Research on Compensation of Superconducting Cavity Failures in C-ADS Injector-I

J. P. Dai, Z. Xue, Y. Shao, C. Meng, F. Yan,

Institute of High Energy Physics, CAS

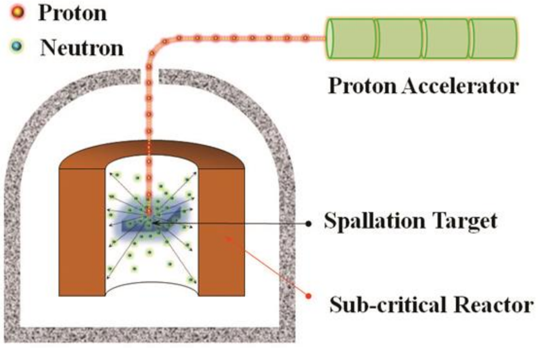
2017/05/18

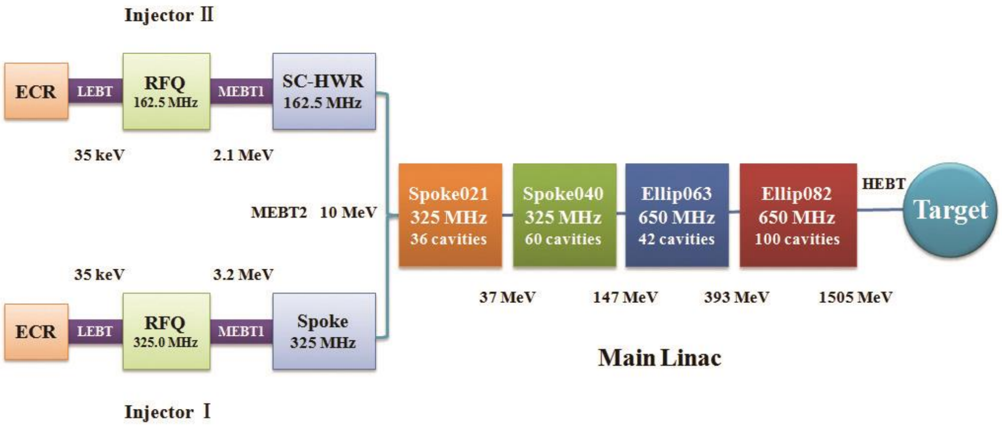
jpdai@ihep.ac.cn

Outline

1. **Introduction**
2. **C-ADS Injector-I**
3. **On-line Computation Method**
   - **Equivalent Model**
   - **Generic Algorithm**
   - **TRACEWIN Verification**
   - **Computing Time**
4. **Summary**

The C-ADS (China Accelerator-Driven Subcritical System) project is a strategic plan to solve the nuclear waste problem in China.





Principle of C-ADS

Layout of the C-ADS CW proton Linac

- ➢ **High energy**
- ➢ **High power**
- ➢ **High reliability**
  - **Strong design:**
    Strong beam dynamics && all hardware systems design
  - **Redundancy:**
    Duplicated injector area, solid-state amplifiers for RF
  - **Repair ability**
    Fault compensation

Main parameters of C-ADS Linac

Particle	Proton	
Energy	1.5	GeV
Current	10	mA
Beam power	15	MW
RF frequency	(162.5)/325/650	MHz
Duty factor	100	%
Beam Loss	<1	W/m
Beam trips/year[1]	<2500 <2500 <25	1 s<t<10 s 10 s<t<5 m t >5 m

[1]H. A ¨ ıt Abderrahim, et al., "Accelerator and Target Technology for Accelerator Driven Transmutation and Energy Production", ttp://science.energy.gov/_/media/hep /pdf/ files/pdfs/ADS White Paper final.pdf (2010)

Fault Compensation Schemes

- **Global fault compensation** [2] (SNS),[3]
  - Retuning all following elements (typically a few minutes)
  - Lattice update every time
  - Little redundancy (Save cost, Complicated and more time)

[2] J. Galambos, et al. MOP057, Proceedings of LINAC 2006, Knoxville,
[3] Jean-Luc Biarrotte1, Didier Urio, PhysRevSTAB, 11, 072803 (2008)

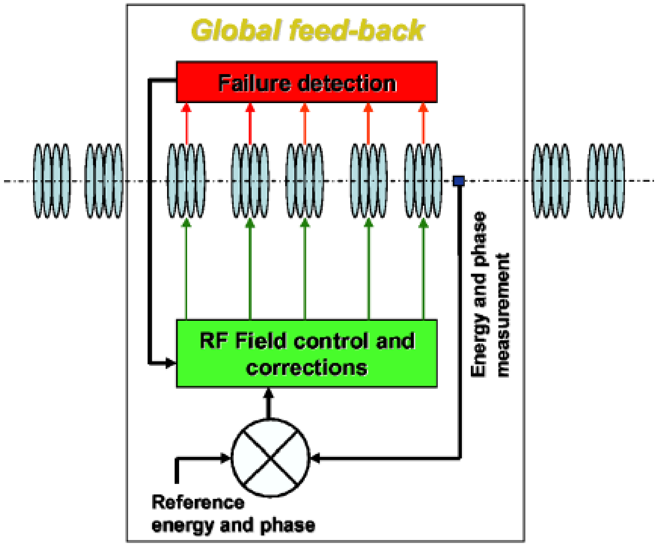


FIG. 1. (Color) Global feedback scheme for rf-fault recovery procedures.

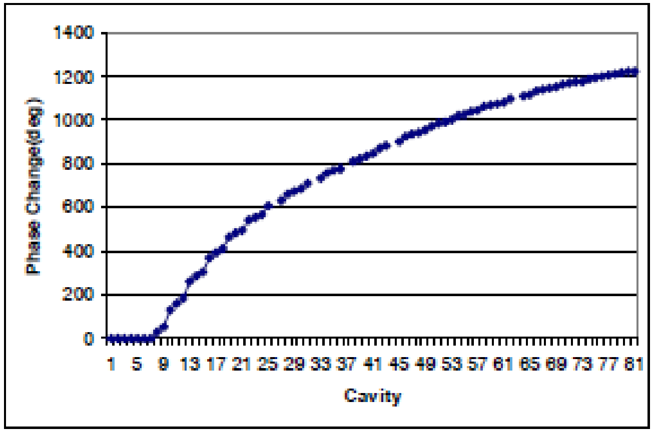


Figure 5: The predicted cavity phase setpoint change resulting from turning off cavity 7. The last cavity phase setpoint was checked with a phase scan and found to be within 1 degree of this prediction.

Fault Compensation Schemes (cont’d)

- **Local fault compensation** [4], [5]
  - Retuning neighbouring elements
  - 30 % accelerating gradient redundancy, ~70% power supply margin
  - Independence and locality (Save Time)

[4] F. Bouly, et al., MOPP103,Proceedings of LINAC2014, Geneva
[5] Biao Sun, et al.,NIMA 785(2015) 77-86

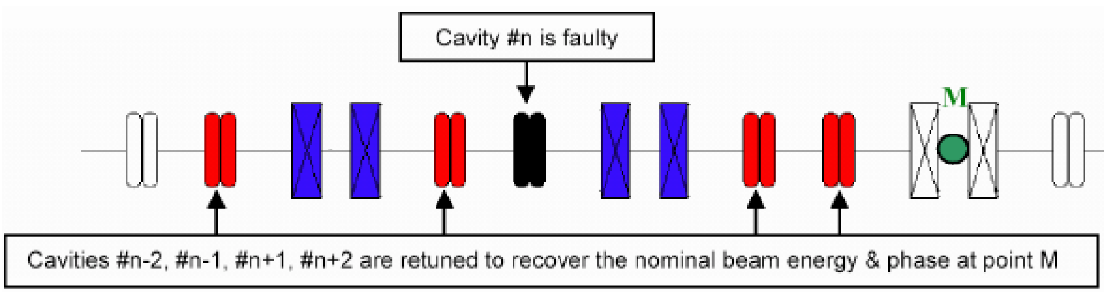


FIG. 2. (Color) Principle of the local compensation method.

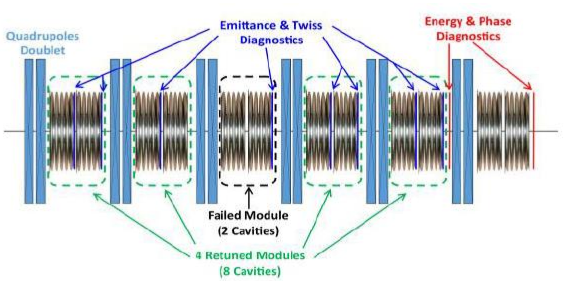


Figure 1: Principle of the local fault compensation method. Example of the retuning strategy used in TraceWin for the failure compensation of one cryomodule in the medium energy section (β_{opt}=0.51).

