ASTRA BASED SWARM OPTIMIZATIONS OF THE B*ERL*inPro INJECTOR

M. Abo-Bakr

FRAAC4

Content:

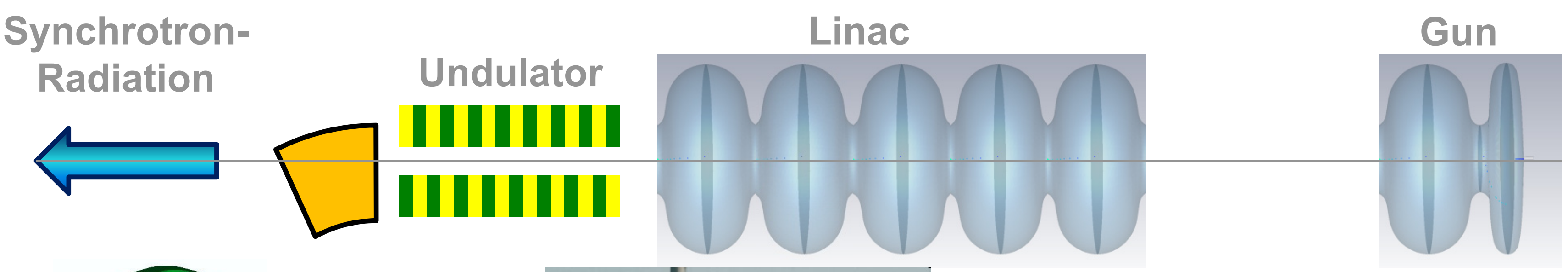
- Introduction ERL´s
- Introduction B*ERL*inPro
- Motivation
- Computational Aspects
- Optimization of the B*ERL*inPro Injector
- potential, future applications

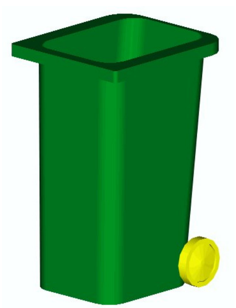
Introduction ERL's

Storage Rings: beam properties = equilibrium state, medium performance @ high repetition rate, multi user facility

Linear Accelerators: beam properties adopted from source → excellent beam properties (photo-injectors), ultimate peak performance @ low repetition rate

P = U * I, e.g.: I = 100 mA, U = 5 GV → P = 500 MW !!!





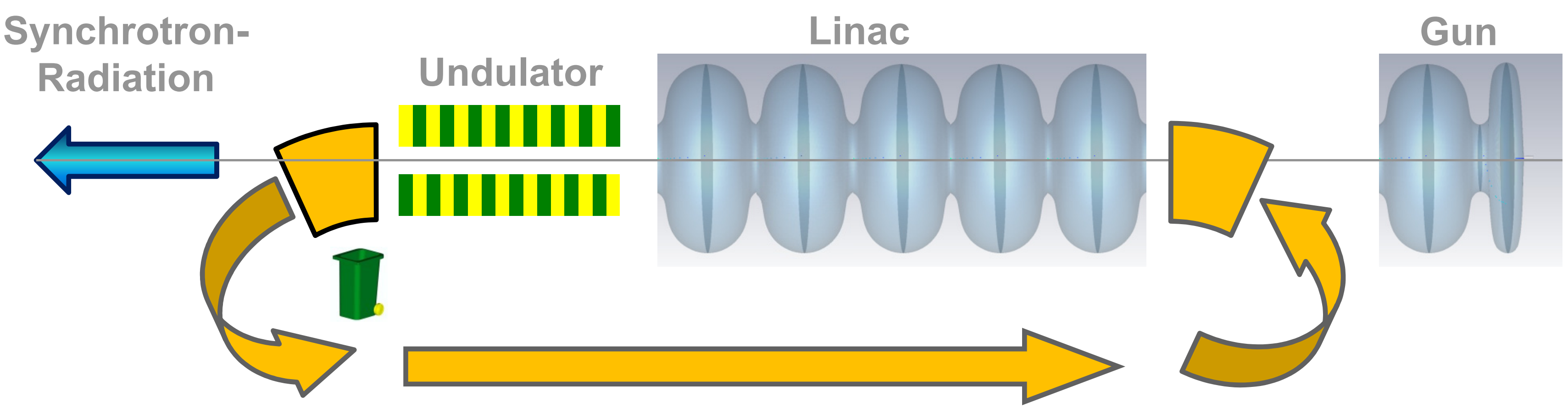
Introduction ERL's

Storage Rings: beam properties = equilibrium state, medium performance @ high repetition rate, multi user facility

Linear Accelerators: beam properties adopted from source → excellent beam properties (photo-injectors), ultimate peak performance @ low repetition rate

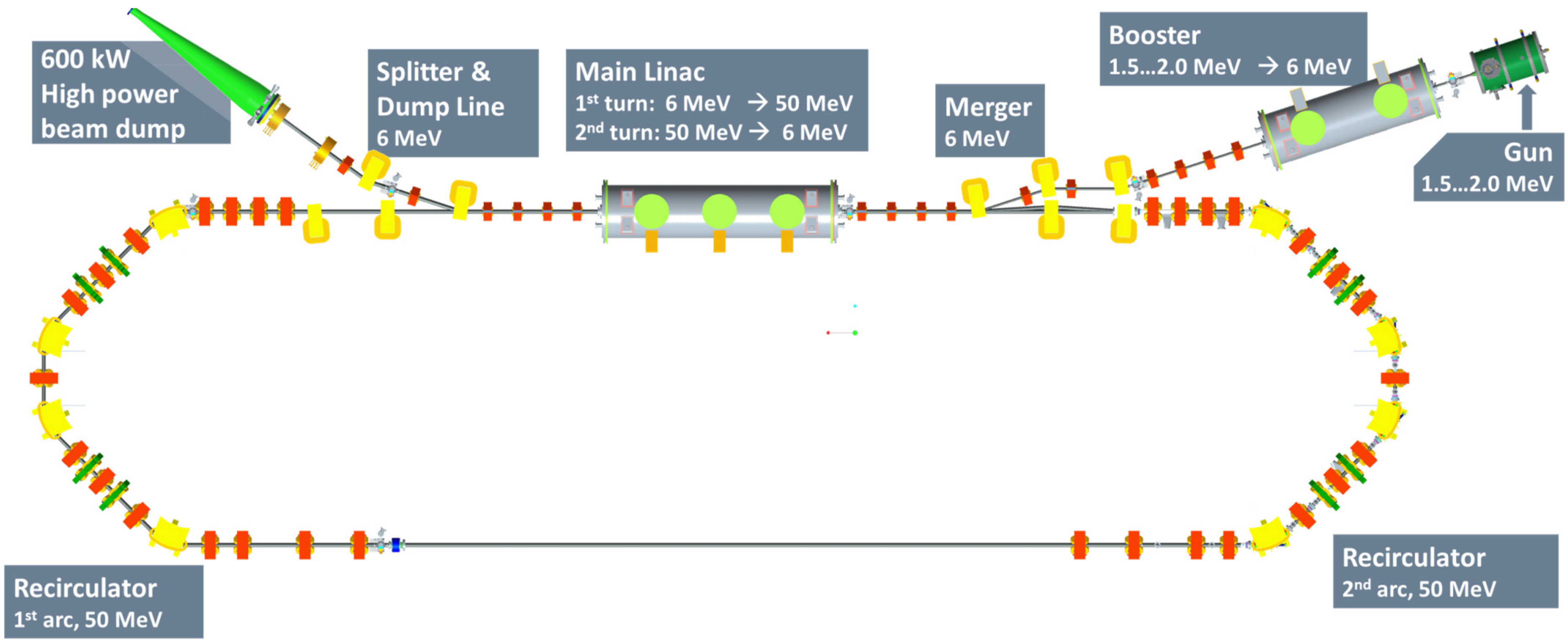
P = U * I, e.g.: I = 100 mA, U = 5 GV → P = 500 MW !!!

