A GUI tool for beta function measurement using MATLAB*

CHEN Guang-Ling(陈光玲)[1,2;1)] TIAN Shun-Qiang(田顺强)[1,2]
JIANG Bo-Cheng(姜伯承)[1] LIU Gui-Min(刘桂民)[1;2)]

1 (Shanghai Institute of Applied Physics, CAS, Shanghai 201800, China)
2 (Graduate University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract The beta function measurement is used to detect the shift in the betatron tune as the strength of an individual quadrupole magnet is varied. A GUI (graphic user interface) tool for the beta function measurement is developed using the MATLAB program language in the Linux environment, which facilitates the commissioning of the Shanghai Synchrotron Radiation Facility (SSRF) storage ring. In this paper, we describe the design of the application and give some measuring results and discussions about the definition of the measurement. The program has been optimized to solve some restrictions of the AT tracking code. After the correction with LOCO (linear optics from closed orbits), the horizontal and the vertical root mean square values (rms values) can be reduced to 0.12 and 0.10.

Key words SSRF storage ring, optical measurement, beta function, MATLAB

PACS 07.05.Dz, 29.20.D-, 29.50.+v

1 Introduction

The Shanghai Synchrotron Radiation Facility (SSRF) is controlled with EPICS (Experimental Physics and Industrial Control System)[1]. Generic low-level applications are provided by the EPICS toolkit, while high-level physics applications for commissioning and operation are mainly developed using MATLAB, which is nowadays becoming more and more popular in accelerator physics applications. As a matrix oriented programming language, MATLAB provides an active variable workspace, a built-in math library, powerful graphics capabilities and on-going development of new software features. As the middle layer tools of accelerator physics, AT (accelerator toolbox)[2] and MCA (MATLAB channel access toolbox)[3] toolboxes developed at SSRL and SLAC have been extensively used at several facilities, and SSRF has also adopted these toolboxes for the development of high-level applications.

One of the applications is a GUI tool named “BetaMeas” which is developed for beta function measurement. Detailed information is presented in this paper. Using this tool, some important processes for the measurement can be conveniently achieved such as the choosing of quadruples, the setting of the current that is quadrupole varied, automatic measurement, results saving/loading and the results shown after the measurement.

2 Algorithm of the beta function measurement

As an important parameter in the synchrotron storage ring, beta functions along the ring identify the optical layout status and show the variation of the beam size along the ring. The beta function measurement in the commissioning period is regarded as an indispensable process to check the symmetry and period of the optics of the ring. If a machine has aberrant beta functions, it will generate more sextupole resonances, and thus degenerate the dynamic

Received 14 May 2008

* Supported by SSRF Project

1) E-mail: chenguangling@sinap.ac.cn

2) E-mail: liuguimin@sinap.ac.cn

acceptance[4]. These effects reduce the performance of the machine. A common method to restore the symmetry and period of the beta functions is linear optics from closed orbits (LOCO)[5], which is an experimental determination of the optics using an orbit response measurement. The determination and the correction of LOCO should be checked by a direct measurement, which can be carried out by detecting the tune shifts as the strength of an individual quadrupole varies, and the beta function at the location of the quadrupole can be approximately calculated by the following formula[6]

$$\beta_{x,y} = \pm 4\pi \frac{\Delta Q_{x,y}}{\Delta k l_{\text{eff}}}, \tag{1}$$

where $\Delta Q_{x,y}$ is the tune shifts for the horizontal and vertical planes respectively, $\Delta k l_{\text{eff}}$ is the variation of the quadrupole integral strength.