Methods of Solving Compensation Scenarios

- **Traditional lookup table method**
  - Calculating the beam dynamics by simulation codes (TraceWin), storing all the fault compensation scenarios in a database
  - When a fault is detected, the database is looked up and the right scenario is found
  - The RF cavities are retuned basing on the new scenario

- **On-line computation method***
  - When a fault is detected, the computation starts and a fault compensation scenario is acquired.
  - The RF cavities are retuned basing on the new scenario.
  - Based on Field Programmable Gate Arrays (FPGAs)

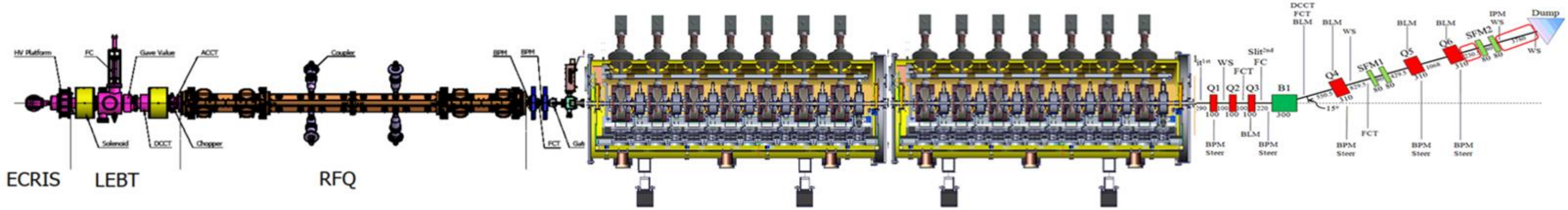
***Proposed and investigated in C-ADS Injector-I**

Outline

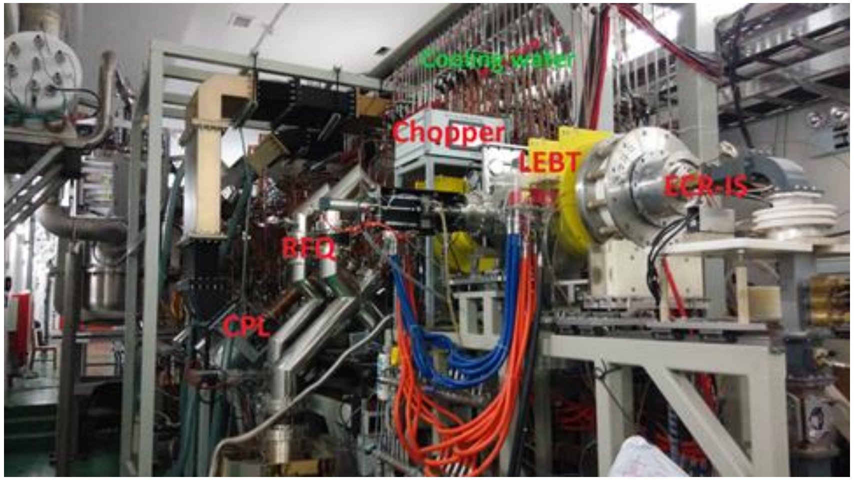
1. Introduction
2. **C-ADS Injector-I**
3. On-line Computation Method
   - Equivalent Model
   - Generic Algorithm
   - TRACEWIN Verification
   - Computing Time
4. Summary

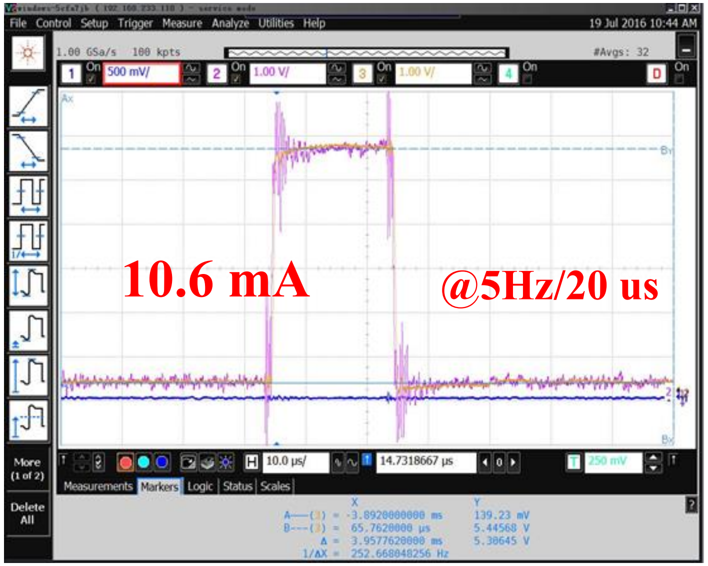
Layout of the C-ADS Injector-I

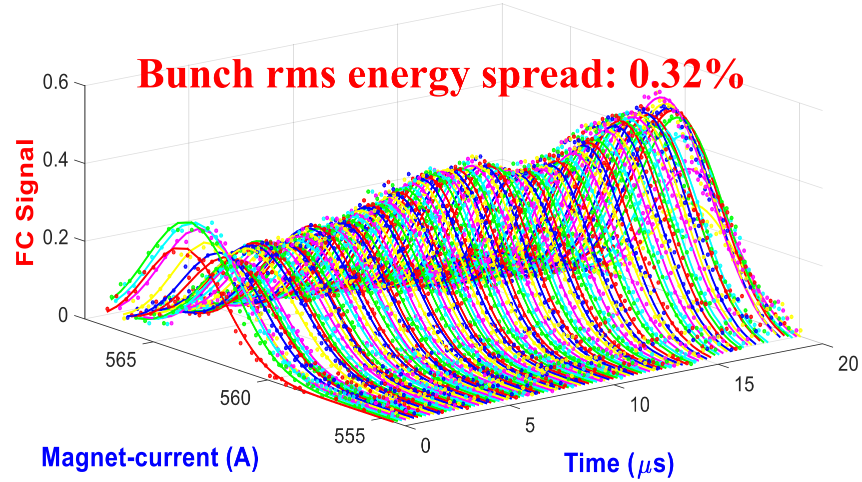
35keV **3.2MeV** **5MeV** **10MeV**

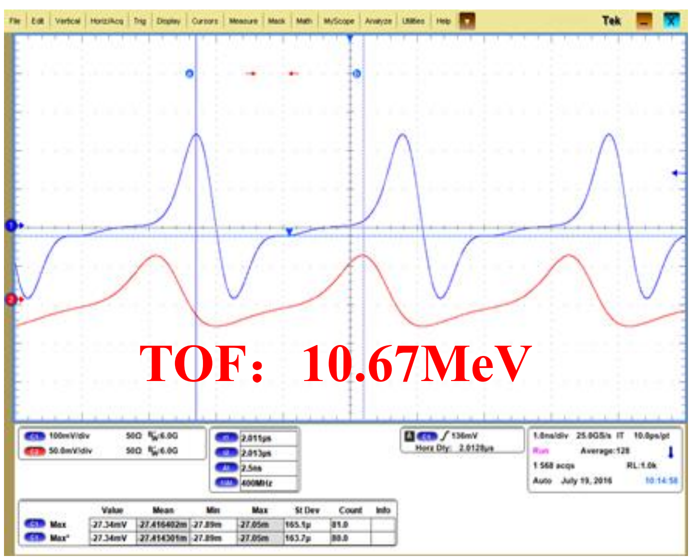


- **Based on the frequency of 325MHz**
- **Composed of an ECR proton source, a LEBT, a 4-vane type copper structure RFQ, a MEBT, a SC section and a beam dump line.**
- **14 Spoke012 cavities and 14 solenoids in the SC section**







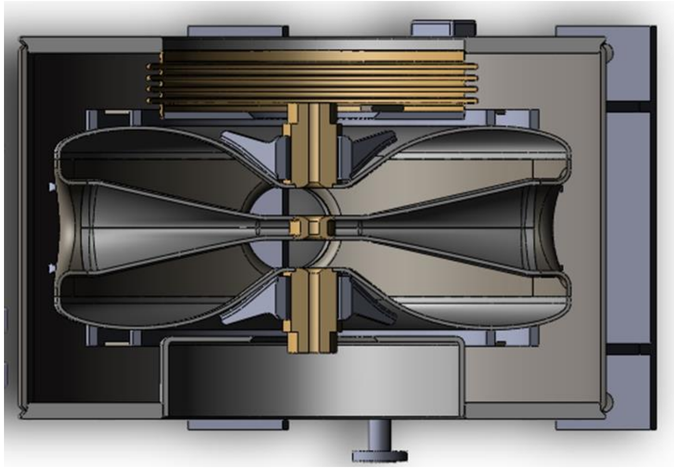


Parameters of C-ADS Injector-I

	Designed value	Achieved value
Energy	10MeV	10MeV
Current	10mA@CW	10.1mA@pulsed 1.6mA@CW
Beam power	100kW	16kW
RF frequency	325	MHz