ERL as kind of combination: excellent beam properties @ high repetition rates



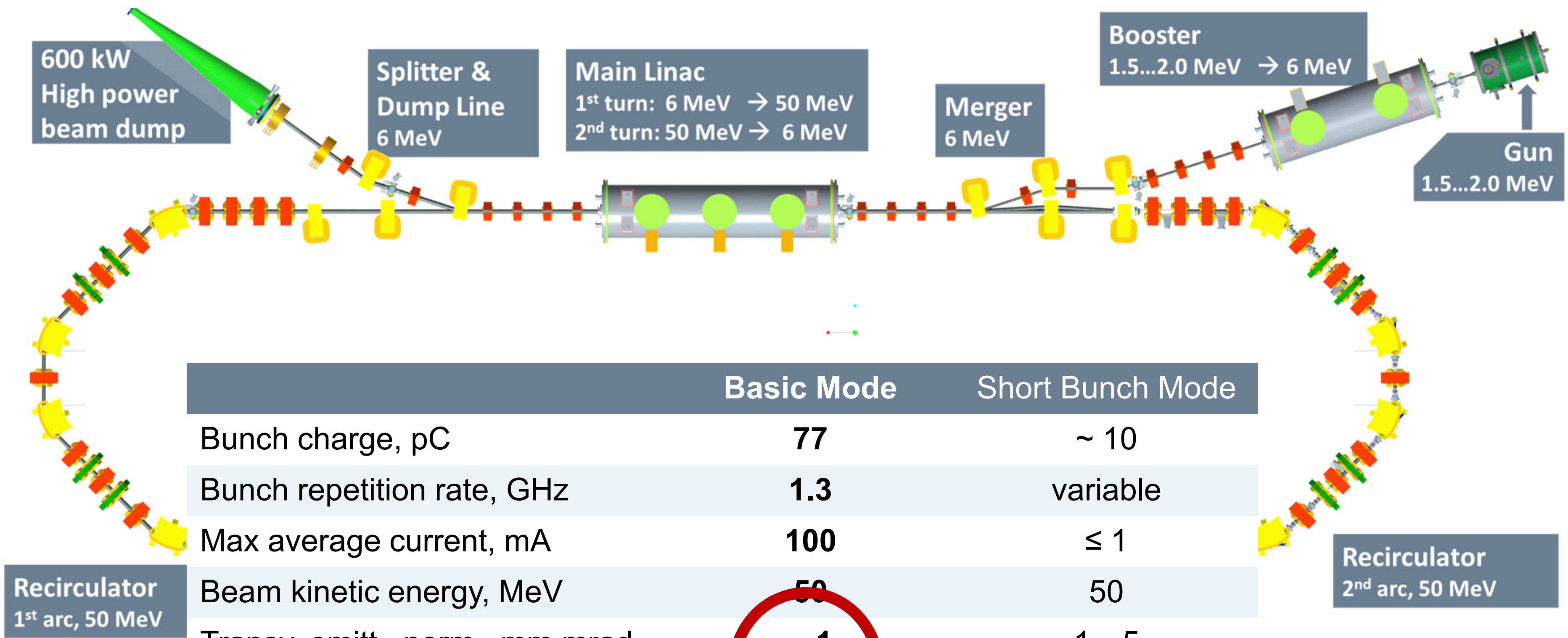
Introduction BERLinPro

Layout as presented in the Conceptional Design Report (CDR):



HZB Helmholtz Zentrum Berlin

Layout as presented in the Conceptional Design Report (CDR):



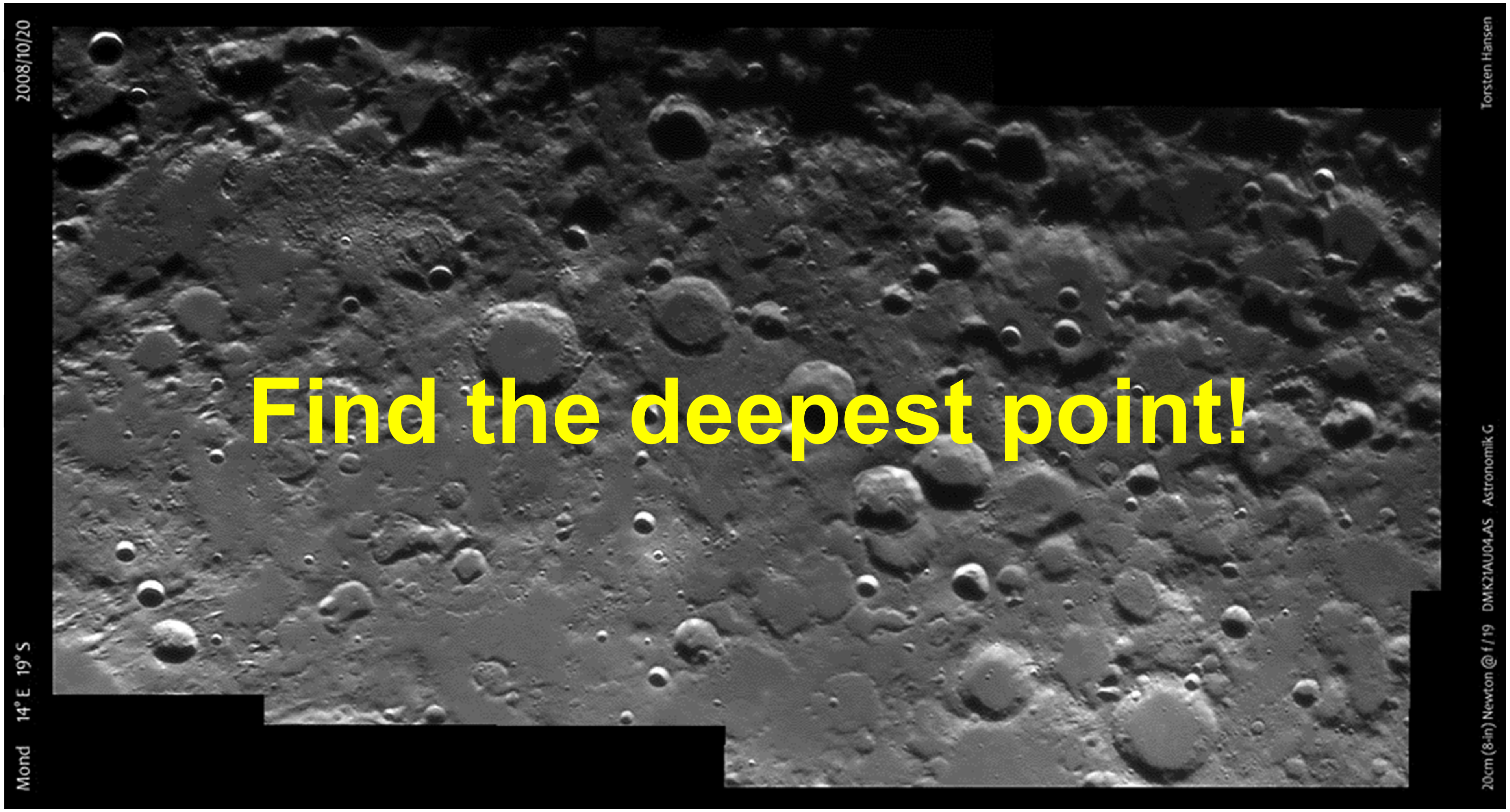
	Basic Mode	Short Bunch Mode
Bunch charge, pC	**77**	~ 10
Bunch repetition rate, GHz	**1.3**	variable
Max average current, mA	**100**	≤ 1
Beam kinetic energy, MeV	**50**	50
Transv. emitt., norm., mm mrad	**~ 1**	1…5
Bunch length, ps, rms	**2.0**	0.1
Relative energy spread, % rms	**~ 0.5**	1…3

Motivation I

HZB Helmholtz Zentrum Berlin

Injector optimization:

multi-objective problem: lots of parameters, constraints & objectives → heuristic methods required



Motivation I

Injector optimization:

multi-objective problem: lots of parameters, constraints & objectives → heuristic methods required

basic beam line setup and optimization: using ASTRA

- 1D parameter scans, "trial and error" parameter tests
- very few automatic routines for partial aspects of the injector (e.g. quad settings for emittance compensation)

builds up deep inside in beam physics and excellent expertise, but…

- very time consuming
- frequent re-iterations are very likely – kind of uninflected work

→ support this work by an additional tool: "brute force", optimizer program

Motivation II

HZB Helmholtz Zentrum Berlin

HZB High-Performance-Computing-Cluster:

- 31 knots with 820 CPU-cores and 1358 GB RAM
- 64-bit OpenSuse Linux, Oracle (Sun) Grid Engine
- Gbit Ethernet connection

Typ: AMD Opteron	Takt [MHz]	L2-Cache je Kern [MB]	Kerne je CPU	Kerne im Cluster	Therm. Verlust-leistung / W	fp_base2000 Benchmark
265 (Master)	1800	2	2	4	95	n/a
865 (alte Knoten)	1800	2	2	64	95	1176
8378 (1 u. 2. Ausbau)	2400	0,5	4	192	75	1924
6168 (3. Ausbau)	1900	0,5	12	96	80	1744
6172 (4. bis 6.Ausbau)	2100	0,5	12	336	80	1961
6276 (7. Ausbau)	2300	1	16	128	80	n/a

Computational Aspects I

→ use the HZB cluster to run the optimizer on:

- run parallel many single ASTRA processes
- no real parallelization required
- Start, control & analysis by a master program
- learned @ ICAP2012: there are in principal similar but more general optimizers under development e.g. from PSI, TRIUMF, …

Optimizer = Master - Program:

- basically a wrapper for ASTRA
- ASTRA: version 3.0 incl. dipole magnets, free format, comb. 2D → 3D calculations
- used code: Fortran (intel compiler)
- system calls use of operating system standard commands

Computational Aspects II

program flux:

collect:
- parameters to be varied,
- start values, limits & initial variation step sizes
- other variation constraints

generate a randomly varied parameter set,
build ASTRA input files from reference samples for each iteration step (incl. start distribution)

loop

run & control the ASTRA batch job
collect the run results: rms values of 6D beam dimensions, emittances, energy & energy spread

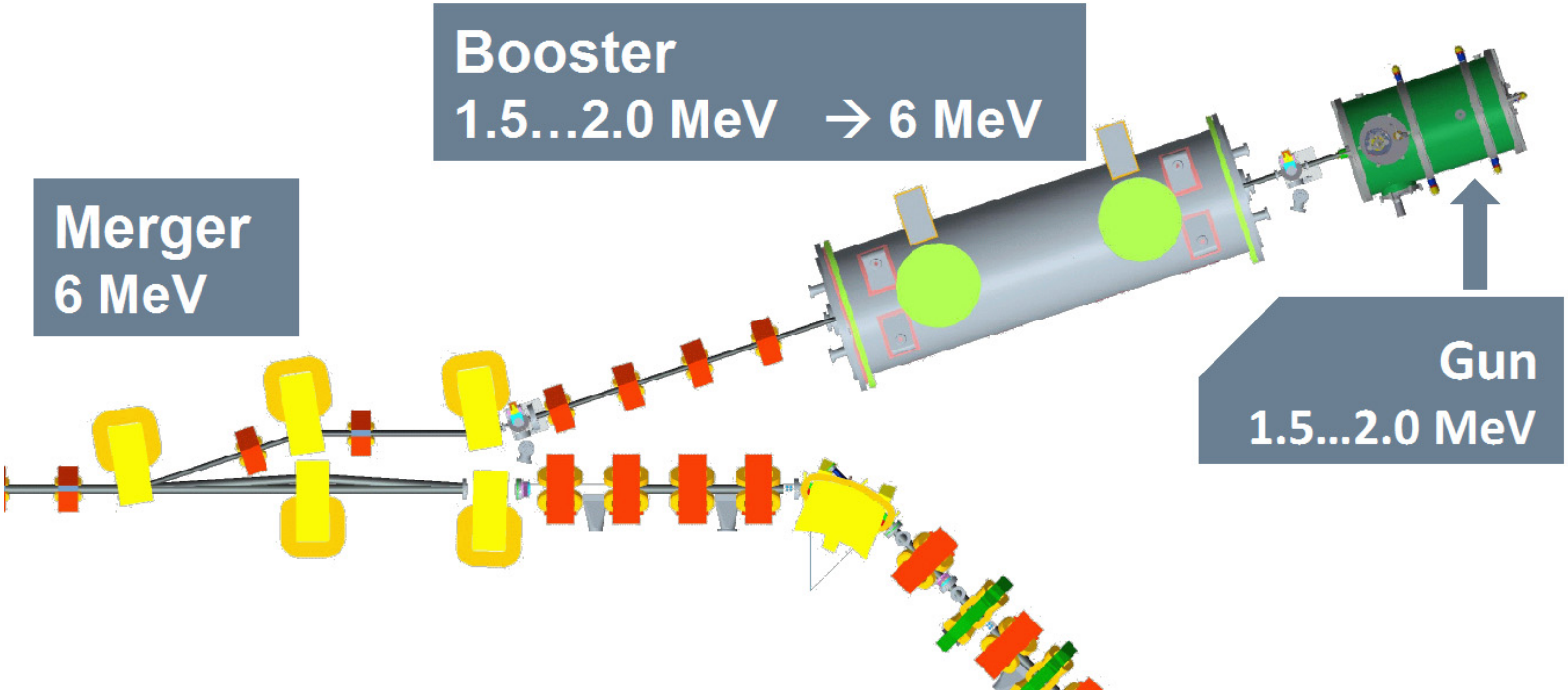
calculate a weighted goal function (to be minimized)

Check for new "best"
→ save & adopt improvements for the further optimization
reduce variation step size

Computational Aspects III

- **convergence:** reached (forced) by reducing the variation step size whenever a certain number of jobs didn't leed to an improvement
- **special care**:
  - avoid overlap of hardware while varying element positions
  - fixed distances of elements
  - ASTRA long runs: control the run time of each job, cancel on limit
  - split beam line for ASTRA at direction changes → more clear geometry description, rotation of phase space & correct job assignment
- **final result:** best solution stored + evolution of hardware & beam parameters
- **post processing:** human beings learning effect, but challenging, tools for visualization of results, search for correlations and most effective knobs
- **Swarm?** program runs many ASTRA jobs in parallel → kind of "swarm" search is performed (random walk, MC), advantages of other algorithms (genetic ?)

Optimization: B*ERL*inPro Injector



B*ERL*inPro injector (CDR case):

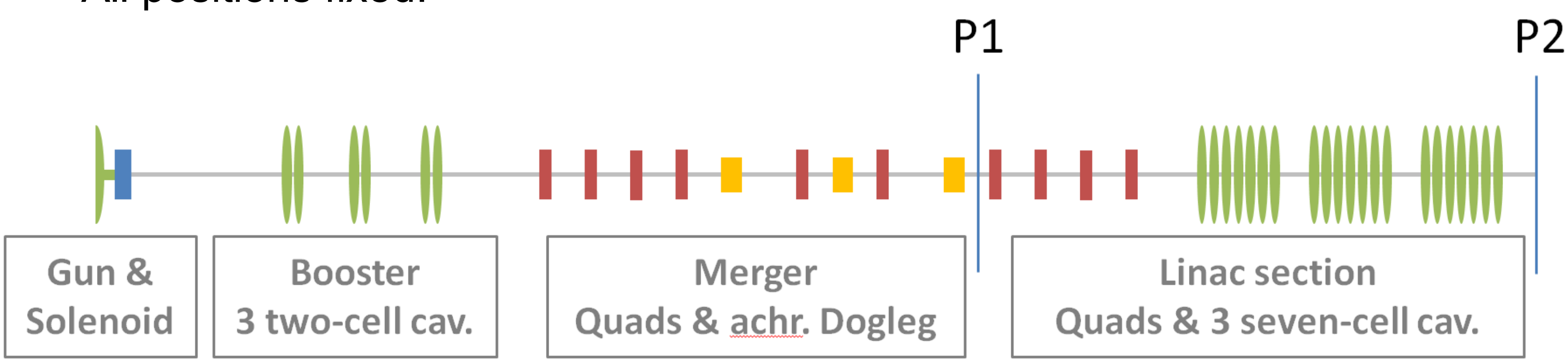
- gun module with a 0.6 cell gun cavity and sc solenoid
- booster module: three 2-cell cavities (one only at “zero crossing”)
- 18 degree dogleg merger section, built from three bends, with quadrupole magnets upstream, inside & downstream the merger

Optimization: B*ERL*inPro Injector

Parameters varied:

- cathode start distribution: laser pulse shape fixed to long. Gaussian & transverse uniform, varied: pulse duration and spot size on cathode
- gun cavity: E_{max} & RF phase
- solenoid: B_{max}
- booster cavities: E_{max} & RF phase (not BC1)
- merger: bend parameters fixed, varied: all quadrupole magnets
- linac section: quadrupole magnets, linac cavity parameters fixed

All positions fixed!