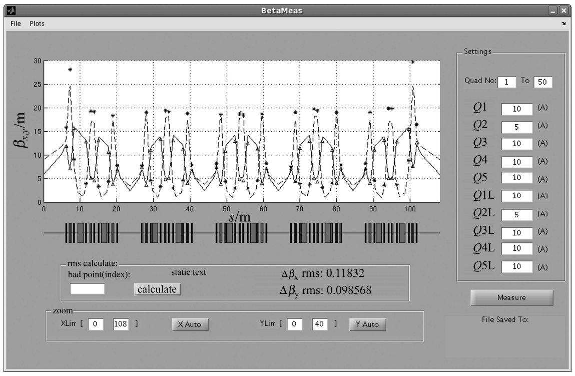
The precision of the measured beta function depends on the precision of the measured tunes. Two aspects of the tune measurement should especially be emphasized. The first one is the use of a high-accurate method for extracting the betatron tunes. The simplest method is fast fourier transform (FFT) with precision of $1/N$ (N is the number of turn-by-turn data), whereas the interpolated fast fourier method[7] with a Hanning filter and the numerical analysis of fundamental frequency (NAFF)[8] is more acceptable because of the trigger precision with $1/N^3$. The method of tune determination can be selected according to the acquirement. The second one is to suppress the de-coherence of the beam oscillation, in order to avoid the jump of the measured tune or the additional degree of the imprecision. It can be done by setting the chromaticity to zero or so and exciting the weaker kicker strength. Even the data acquirement could be done with a single bunch beam, with the higher beam current it seems to give a better resolution in the beam positions.

3 Beta function measurement program

3.1 “BetaMeas” GUI tool

As the GUI tool is run by typing “BetaMeas” in the MATLAB command line, a data structure called “QuadChannels” is established automatically. As MATLAB application-defined data, “QuadChannels” is subjected to the handle of the figure object, built by the module ChannelsInit() which performs channels initialization according to the arrangement of 200 quadruples.

Figure 1 shows the “BetaMeas” GUI tool. In the setting panel rightward, the range of measurement can be manually set by choosing the serial number of the quadruples, 1—50 (one super-period of SSRF) for default. The current shift of each family of quadruples can be edited to avoid an instable optics or beam loss caused by the modification of some sensitive



bad point(index):

$\Delta\beta_x$ rms: 0.11832

$\Delta\beta_y$ rms: 0.098568

XLim [0 108]

YLim [0 40]

Quad No: 1 To 50

Q1 10 (A)

Q2 5 (A)

Q3 10 (A)

Q4 10 (A)

Q5 10 (A)

Q1L 10 (A)

Q2L 5 (A)

Q3L 10 (A)

Q4L 10 (A)

Q5L 10 (A)

Fig. 1. The beta function measurement GUI tool.

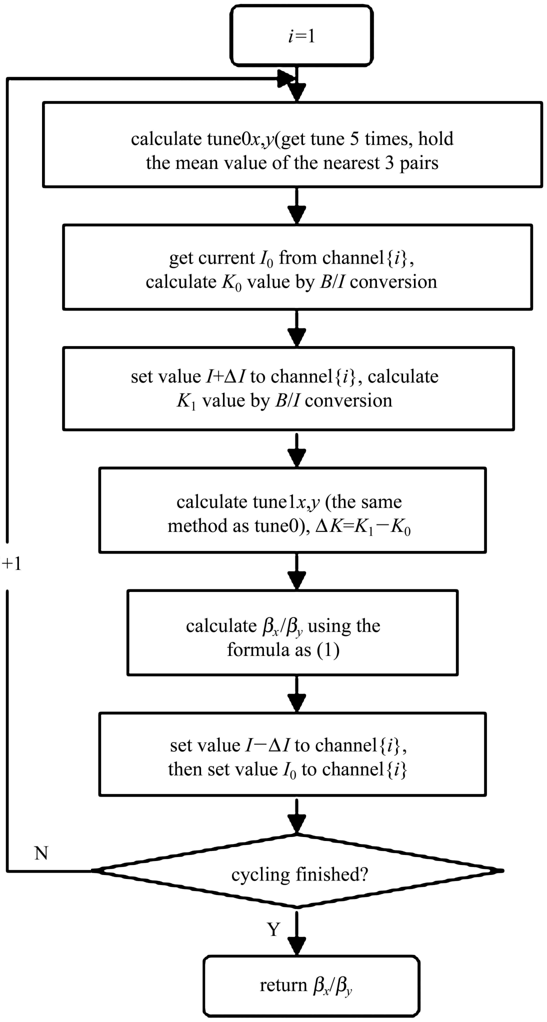

i=1
calculate tune0x,y(get tune 5 times, hold the mean value of the nearest 3 pairs
get current I_0 from channel{i}, calculate K_0 value by B/I conversion
set value I+ΔI to channel{i}, calculate K_1 value by B/I conversion
calculate tune1x,y (the same method as tune0), ΔK=K_1−K_0
calculate β_x/β_y using the formula as (1)
set value I−ΔI to channel{i}, then set value I_0 to channel{i}
cycling finished?
N
Y
i+1
return β_x/β_y