Parameters of cavities and solenoids of Injector-I

	1#	2#	3#	4#	5#	6#	7#	8#	9#	10#	11#	12#	13#	14#
Cavity Eacc (MV/m)	4.0	4.5	4.9	5.3	5.6	4.7	4.5	3.7	3.3	6.2	6.2	6.3	6.3	6.3
Cavity Phase (°)	-35	-33	-31	-29	-28	-26	-43	-56	-26	-25	-25	-25	-25	-25
Solenoid B (T)	3.3	3.4	3.4	3.4	3.4	2.3	3.4	3.3	2.2	3.1	3.0	3.0	2.9	2.9

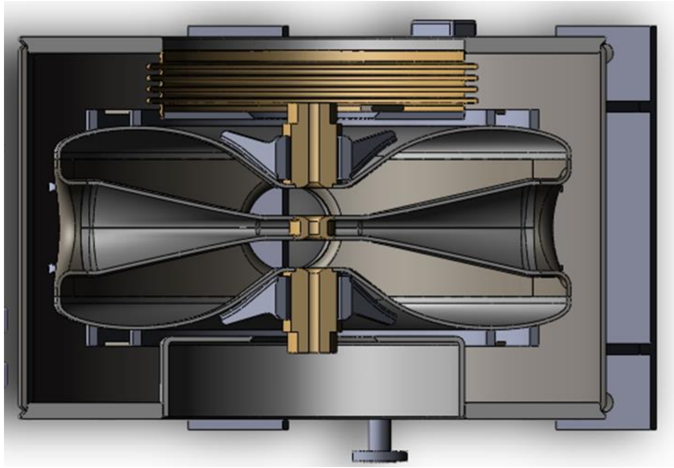


Cross section of Spoke012 cavity

Seven Spoke012 cavities ready for assembly in CM1

Parameters of cavities and solenoids of Injector-I

	1#	2#	3#	4#	5#	6#	7#	8#	9#	10#	11#	12#	13#	14#
Cavity Eacc (MV/m)	4.0	4.5	4.9	5.3	5.6	4.7	4.5	3.7	3.3	6.2	6.2	6.3	6.3	6.3
Achieved Eacc (MV/m)	3.84	5.48	6.53	5.91	6.85	7.55	5.89	5.83	5.83	6.41	6.24	6.95	6.88	4.31
Solenoid B (T)	3.3	3.4	3.4	3.4	3.4	2.3	3.4	3.3	2.2	3.1	3.0	3.0	2.9	2.9

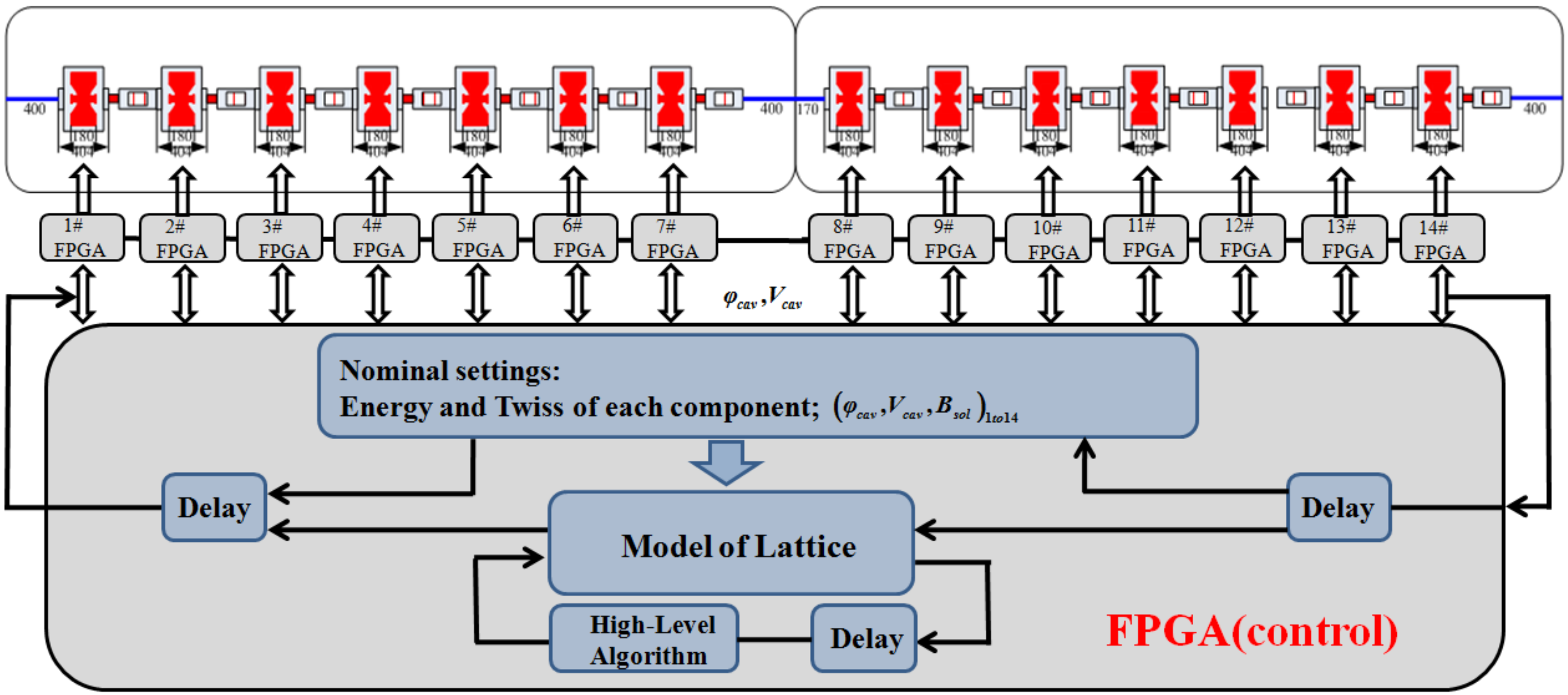


Cross section of Spoke012 cavity

Seven Spoke012 cavities ready for assembly in CM1

Outline

1. Introduction
2. C-ADS Injector-I
3. On-line Computation Method
   - Equivalent Model
   - Generic Algorithm
   - TRACEWIN Verification
   - Computing Time
4. Summary



Schematic diagram of on line simulation and fault compensation at Injector-I

The upper FPGAs are responsible for finding the optimal solutions and calculate the objective parameter according to the nominal setting. When lower FPGAs detects and transports cavity failure signal, the upper FPGAs repeatedly re-calculate the whole process by high-level algorithm until the optimal solutions satisfy the objective function. At last, the modified setting for key elements is transported to lower FPGAs and LLRF by hardware interfaces.

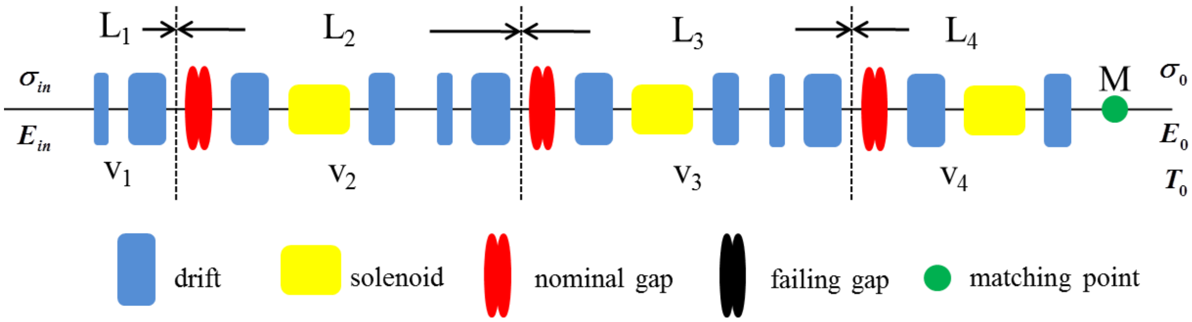
Advantages of the on-line computation method

① **Faster Operating Speed.** As large scale integration consist of logic gates, FPGAs can easily realize parallel computing and synchronous processing during the calculation. This is widely used in control systems and some optimal algorithm.

② **Easier Hardware Interaction.** The technologies of FPGAs are gradually applied to digital LLRF. It is more convenient to use FPGAs instead of PC to do some operations, which can save the time of data transportation through EPICS .