Optimization: B*ERL*inPro Injector

HZB Helmholtz Zentrum Berlin

Goal function:

Main contribution (strongest weighted): projected transverse emittances (100%, no reduction to core, no cut off)

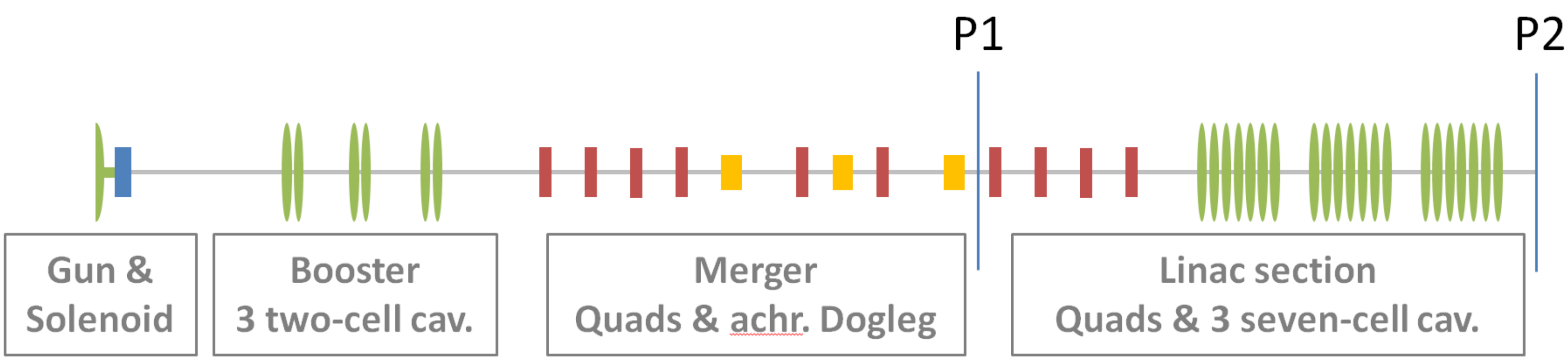
Constraints (medium weighted): energy from the booster ≥ 6 MeV, bunch length out of the merger ≤ 4.5 ps

Program parameter:

ASTRA: 2000 macro particles, 2D → 3D downstream the booster module,

split beam line at the merger end P1, t_{CPU} ~ 15 min

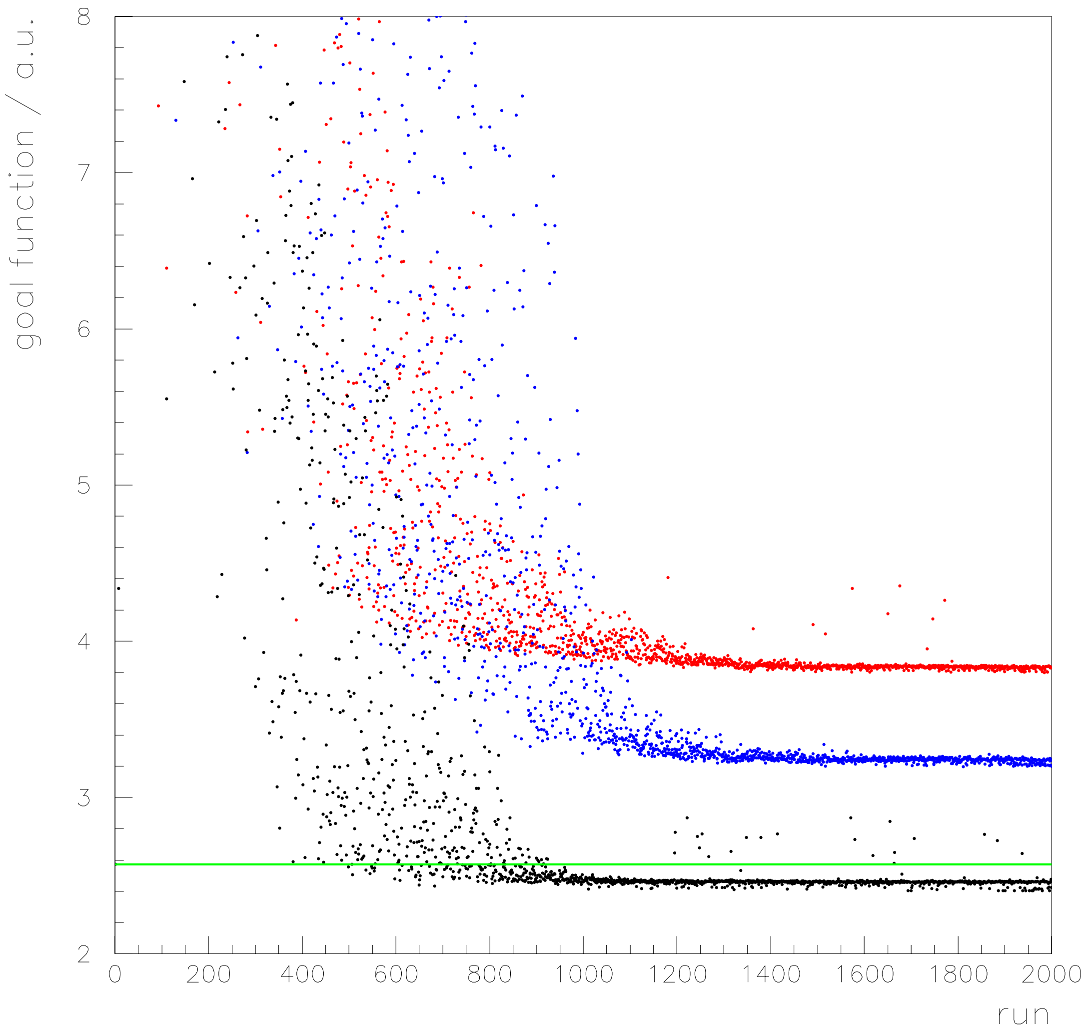
Optimizer: 2000 ASTRA jobs, 50-60 CPU's (max = 64)→ 6-12 hours run time



Optimization: B*ERL*inPro Injector

HZB Helmholtz Zentrum Berlin

SWARM results: evolution of goal function



3 cases with different start values:

- „CDR case"
- medium modifications
- strong modification

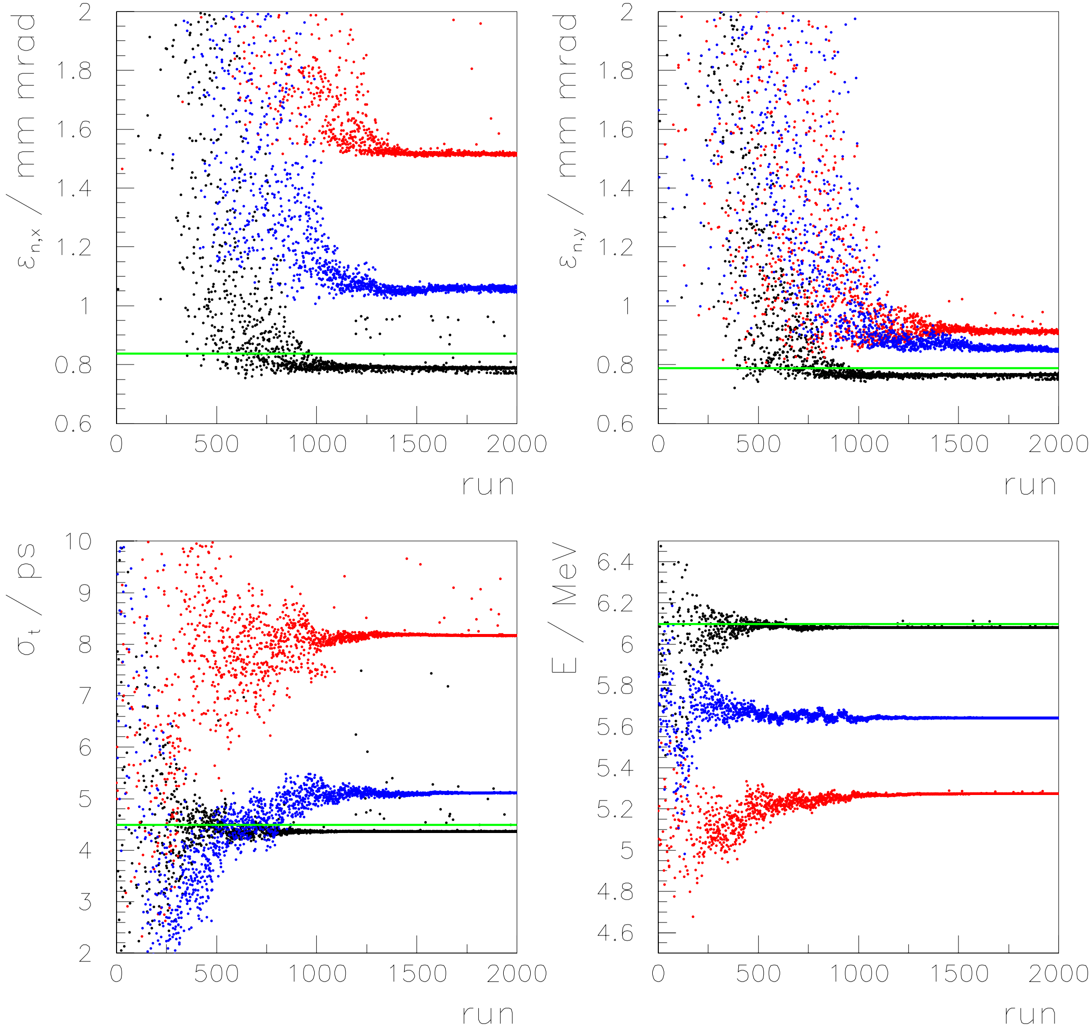
Possible reasons:

