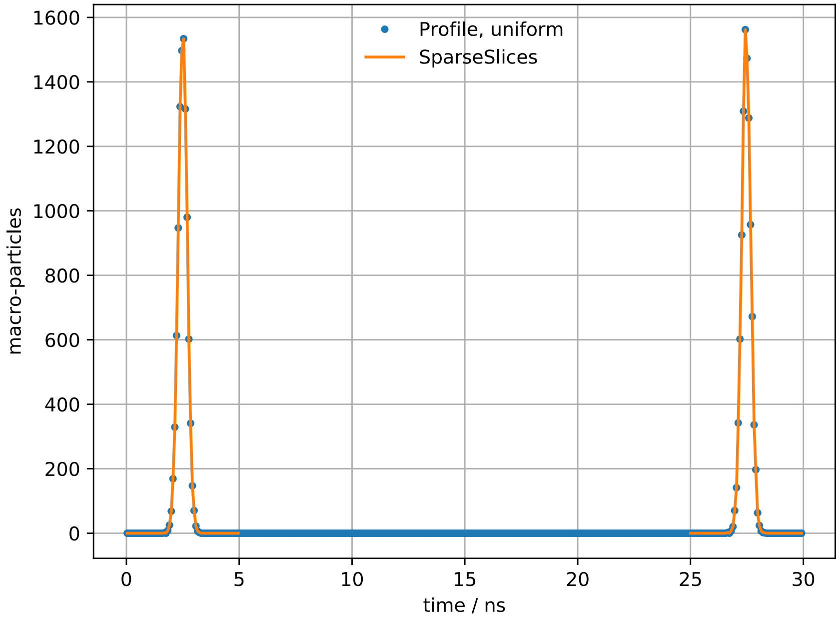
- "Wake" matrix $\mathcal{W}_{j,n}$ needs to be computed only once
- Only need bunch profile λ_n (not its Fourier transform)
- Computation time is $\mathcal{O}(N^2)$

InducedVoltageSparse Syntax

- Algorithm implemented in BLonD on my fork
- `InducedVoltageSparse(Beam, profile_object, impedance_source_list, init_frequency_array, adaptive_frequency_sampling=False)`
- `profile_object` can either be a `Profile` or `SparceSlices` object
- `init_frequency_array`: (initial) frequencies at which to sample integrand (ω_m, Y_m)
- If `adaptive_frequency_sampling` is `True`, integrand is sampled adaptively (only at initialization)

Sparse Profile in BLonD

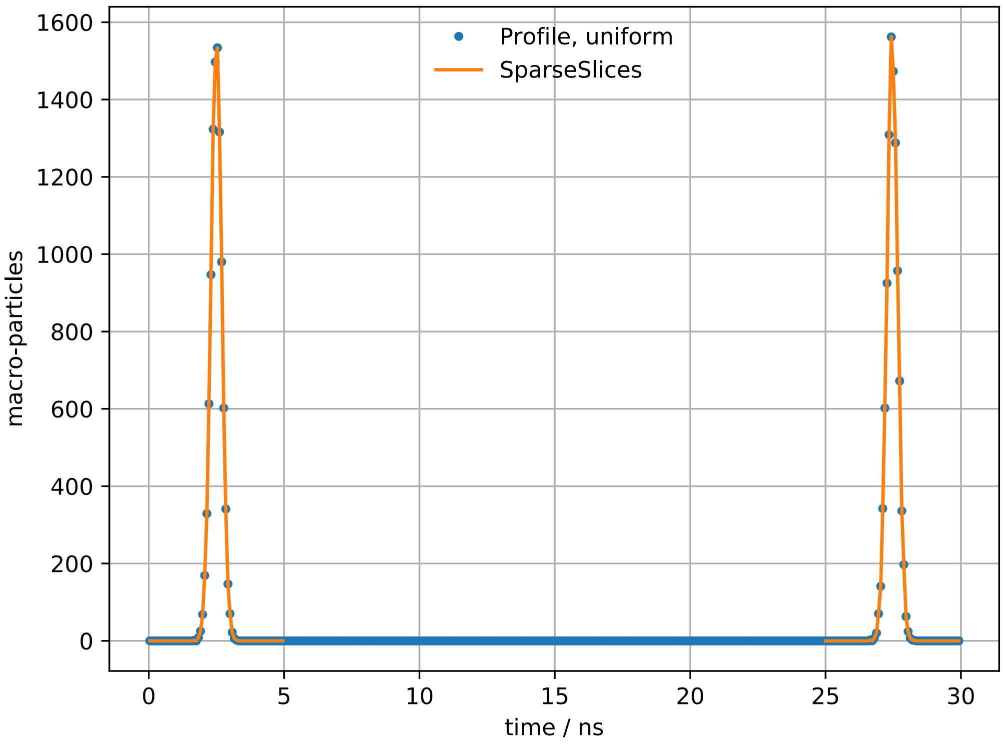
- BLonD already has `SparseSlices`* object, to compute profile only at selected RF buckets