③ **Better Portability and Repeatability.** Due to the periodicity of linac accelerator, a lot of repetitive work of compensation and rematch can be done by FPGAs. At the same time, modularized construction in FPGAs can be easily maintained even the lattice changes.

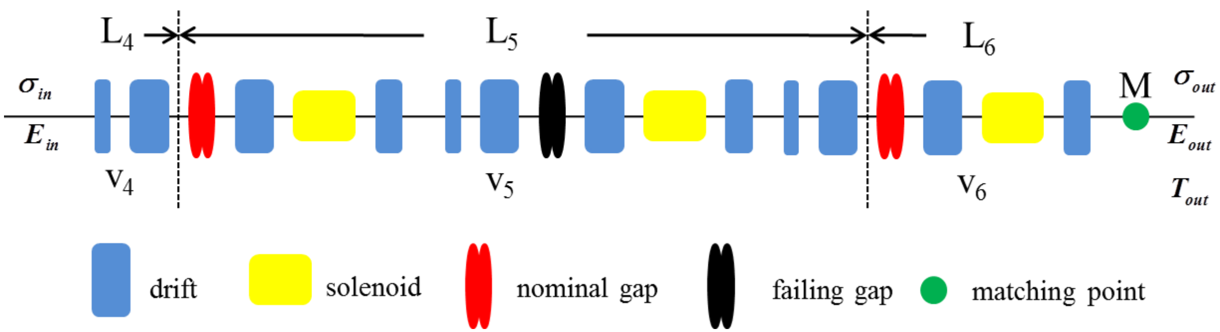
◆ What to be modeled?



$$E_0 = E_{in} + q\sum_{i=1}^{n} V_i cos\varphi_i$$

$$T_0 = \frac{L_1}{v_1} + \frac{L_2}{v_2} + \frac{L_3}{v_3} + \frac{L_4}{v_4}$$

$$\sigma_0 = M_n \ldots M_2 M_1 \sigma_{in} M_1{}^T M_2{}^T \ldots M_n{}^T$$



$$E_{out} = E_{in} + q\sum_{j=1}^{n-1} V_j cos\varphi_j$$

$$T_{out} = \frac{L_4}{v_4} + \frac{L_5}{v_5} + \frac{L_6}{v_6}$$

$$\sigma_{out} = M_{n-1} \ldots M_2 M_1 \sigma_{in} M_1{}^T M_2{}^T \ldots M_{n-1}{}^T$$

Compensation and Rematch ⇨ $E_{out} = E_0$ $T_{out} = T_0$ $\sigma_{out} = \sigma_0$

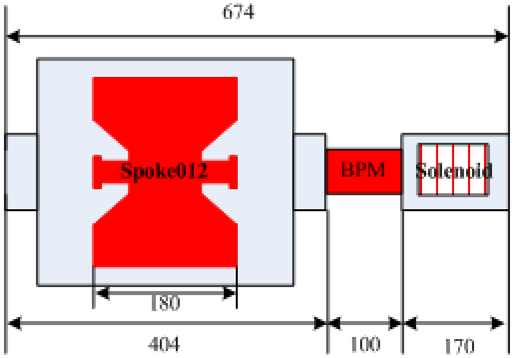
◆ How to model?

✓ **Regression Analysis Prediction Method is used**

✓ **FPGA is consisted of lots of logic gate. The easiest way to get the result of algorithm is to use addition, subtraction, multiplication. So linear basis function models are chosen. (Polynomial model)**

$$y(x,\omega)=\omega_0+\sum_{j=1}^{M-1}\omega_j\varphi_j(x)$$

✓ **With the help of TraceWin, Matlab, Quicktest, the equivalent models of drift/BPM, solenoid, cavity and space charge effect (all components in SC section of Injector-I) are built.**



One period of Injector-I SC section

◆ Modeling of drift and solenoid

Drift

$$R_{xx}=R_{yy}=\begin{pmatrix}1 & \Delta s\\ 0 & 1\end{pmatrix}\quad R_{zz}=\begin{pmatrix}1 & \dfrac{\Delta s}{\gamma^2}\\ 0 & 1\end{pmatrix}$$

⇨ $$f=a+b*E_n$$

✓ **Cosine and sine may be directly calculated by CORDIC of FPGAs**

Solenoid

$$R_{xx}=R_{yy}=\begin{pmatrix}\cos^2(k\Delta s) & \dfrac{\sin(k\Delta s)\cos(k\Delta s)}{k}\\ -k\sin(k\Delta s)\cos(k\Delta s) & \cos^2(k\Delta s)\end{pmatrix}$$

$$R_{xy}=-R_{yx}=\begin{pmatrix}\sin(k\Delta s)\cos(k\Delta s) & \dfrac{\sin^2(k\Delta s)}{k}\\ -k\sin^2(k\Delta s) & \sin(k\Delta s)\cos(k\Delta s)\end{pmatrix}$$

⇨

$$k_{sol}=\sum_{i=0}^{1}\sum_{j=0}^{3}c_{ij}B^iE_{in}{}^j$$

$$\frac{1}{k_{sol}}=\sum_{i=0}^{6}\sum_{j=0}^{2}d_{ij}B^iE_{in}{}^j$$

◆ Modeling of cavity

✓ **The structure of "drift + gap + drift" is chosen to be equivalent to the SC cavity.**

Energy

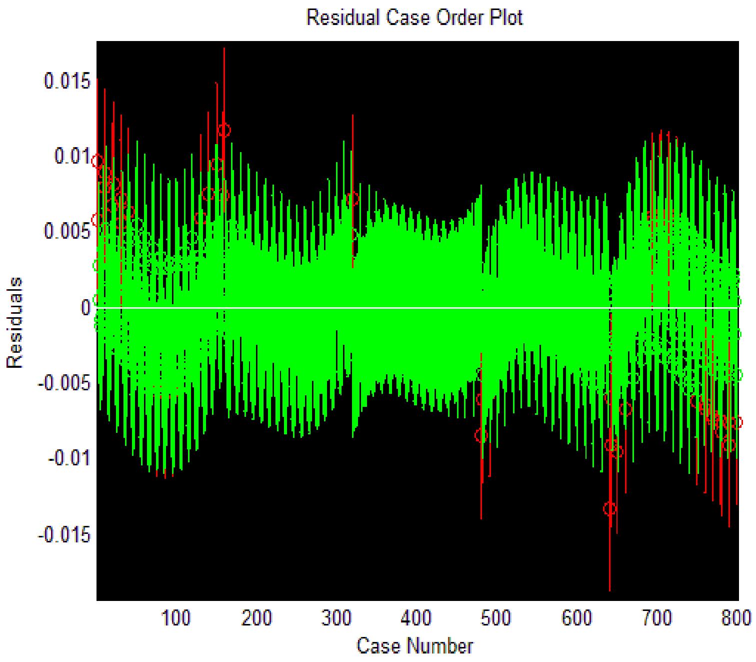
$$E_{out} = E_{in} + E_0 TL\cos(\varphi_s) \quad \Rightarrow \quad E_{out} = \sum_{i=0}^{2}\sum_{j=0}^{2}\sum_{k=0}^{2} a_{ijk} E_{in}{}^{i} \varphi_s{}^{j} Eacc^{k}$$

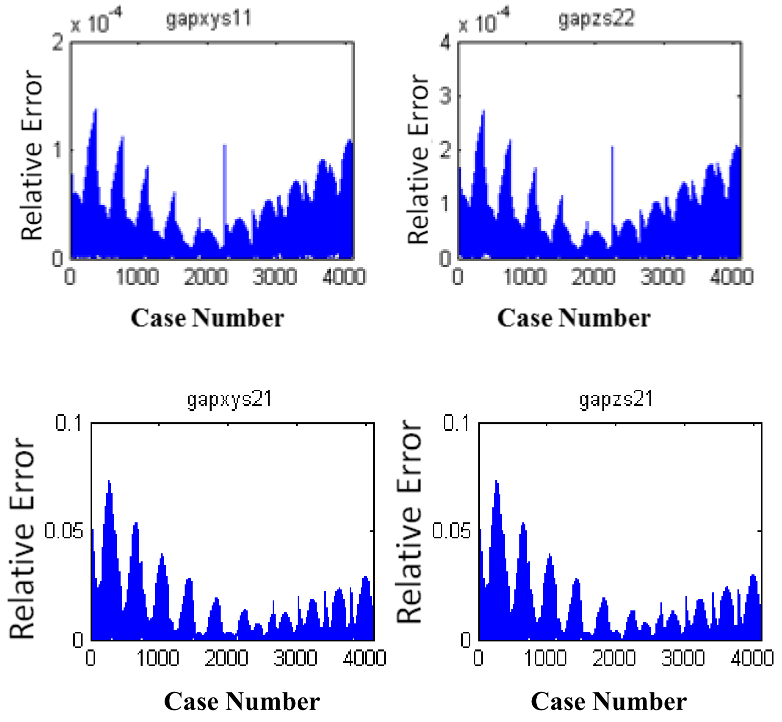
Matrix of gap

$$R_{xx} = R_{yy} = \begin{pmatrix} \sqrt{\frac{(\beta\gamma)_i}{(\beta\gamma)_o}} & 0 \\ \frac{k_{xy}}{(\beta\gamma)_o} & \sqrt{\frac{(\beta\gamma)_i}{(\beta\gamma)_o}} \end{pmatrix} \qquad k_{xy} = -\frac{q\pi E_0 TL_s \sin(\varphi_s)}{mc^2 \overline{\beta}^2 \overline{\gamma}^2 \lambda}$$