- start too far away from the minimum ?
- too few runs !
- too fast freezing of parameter set !
- ???

Optimization: B*ERL*inPro Injector

HZB Helmholtz Zentrum Berlin

SWARM results: evolution of goal function contributors ε_{xy}, σ_t, E_{kin}



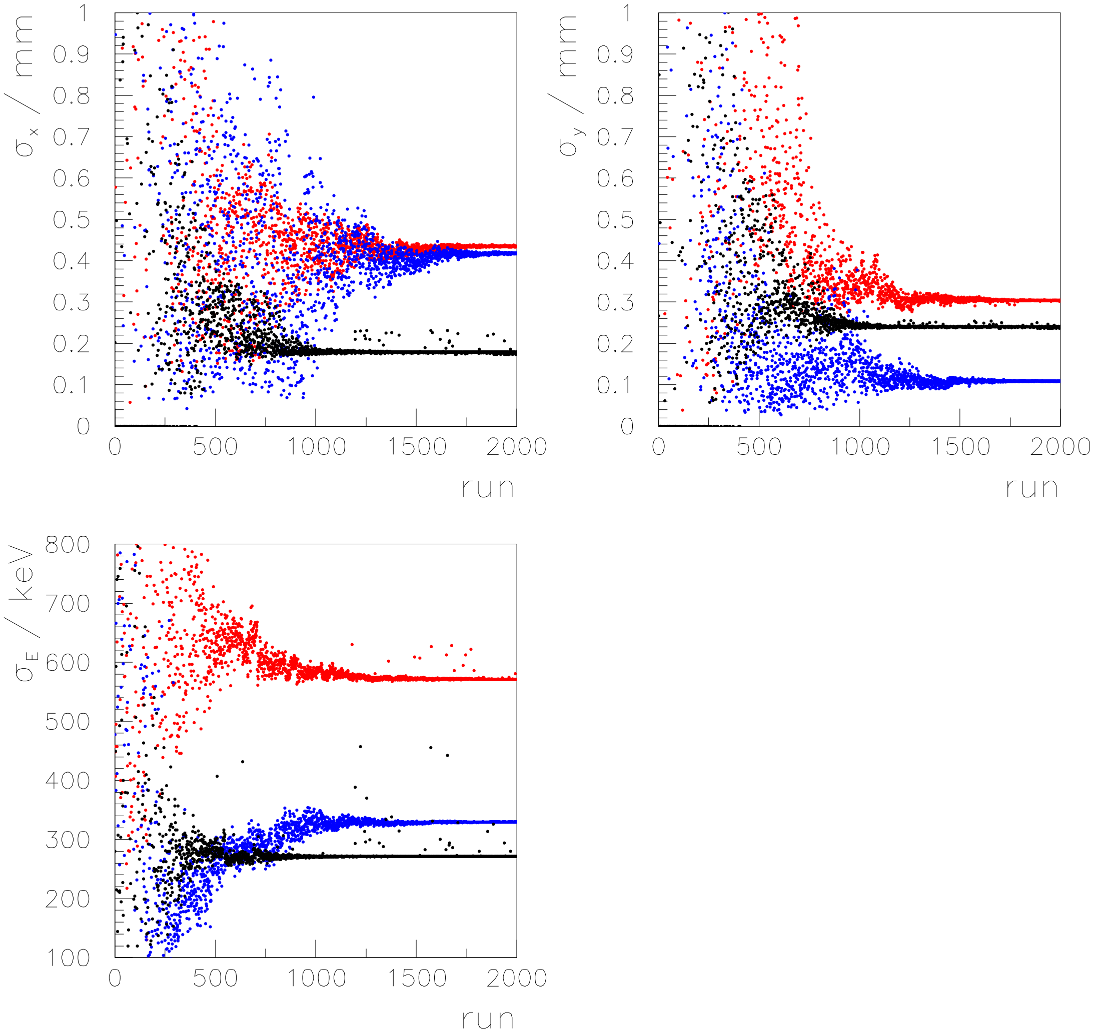
„CDR case“

medium mod.

strong mod.

Optimization: BERLinPro Injector

SWARM results: evolution of transverse beam size and energy spread



„CDR case"

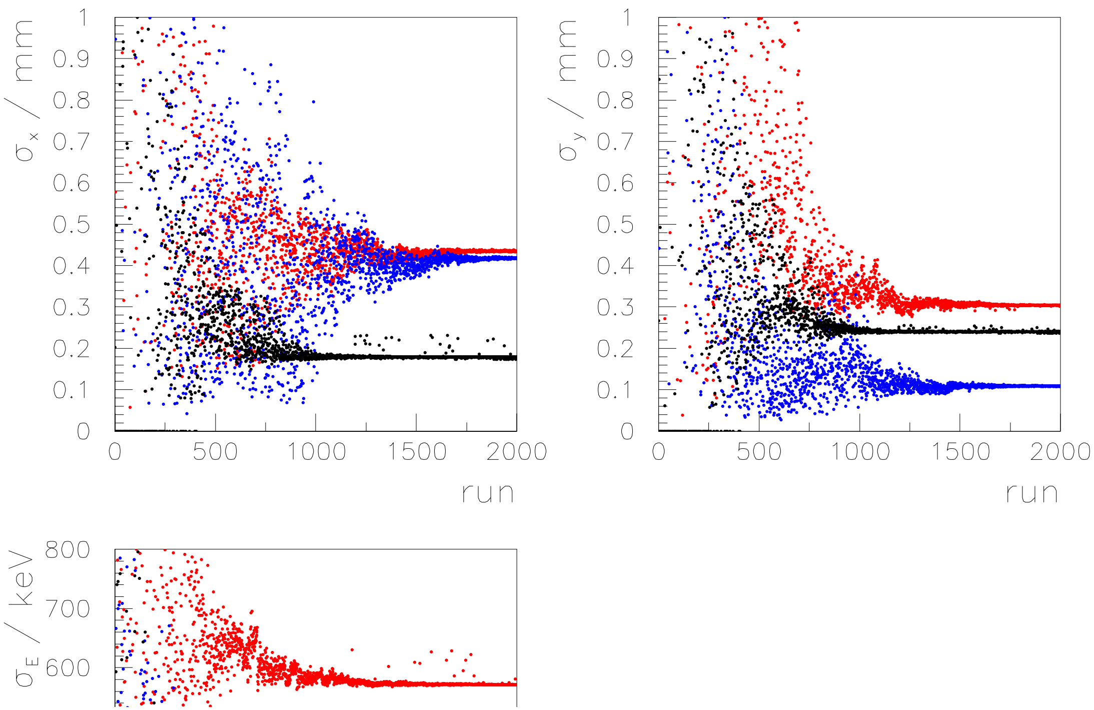
medium mod.

strong mod.