*Rename to `SparseProfile`?

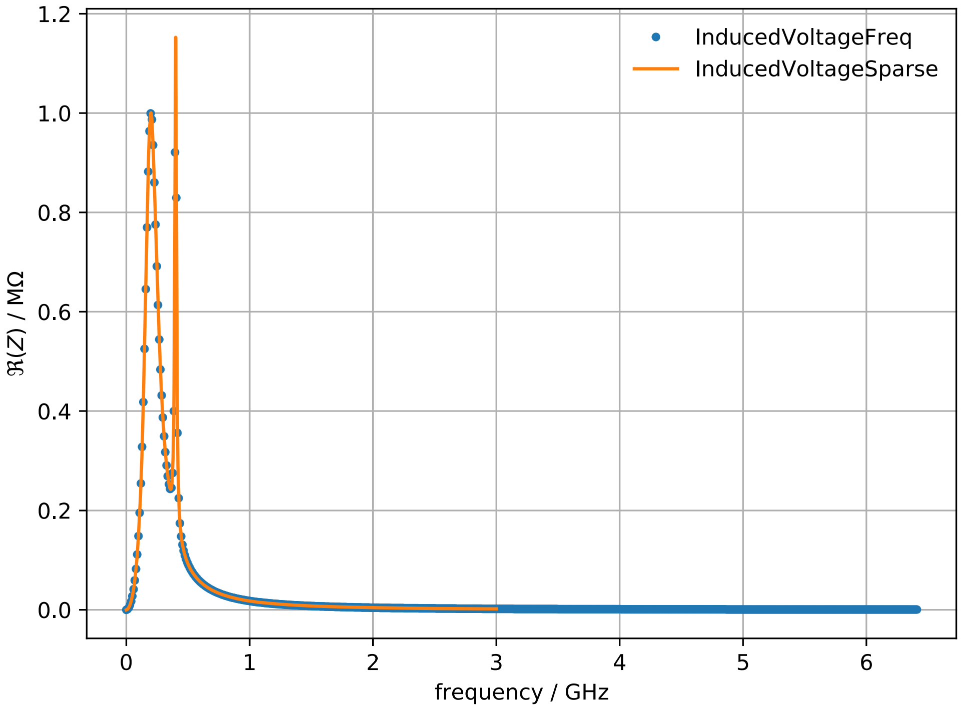
Example: Profile

- 2 Gaussian bunches spaced 5x5 ns apart
- Use either `SparseSlices`* or `Profile` object