$$\Rightarrow \quad Matrix\_element\_gap = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} b_{ijk} E_{in}{}^{i} \varphi_s{}^{j} Eacc^{k}$$

$$R_{zz} = \begin{pmatrix} 1 & 0 \\ \frac{k_z}{(\beta\gamma)_o} & \frac{(\beta\gamma)_i}{(\beta\gamma)_o} \end{pmatrix} \qquad k_z = \frac{2q\pi E_0 TL_s \sin(\varphi_s)}{mc^2 \overline{\beta}^2 \lambda}$$





Model errors of energy and transfer matrix for “gap”

◆ Modeling of space charge effect

- ✓ Considering the linear space charge, the components could be divided into short slices, and space charge could be dealt with as a thin lens inserting into each slice.
- ✓ The space-charge transfer matrix applies on a slice of Δs.
- ✓ Since there are so many variables in the equation, ellipsoid form factor, energy and beam size are separated and modeled respectively.

$$R_{ce} = \left[\begin{array}{cc:cc:cc} 1 & 0 & 0 & 0 & 0 & 0 \\ F_x \Delta s & 1 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & F_y \Delta s & 1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \gamma F_z \Delta s & 1 \end{array}\right]$$

$$F_x = \frac{3qI\lambda(1-f)}{20\sqrt{5}\pi\varepsilon_0 mc^3\gamma^3\beta^2(a_x+a_y)a_z a_x}$$

$$F_y = \frac{3qI\lambda(1-f)}{20\sqrt{5}\pi\varepsilon_0 mc^3\gamma^3\beta^2(a_x+a_y)a_z a_y}$$

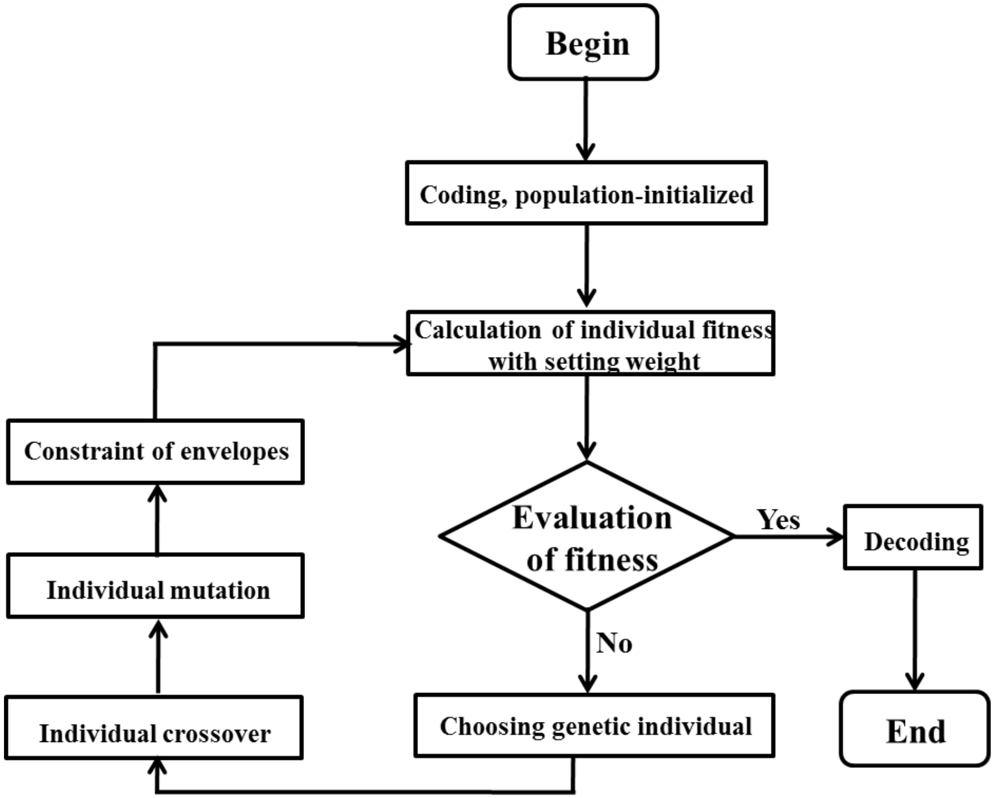
$$F_z = \frac{3qI\lambda(1-f)}{20\sqrt{5}\pi\varepsilon_0 mc^3\gamma^2\beta^2 a_x a_y a_z}$$

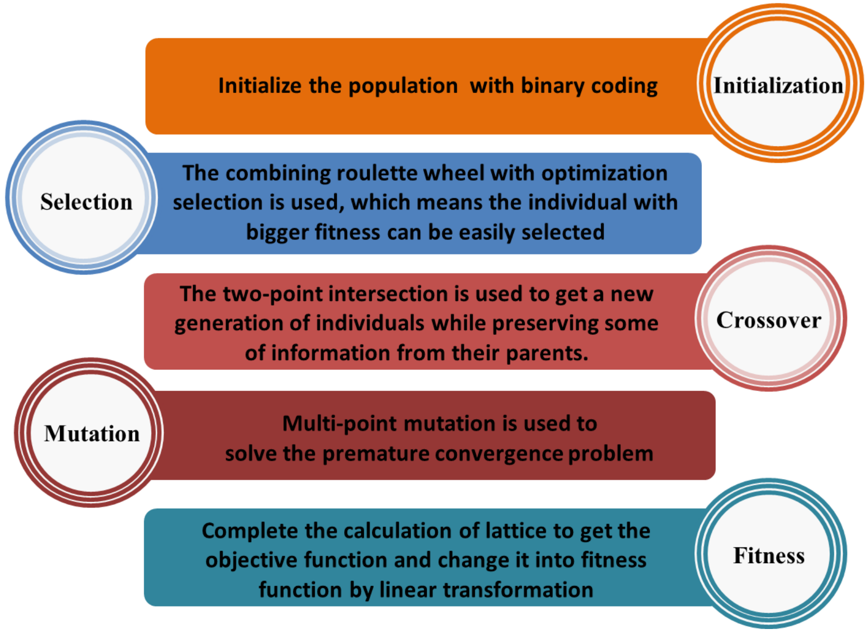
◆ Generic algorithm

- ✓ **Genetic algorithms is a highly parallel, random and adaptive searching algorithm developed by means of natural selection and evolution of biological circles.**

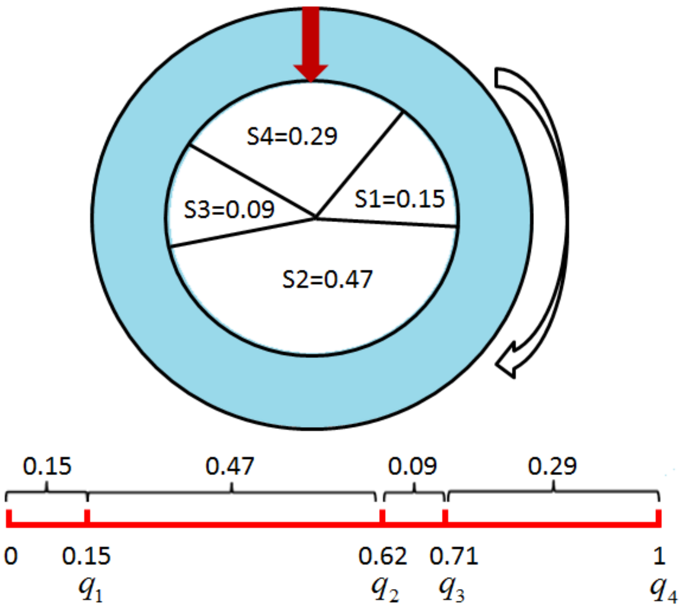
- ✓ **Combined with polynomial model of accelerator lattice, it is a good choice to realize the compensation on FPGAs.**