Optimization: B*ERL*inPro Injector

HZB Helmholtz Zentrum Berlin

SWARM results: evolution of transverse beam size and energy spread



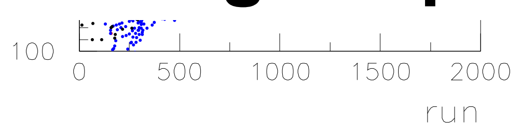
„CDR case“

medium mod.

strong mod.

Can we learn something?

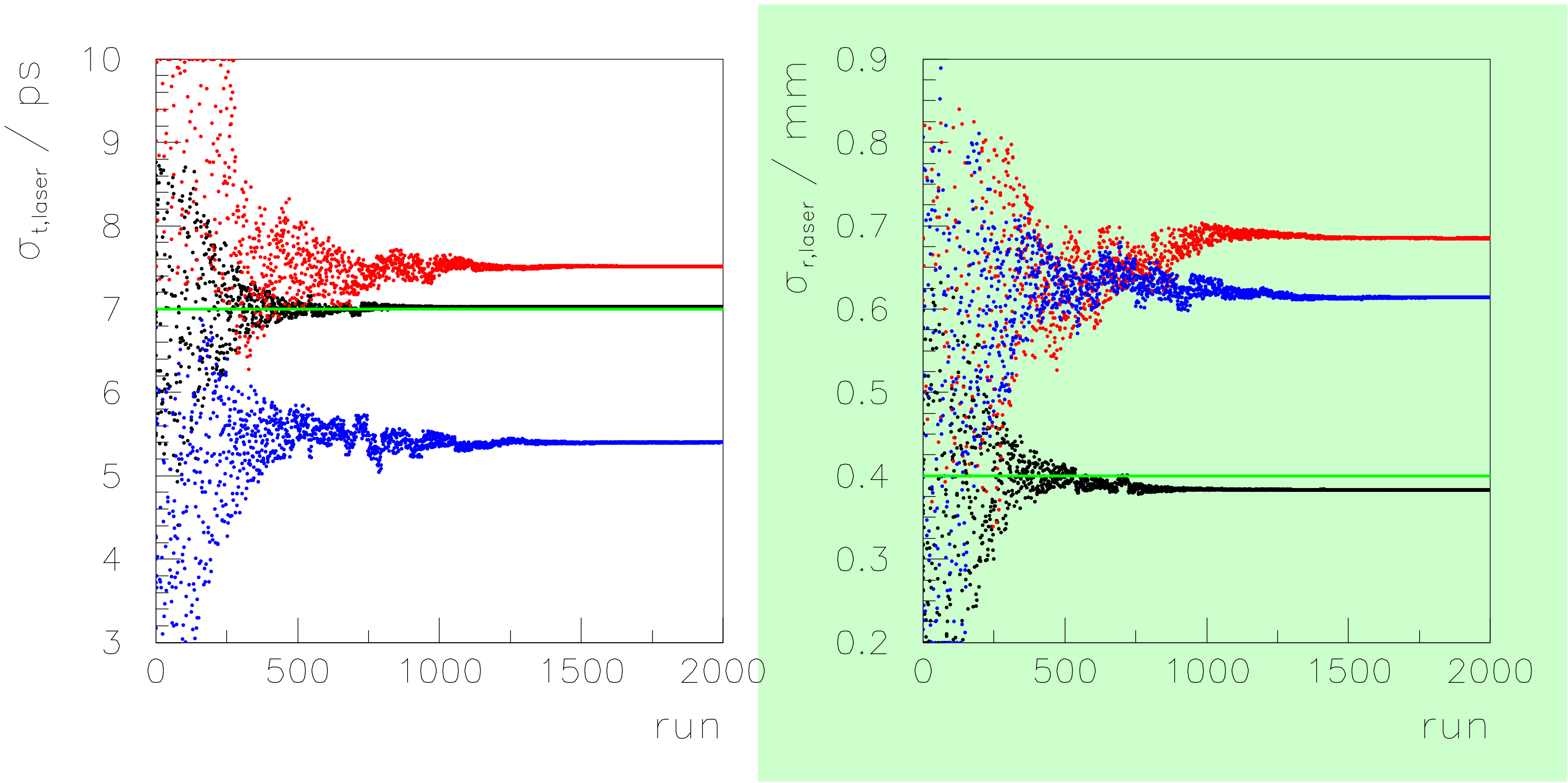
Checking the parameter evolution!



Optimization: B*ERL*inPro Injector

HZB Helmholtz Zentrum Berlin

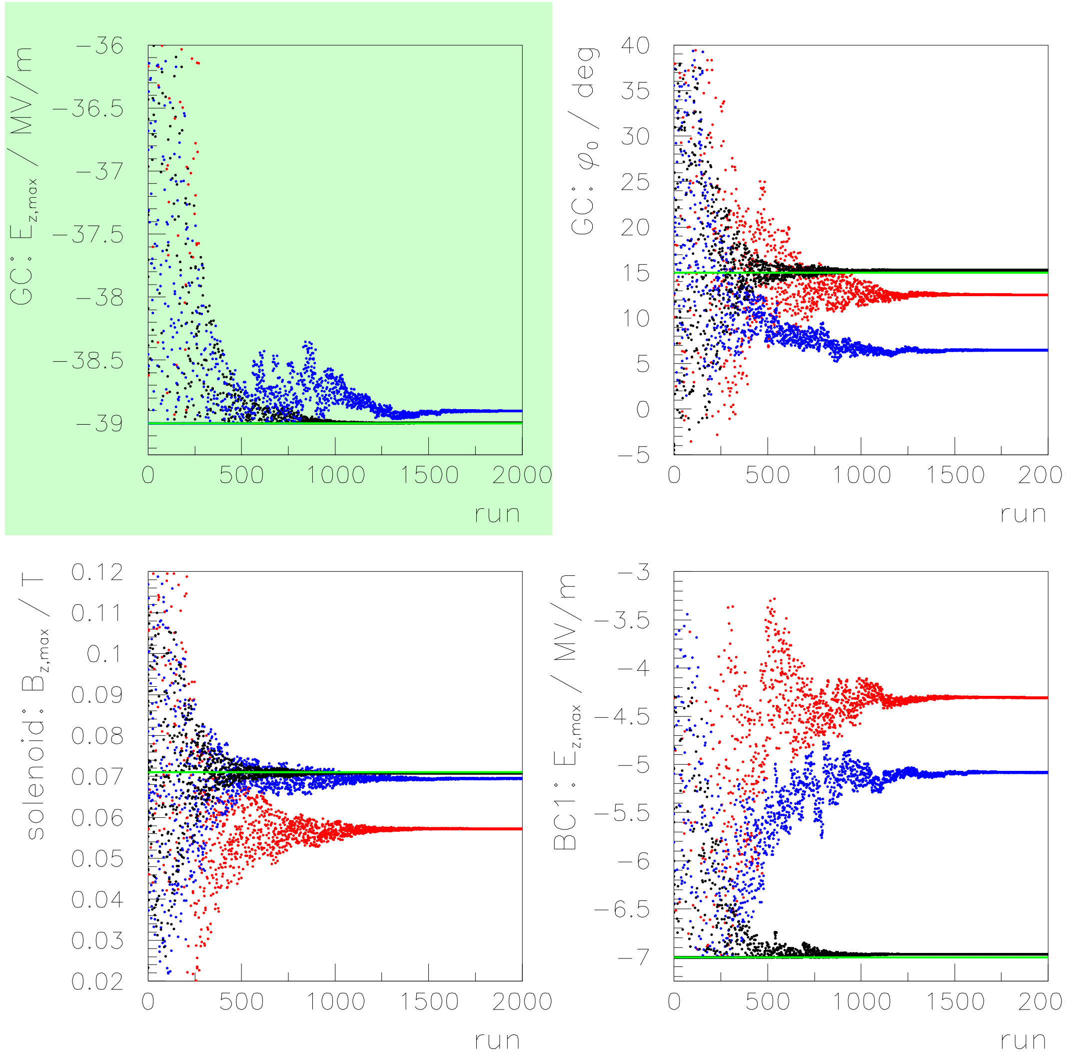
SWARM results: evolution of laser duration & spot size