Fig. 2. The flow chart of the measurement module.

quadruples. It should be noted that we just set the current shift for one family, and in the process of measurement, the excited current of an individual quadrupole will be changed one by one. After the settings of the quadruples shift, press the measure button to call the automatic measurement process. Fig. 2 shows the flow chart of the measurement module, in which channel denotes the data structure of the chosen channels from "QuadChannels" according to the range of measurement, and the betatron tune is calculated by the interpolated FFT to the acquired beam position monitor (BPM) turn by turn data provided from the diagnosis system.

When the measuring work is down, modelled and measured beta functions can be plotted in the axes, and rms values can be calculated and shown below the axes. In Fig. 1, the broken line and the real line denote the horizontal and vertical beta functions, modelled by twissring, a tracking function in AT. The stars and triangles denote the measured values at the locations of quadruples.

A useful function in the tool is the management of results data. Once the measurement is finished, the results file named by the current date can be saved automatically in an appointed directory, and correspondingly, the results data can be loaded and plotted by clicking the file → open menu.

3.2 Machine-independent

"BetaMeas" GUI tool can well apply to SSRF, but also easily port to other machines using MATLAB for high level applications. The configuration work includes two aspects, the initialization of quadruple channels and the special settings in the tune measurement module.

1) Channel initialization: Before the "BetaMeas" GUI tool is used, the cell data structure "Quad" Channels should be redefined in the module ChannelsInit() according to the quadruple arrangement.

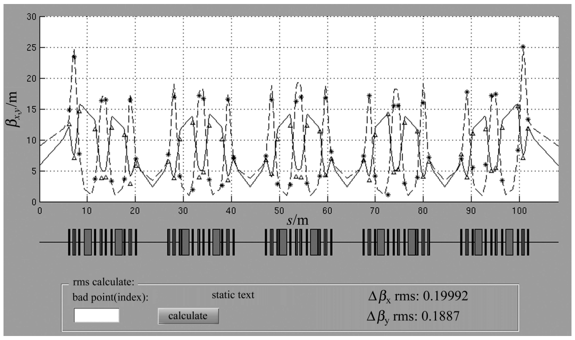
2) Settings of the tune measurement module: In the tune measurement module of "BetaMeas", the betatron tune is calculated by the interpolated FFT to the acquired BPM turn by turn data provided from the diagnosis system. A simple and operational way to achieve the function of machine-independent is to set a couple of TUNE channels in the accelerator object[9] initial file. One of them is used as the set channel operated by the diagnosis system, and the other for physics applications is used as a get channel. Thus two branches have been programmed in the gettune() module of "BetaMeas", one for getting the tune values via TUNE channels, and the other for the user-defined interface of the tune measurement codes.

4 Beta function measurement results

Quadrupole is not a thin-lens because of its nonzero length ($l_q \neq 0$), so the measured beta function is the mean value inside the individual quadrupole, and it can be described as the following formula

$$\beta_{\text{measure }x,y} = \frac{1}{l}\int_{l_{\text{quad}}} \beta_{x,y}\mathrm{d}l\ . \qquad (2)$$

With the restriction of the AT code, the beta functions only in the entrance and exit of an element can be tracked out. Imprecision is obvious. Fig. 3 shows an earlier measurement result with the rms values of 0.19992 and 0.1887.



rms calculate:

bad point(index):

static text

$\Delta\beta_x$ rms: 0.19992

$\Delta\beta_y$ rms: 0.1887

Fig. 3. An earlier measurement result with the rms values of 0.19992 and 0.1887.