- ✓ **During the whole genetic algorithm, some classical methods are used in all kinds of modules, such as binary coding, "roulette wheel" selection operator, two-point intersection, multipoint mutation and linear feedback shift register in random-number processing.**





Flowchart of the genetic algorithm



An example of Roulette wheel operator

Before crossover:

Parent 1: 010100|0000001000100|0011111

Parent 2: 100110|1111010111011|1000000

intersection 1 intersection 2

After crossover:

Child 1: 010100|1111010111011|0011111

Child 2: 100110|0000001000100|1000000

An example of crossover

Before mutation:

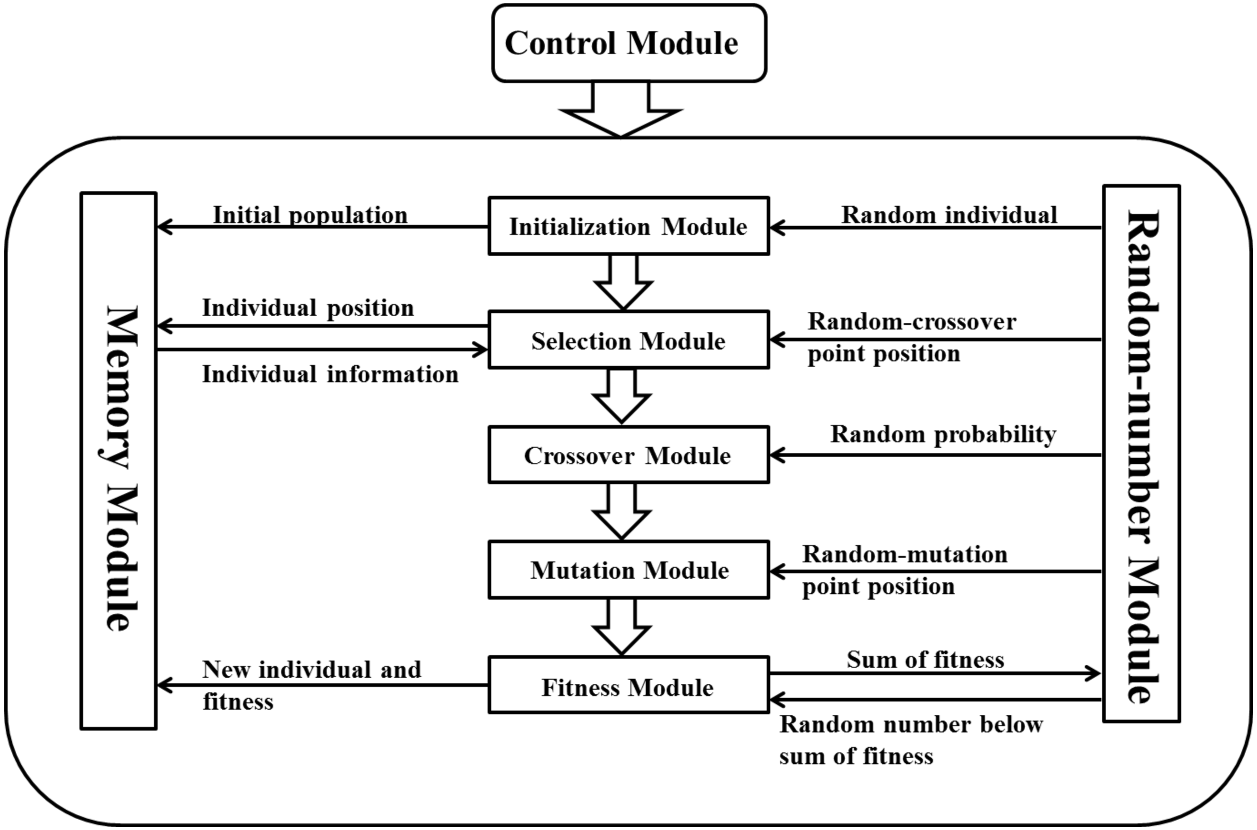
Parent: 0000001000100011111

Mutation 1 Mutation 2

After mutation:

child: 0000101010100001111

An example of mutation



Function module of genetic algorithm

1) External system gives a reset signal: random-number module begins to produce random seeds, while the control module reset to work.

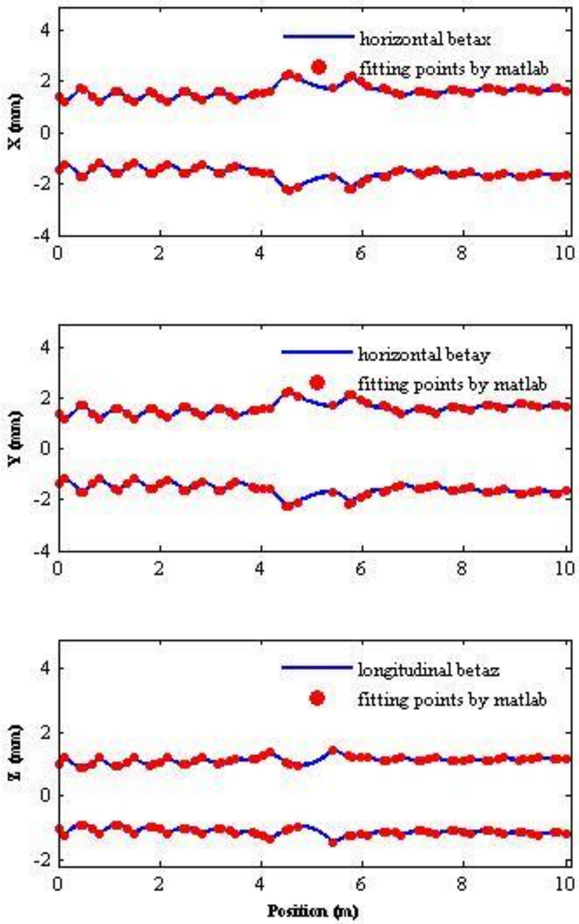
2) Initialization module start to produce the first generation of population.

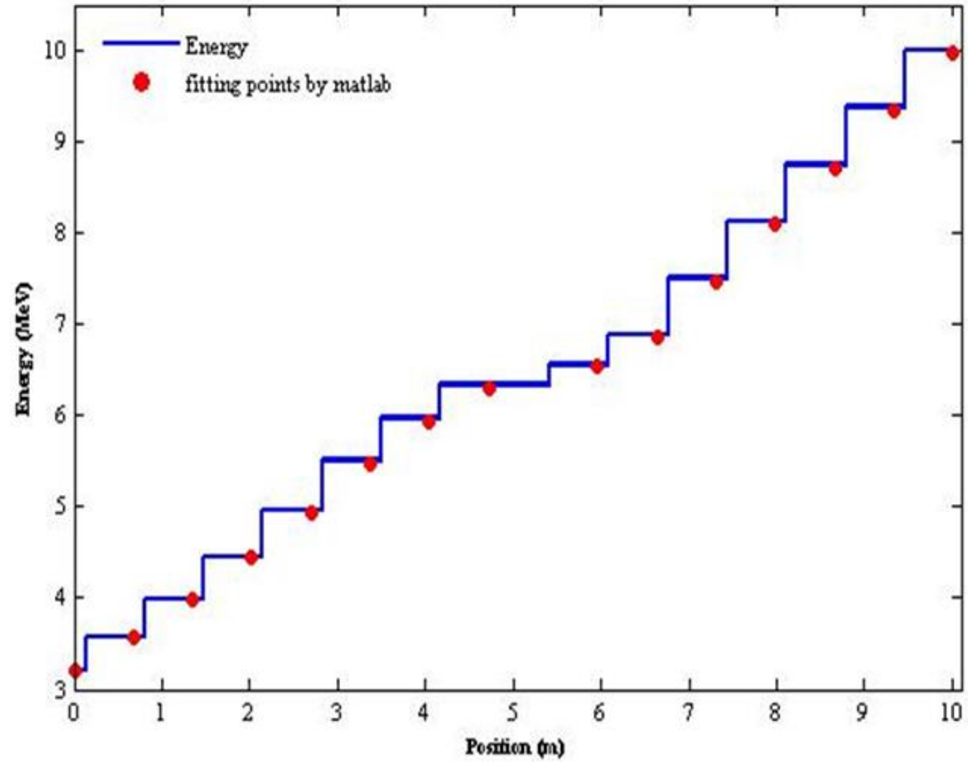
3)After initialization, control module gives a signal to step into evolutionary period.

4) Each genetic algorithm function module operates. At the same time, memory module records new individual information after selection, crossover and mutation. This step will not stop until the evolutionary calculation of this generation is over.

5) Keep the optimal solution and compare with last one. If the specified fitness level or the maximum number of generation has reached, the finished signal will be transferred. Otherwise, go back to step 4.