Optimization: B*ERL*inPro Injector

HZB Helmholtz Zentrum Berlin

SWARM results: evolution of gun cav. field & phase, solenoid & BC1 field



„CDR case“

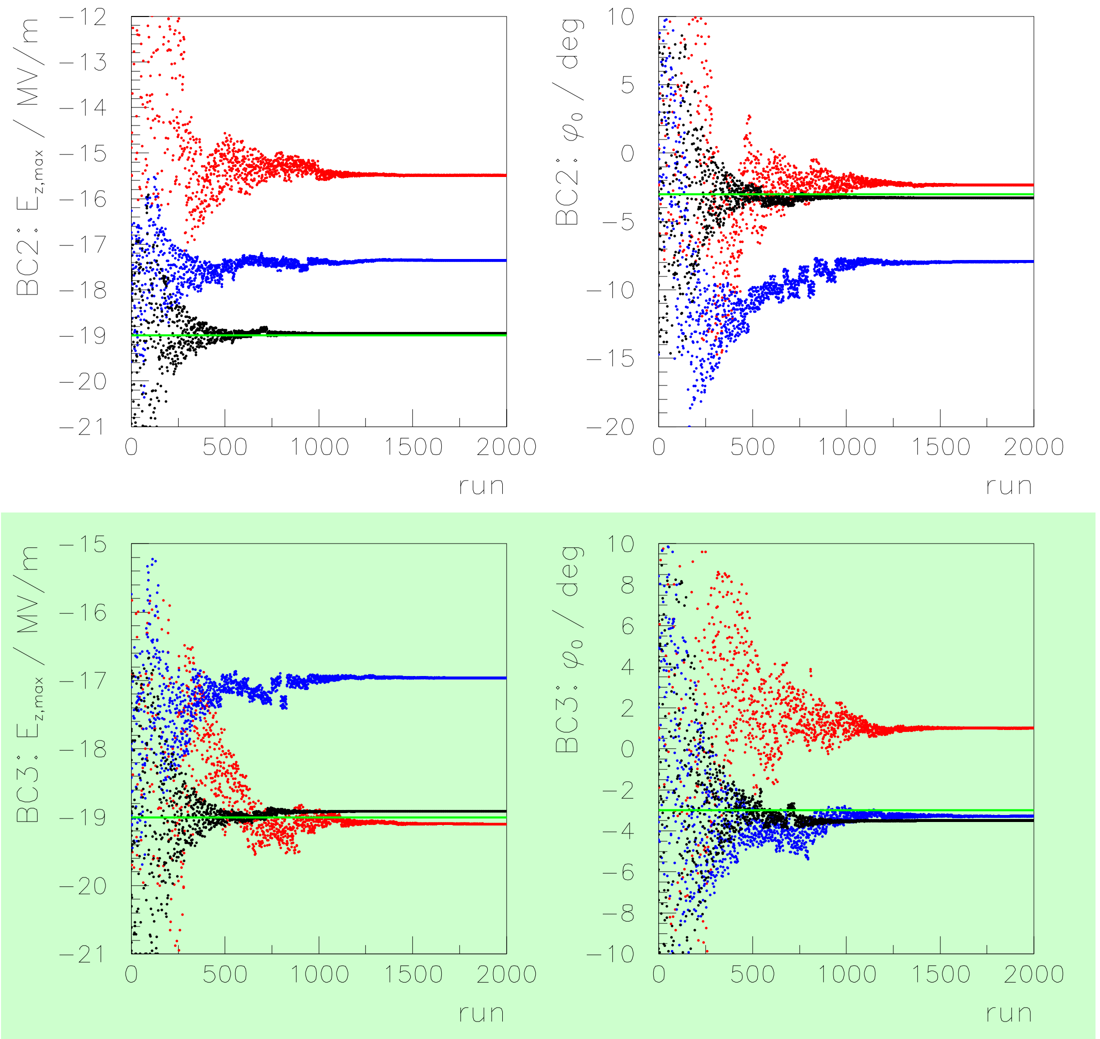
medium mod.

strong mod.

Optimization: B*ERL*inPro Injector

HZB Helmholtz Zentrum Berlin

SWARM results: evolution of booster cavity fields & phases



„CDR case“

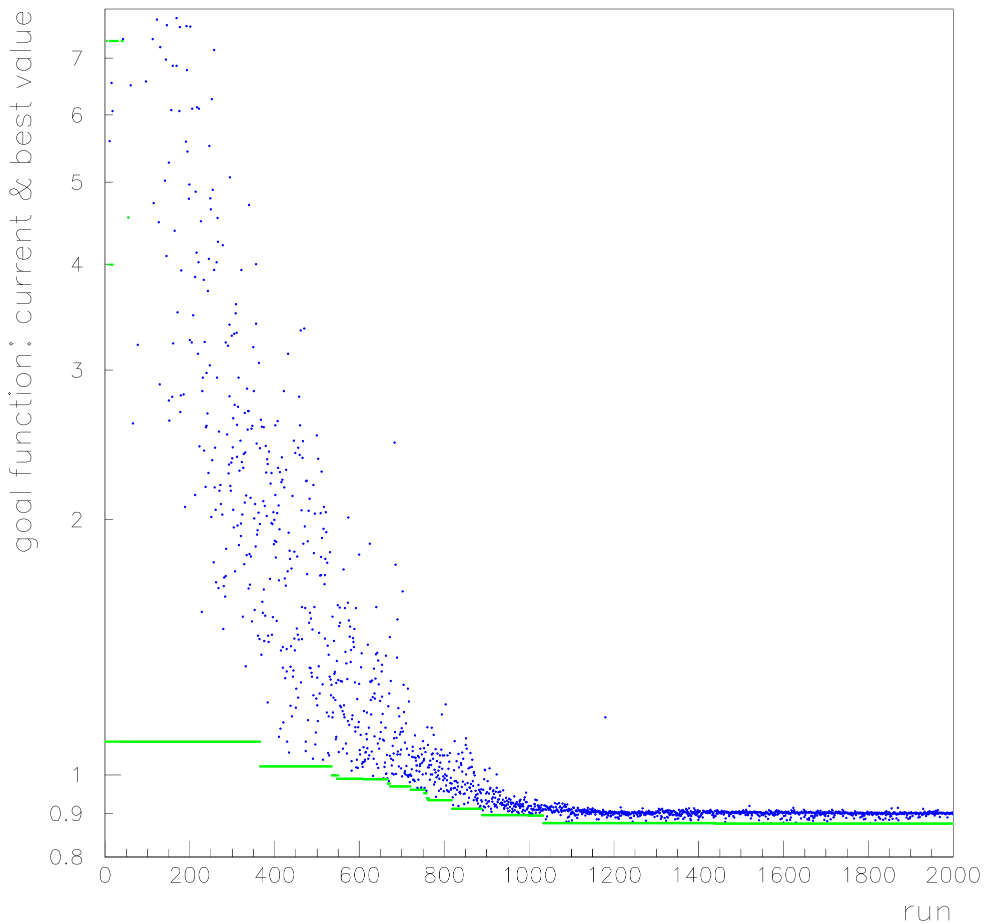
medium mod.

strong mod.