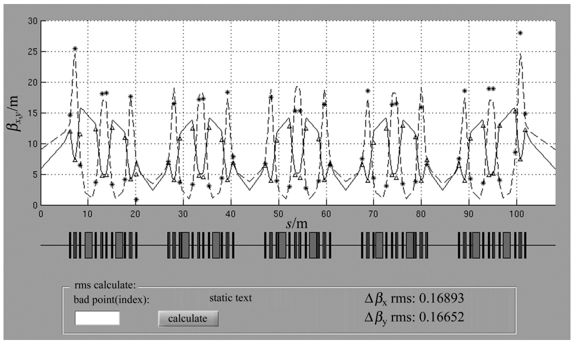
Assuming the beta function in the center of quadrupole takes the extremum β_0, then $\alpha_0 = 0$, and $\nu_0 = \dfrac{1}{\beta_0}$, the beta function at the exit of quadrupole can be described as the following formula

$$\beta = \cos^2\sqrt{K}\Delta L\beta_0 + \frac{1}{K}\sin^2\sqrt{K}\Delta L\frac{1}{\beta_0}. \qquad (3)$$

In the reformative edition, quadrupoles are all divided into two elements ($Q1$ and $Q2$) with the same strength and effective length. We take the mean value of the 4 points below as the model beta function of each quadrupole

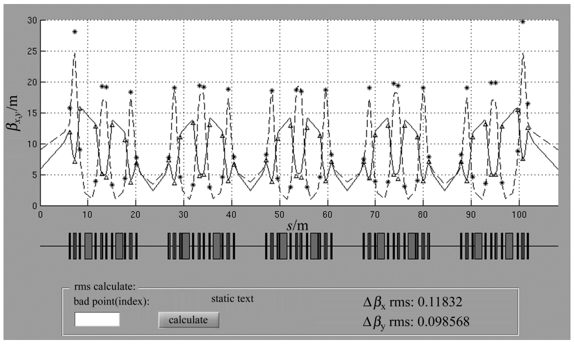
$$\beta \approx \frac{\beta_{Q1_{\text{in}}} + 2\beta_{Q1_{\text{out}}} + \beta_{Q2_{\text{out}}}}{4}\ . \qquad (4)$$

In SSRF, for example, the length of the longest quadrupole is 580 mm, and the value of K is 1.53 T/m, the error between the mean value of beta integral and the mean value of 4 points is less than 3%. The rms values can be reduced to 0.16893 and 0.16652 as shown in Fig. 4.



rms calculate:
bad point(index): static text
$\Delta\beta_x$ rms: 0.16893
$\Delta\beta_y$ rms: 0.16652

Fig. 4. In the reformative edition the rms values are reduced to 0.16893 and 0.16652.



rms calculate:
bad point(index): static text
$\Delta\beta_x$ rms: 0.11832
$\Delta\beta_y$ rms: 0.098568

Fig. 5. After LOCO, the rms values are reduced to 0.11832 and 0.09856.

LOCO has been used in SSRF to find bad BPMs, to calibrate BPM gains, to find quadrupole gradient errors and to restore the lattice periodicity. After LOCO, beta functions have been measured again using the BetaMeas GUI tool, rms values can reduce to 0.11832 and 0.09856 as shown in Fig. 5.

5 Conclusions

The “BetaMeas” GUI tool using MATLAB is a good tool for the measurement of beta function in the SSRF commissioning. The measurement process is completely automatic and the results are reasonable. The program has been optimized to solve some restrictions of the AT tracking code. After LOCO, the horizontal and the vertical rms values of the measurement result differing from the designed values can reduce to 0.09 and 0.16. It is convenient to check the symmetry and period of the optics during commissioning. The GUI tool can easily be applied to SSRF, but can also be easily ported to other machines using MATLAB for high level applications.

References

1 www.aps.anl.gov/epics/

2 Terebilo A. Accelerator Modeling with MATLAB Accelerator Toolbox. In: PAC. Chicago, USA, 2001. 3203—3205

3 Terebilo A. Channel Access Toolbox for MATLAB. In: ICALEPCS. San Jose, USA, 2001. THAP030

4 Robin D, Safranek J, Decking W. Physical Review Special Topics-Accelerators and Beams, 1999, **2**: 044001

5 Safranek J. Nuclear Instrument and Method in Physics Research A, 1997, **388**: 27

6 Zimmerman F. Measurement and correction of accelerator optics. SLAC-PUB. 1998, **7844**: 21

7 Bartolini R, Giovannozzi M, Scandale W. Algorithms for a Precise Determination of the Betatron Tune. EPAC. 1996. 1329—1331

8 Laskar J. Introduction to frequency map analysis. PAC. Portland Oregon USA, 2003. 378—382

9 Corbett J, Portmann G, Terebilo A. Accelerator Control Middle Layer. PAC. Portland Oregon, USA, 2003. 2369—2371