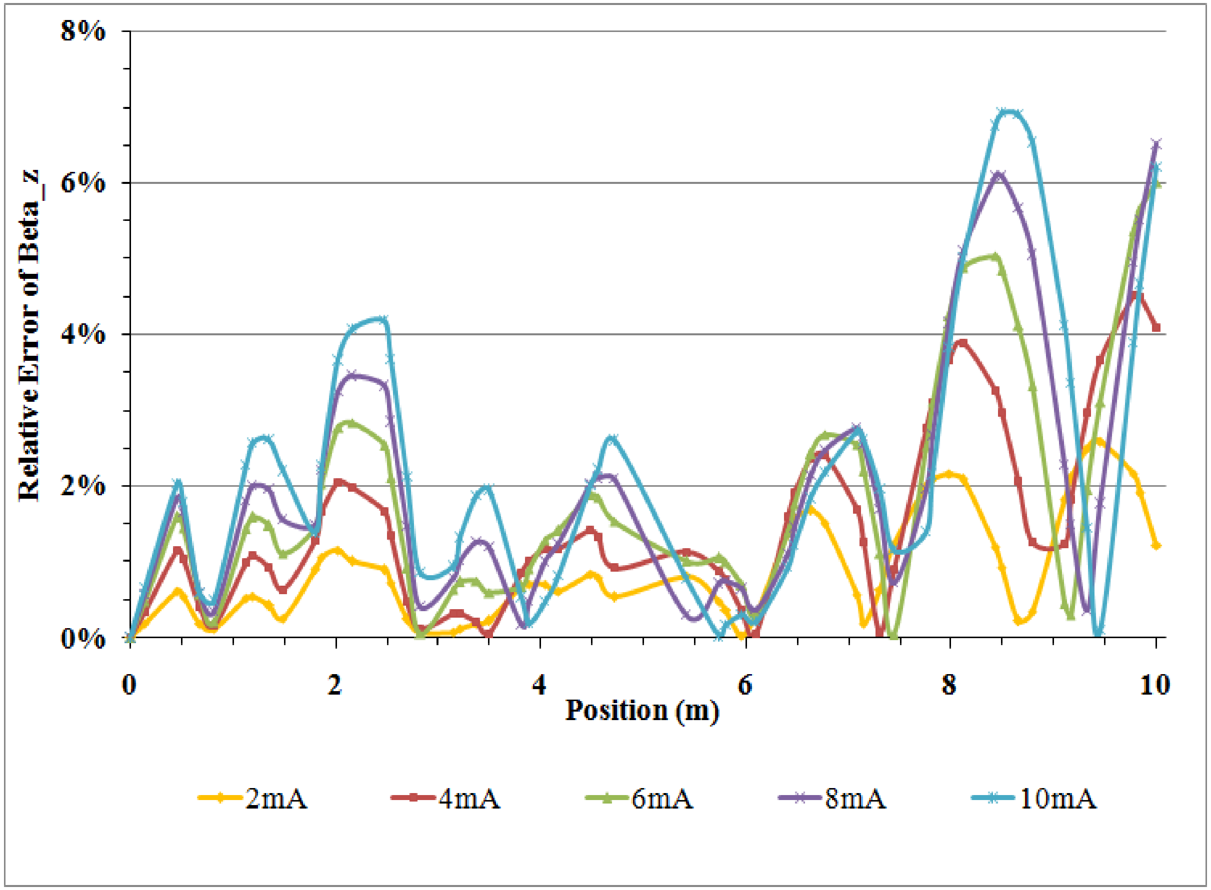
◆ TraceWin verification





Envelops and energy of C-ADS Injector-I,

calculated by TraceWin (blue line) and polynomial model (red points).

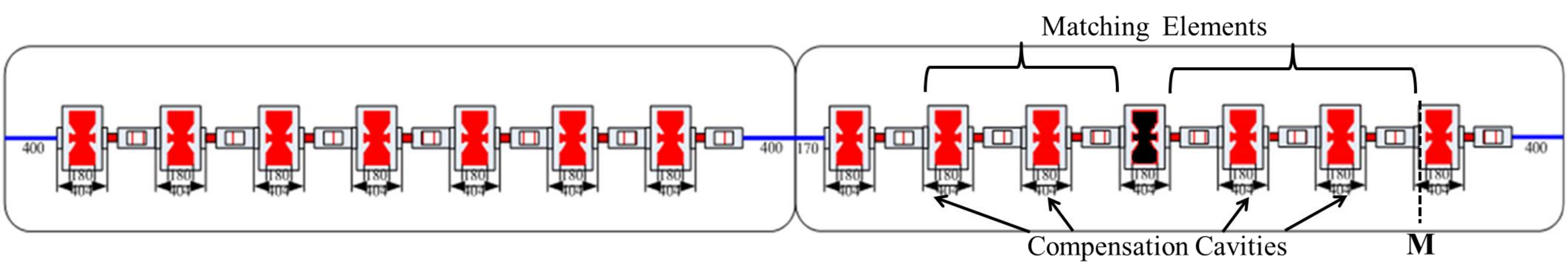


Envelops errors vs. beam current;

Compensation parameters of Cavities and solenoids of Injector-I,

(Achieved by polynomial model and generic algorithm)

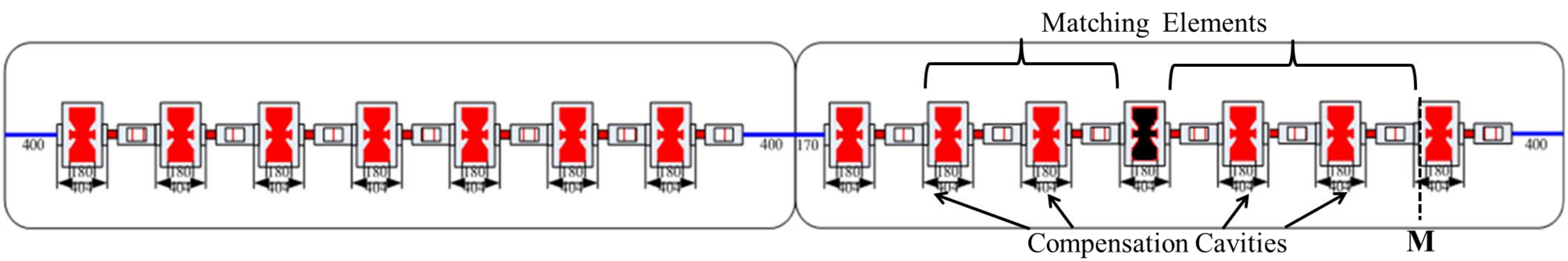
Normal	1#	2#	3#	4#	5#	6#	7#	8#	9#	10#	11#	12#	13#	14#
Cavity Eacc (MV/m)	4.0	4.5	4.9	5.3	5.6	4.7	4.5	3.7	3.3	6.2	6.2	6.3	6.3	6.3
Cavity Phase (°)	-35	-33	-31	-29	-28	-26	-43	-56	-26	-25	-25	-25	-25	-25
Solenoid B (T)	3.3	3.4	3.4	3.4	3.4	2.3	3.4	3.3	2.2	3.1	3.0	3.0	2.9	2.9



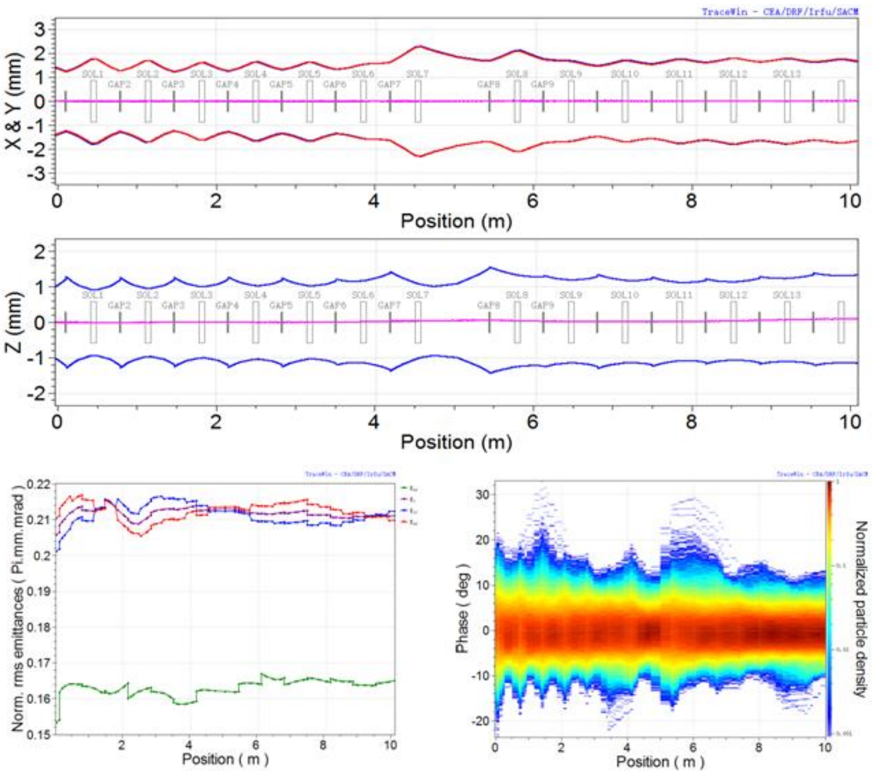
Local compensation and rematch of Spoke012-11# in Injector-I