HZB Helmholtz Zentrum Berlin

Optimization problems: uncomplete convergence ?

current and best goal function

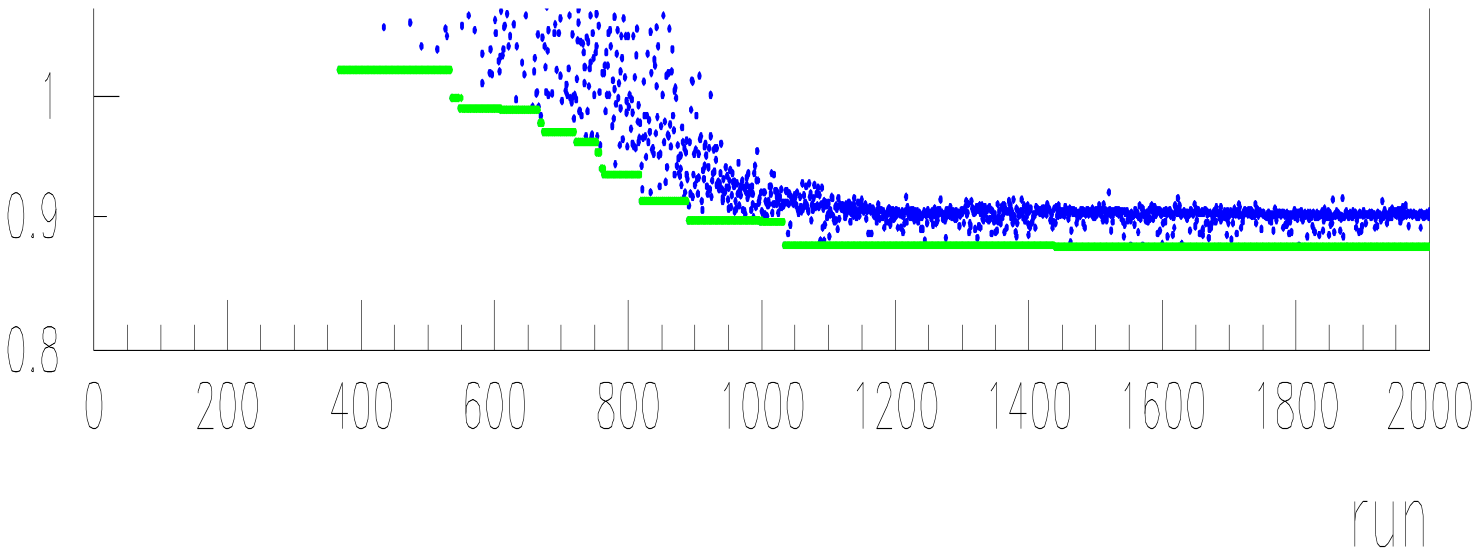


Optimization: B*ERL*inPro Injector

HZB Helmholtz Zentrum Berlin

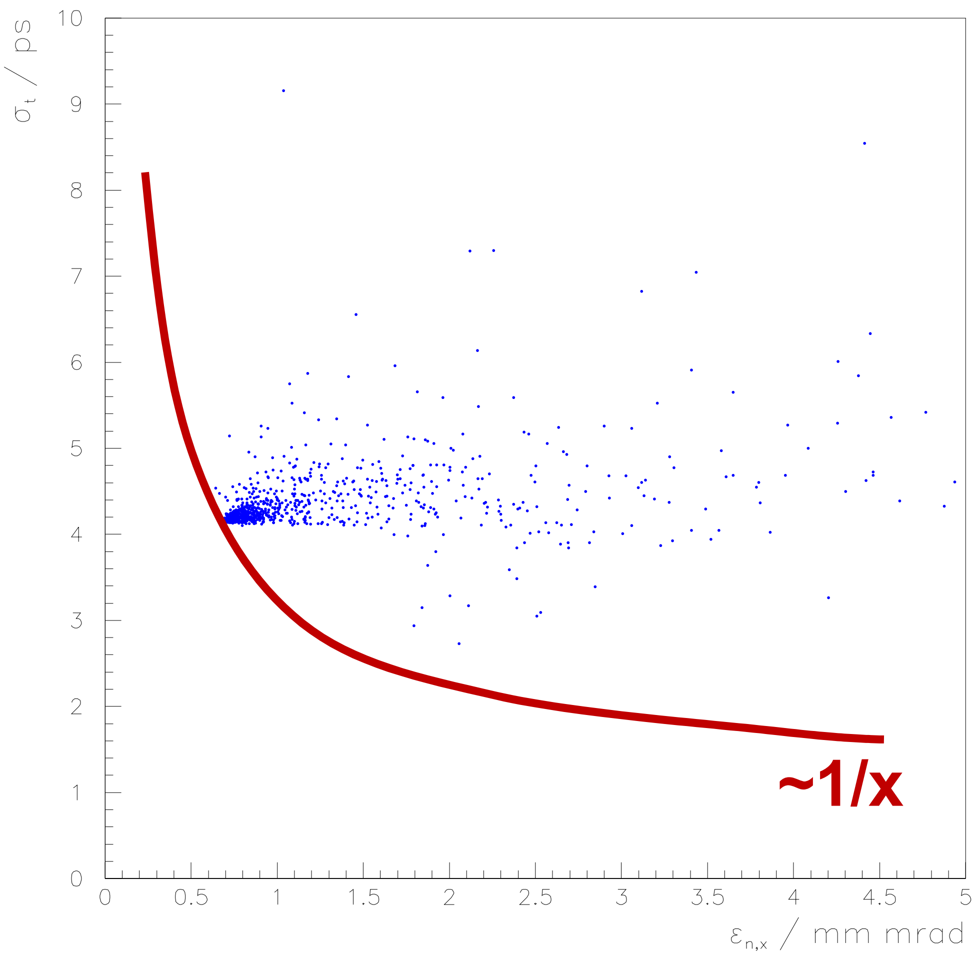
Optimization problems: uncomplete convergence ?

current and best goal function



Optimization problems: Pareto front missing?

bunch length vs emittance



<u>Pareto front expected but missing</u>

- initial goal function with σ_t like a constrain → changed to goal function ε_x*ε_y*σ_t → no change
- too few runs ?
- too fast freezing ?
- ???

Summary & Outlook

HZB Helmholtz Zentrum Berlin

- **ASTRA based swarm optimizer** has been developed
- **first application: BERLinPro injector**
  - parameter set with improved performance compared to CDR case found
  - not generally able to find the CDR solution: very sensitive to start parameters but also to program parameters (e.g. variation range)

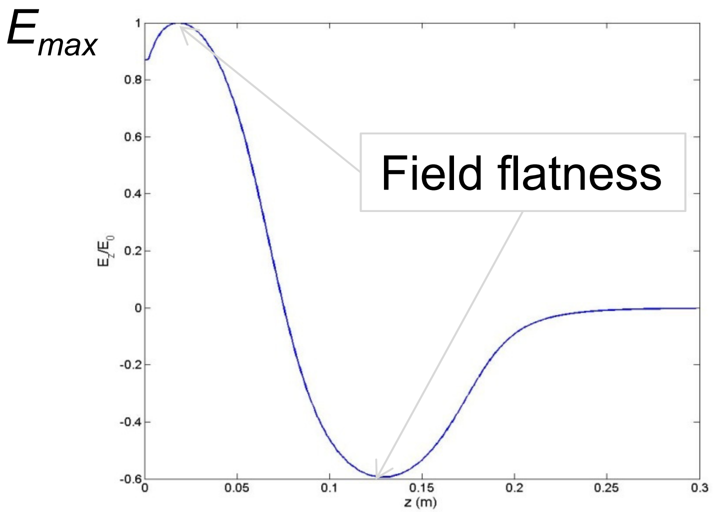
  → more iterations per run: less macro particles, more CPU’s, …

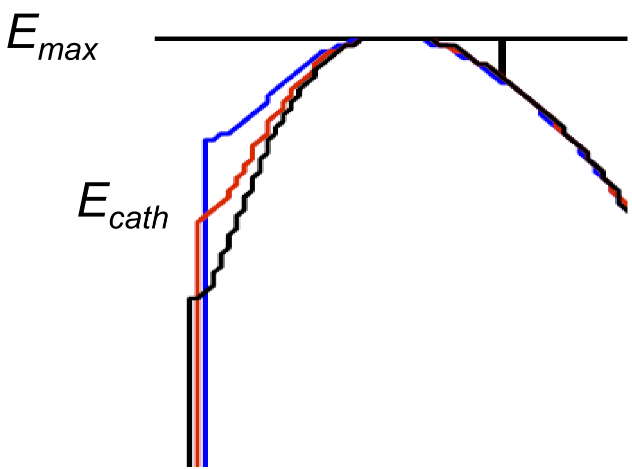
  → more optimizer runs needed for better understanding
- modifications in optimization strategy: genetic algorithms, simulated annealing, ???
- more advanced result analysis tools required: uncover correlations between parameters and performance, evaluate distribution features (shape distortions, beam halo)

Next applications:

• comparison of an 0.4 / 1.4 cell gun cavity based injector for BERLinPro

• Gun cavity field optimization (field flatness, cathode recess,, …) by means of field parameterization or sets of field maps





Thank you!