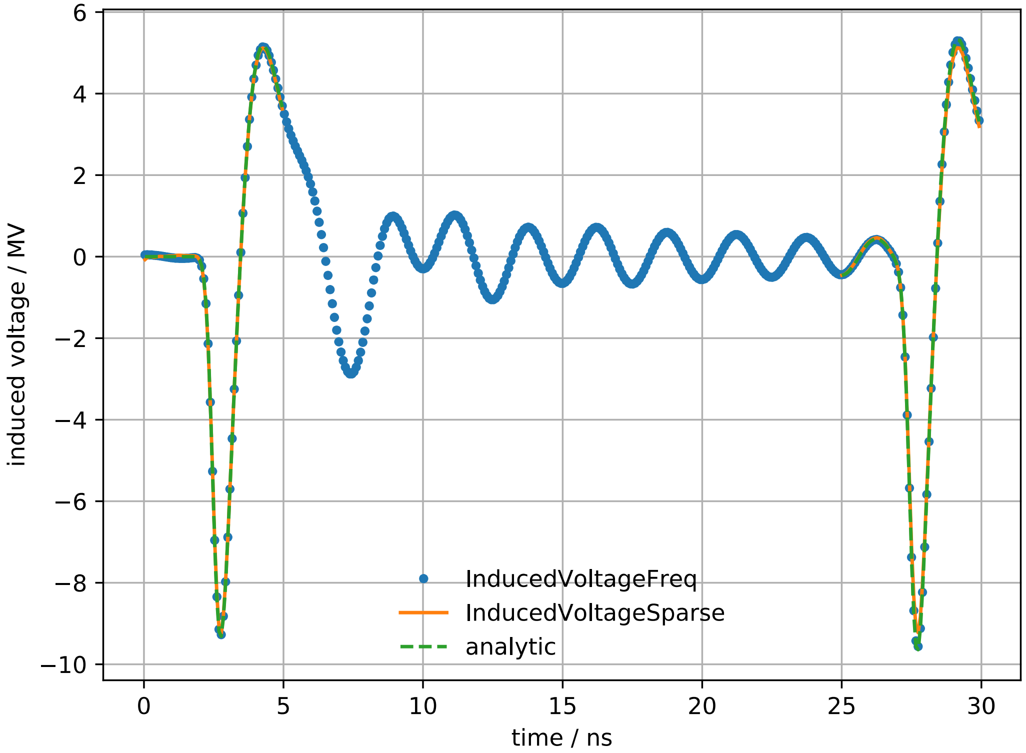


Example: Impedance

- Two resonators
- Adaptive frequency sampling was used for `InducedVoltageSparse`



Example: Induced Voltage



Example: RunTime

- From `%timeit -r 500 -n 10 xxx.induced_voltage_1turn()`

InducedVoltageObject	Run Time
InducedVoltageFreq	22.4 µs ± 3.47
InducedVoltageSparse	21.4 µs ± 8.71

- Similar runtimes for this example

Summary and Outlook

- Introduced InducedVoltageSparse object that computes the induced voltage for arbitrary impedance and non-uniform binning.
  - Computation in frequency domain needs to done only once
  - Can use adaptive frequency sampling
- Similar run time compared to InducedVoltageFreq
- **Still needs bugs fixing!**

Thank you for your attention!

Computation Details I

- Fourier transform of bunch profile at frequency ω_m

$$\Lambda_m = \frac{1}{\omega_m^2} \sum_{n=0}^{N-2} \frac{\lambda_{n+1} - \lambda_n}{t_{n+1} - t_n} \left(e^{i\omega_m t_{n+1}} - e^{i\omega_m t_n}\right)$$

- Integrand $\mathrm{Y_m} = \mathrm{Y}(\omega_m)$ then given by

$$Y_m = Z_m \Lambda_m = \frac{Z_m}{\omega_m^2} \sum_{n=0}^{N-2} \frac{\lambda_{n+1} - \lambda_n}{t_{n+1} - t_n} \left(e^{-i\omega_m t_{n+1}} - e^{-i\omega_m t_{n+1}}\right)$$

Computation Details II

- Inverse Fourier transform of (ω_m, Y_m) from linear interpolation gives $\mathrm{V}_{\mathrm{ind}}(t_j)$

$$V_{ind}(t_j) = \frac{qN_p}{2\pi t_j^2} \sum_{m=0}^{M-2} \frac{Y_{m+1} - Y_m}{\omega_{m+1} - \omega_m} \left(e^{i\omega_{m+1}t_j} - e^{i\omega_m t_j}\right)$$

- Lengthy expression, but factor $(\lambda_{n+1} - \lambda_n)$ can be factored out and summations over m (frequency) and n (time) can be swapped.

Computation Details III, "Wake" matrix

Karlsruhe Institute of Technology

- $N \times (N-1)$ "Wake" matrix $\mathcal{W}_{j,n}$

$$\mathcal{W}_{j,n} = \frac{qN_p}{2\pi t_j^2} \frac{1}{t_{n+1} - t_n} \sum_{m=0}^{M-2} \frac{1}{\omega_{m+1} - \omega_m} \left[\frac{Z_m}{\omega_m^2} \left(e^{-i\omega_m t_n} - e^{-i\omega_m t_{n+1}} \right) - \frac{Z_{m+1}}{\omega_{m+1}^2} \left(e^{-i\omega_{m+1} t_n} - e^{-i\omega_{m+1} t_{n+1}} \right) \right]$$

- Only depends on constant parameters
  $\Rightarrow$ needs to be computed only once!