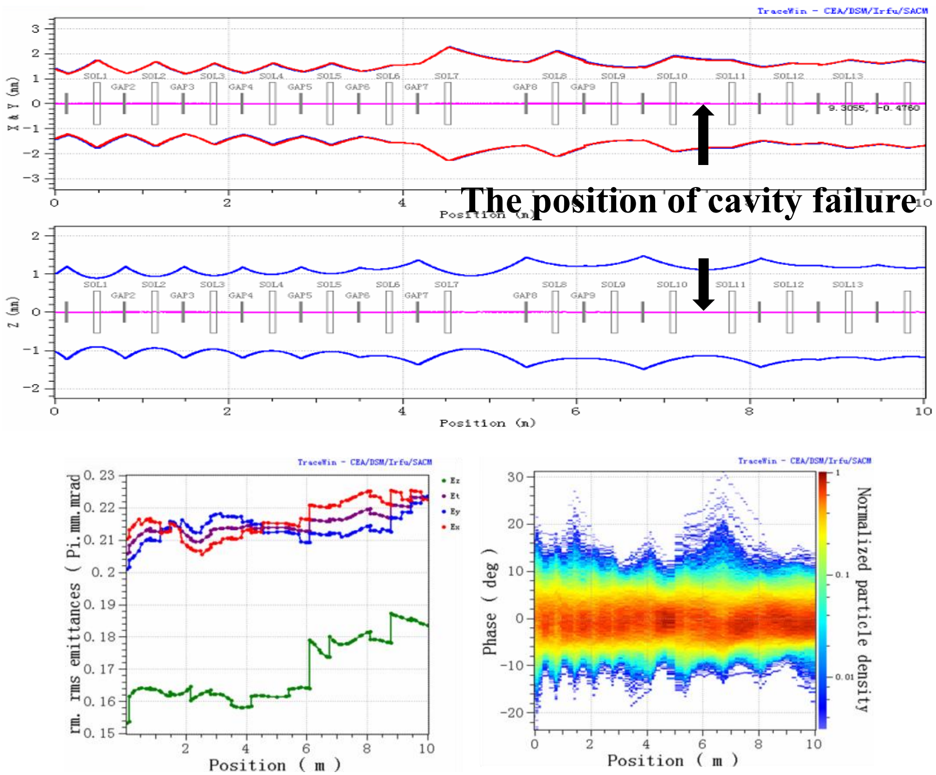
Compensation parameters of Cavities and solenoids of Injector-I,
(Achieved by polynomial model and generic algorithm)

Fault	1#	2#	3#	4#	5#	6#	7#	8#	9#	10#	11#	12#	13#	14#
Cavity Eacc (MV/m)	4.0	4.5	4.9	5.3	5.6	4.7	4.5	3.7	8.3	6.2	/	4.6	8.8	6.3
Cavity Phase (°)	-35	-33	-31	-29	-28	-26	-43	-56	-4	-32	/	-46	-10	-25
Solenoid B (T)	3.3	3.4	3.4	3.4	3.4	2.3	3.4	3.3	0.1	3.3	2.5	2.5	3.0	2.9



Local compensation and rematch of Spoke012-11# in Injector-I





Compensation results of Injector-I with Spoke012-11# failure

(envelopes, and emittances)

Compensation results of Injector-I with Spoke012-11# failure

(Comparison of beam energy and Twiss parameters)

Twiss parameters	Nominal	After compensation	Mismatch factor
Beta-x	1.9548	1.9548	3.72%
Alpha-x	0.5476	0.4683	
Beta-y	1.9856	1.9687	3.94%
Alpha-y	0.5599	0.4787	
Beta-z	1.2822	1.3623	3.90%
Alpha-z	-0.3446	-0.3181	
	Nominal	**After compensation**	**Relative error**
Beam Energy/MeV	10.008	9.976	0.320%

- ✓ **The horizontal and longitudinal envelops can be controlled in ±3mm and ±2mm.**
- ✓ **The longitudinal and horizontal emittances show about 15% and 5%.**
- ✓ **There is no greater jitter in the longitudinal phase.**

Mismatch factor

$$M = \left[1 + \frac{\Delta + \sqrt{\Delta(\Delta+4)}}{2}\right]^{\frac{1}{2}} - 1$$

$$\Delta = (\Delta\alpha)^2 - \Delta\beta\Delta\gamma$$

$$\Delta\alpha = \alpha - \alpha_m$$

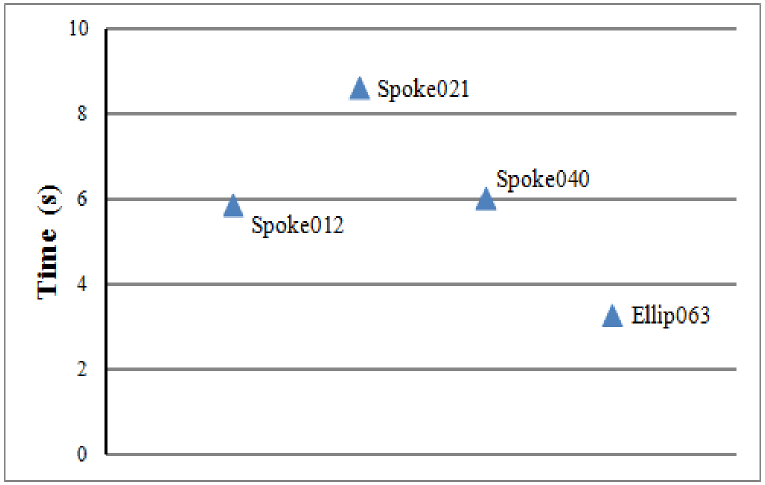
$$\Delta\beta = \beta - \beta_m$$

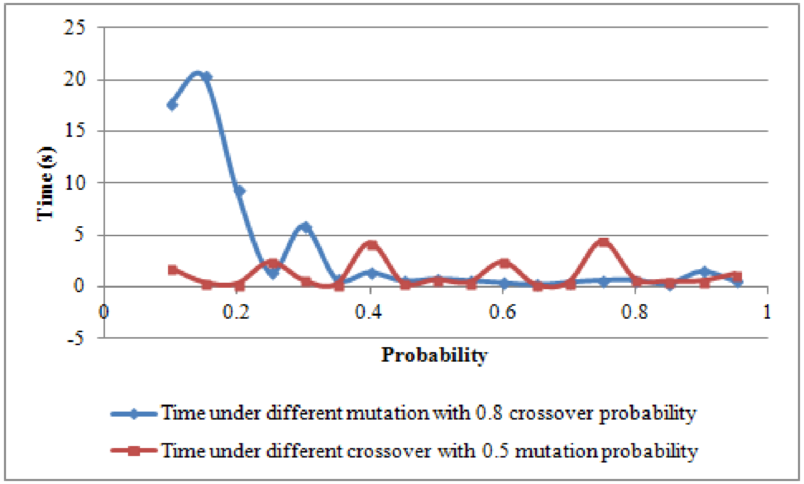
$$\Delta\gamma = \gamma - \gamma_m$$

◆ Computation time

- ✓ Taking no account of the space charge effect, the time of calculating horizontal and longitudinal lattice for Injector-I are 695 ns and 270 ns respectively, with FPGAs operated at 200 MHz clock.
- ✓ With space charge effect, the time to calculate the lattice is up to 392 us.
- ✓ Time needed by TRACEWIN simulation is typically ~1s

Time needed to get the optimal solutions for different energy sections





Summary

- **Extremely high reliability is one of the key requirements for C-ADS SC proton linac.**
- **Fault-tolerance capability is essential in addition to "strong design" and a high degree of redundancy.**
- **A new on-line computation method for cavity failure compensation was proposed and investigated.**
- **Polynomial model and genetic algorithm were implemented using Verilog_HDL language based on Xilinx Kintex7 series FPGA platform**
- **Next, more experimental research will be done in C-ADS Injector-I.**

Thanks for your attention!