Introduction

A high energy proton synchrotron, operating as an accelerator for fixed target physics has often a very simple and periodic structure at least at the design stage. For instance the CERN SPS working in this mode has 108 identical 3900 sections. For a given plane, horizontal or vertical, beam position detectors (BP), all identical, and closed orbit correction dipoles (COD), also identical, can be located near the quadrupoles where the betatron amplitude ($B$) and the phase advance ($\psi$) are constant from one element to the next. The closed orbit correction (only needed at low energy) is then straightforward using the simple beam bump method on three adjacent COD's.

Transforming such an accelerator into a proton-antiproton ($\bar{p}p$) collider significantly alters its periodicity. To increase the luminosity, low beta configurations are inserted at the crossing points, changing $B$ and $\psi$ around the ring and modifying the physical layout of BP and COD. New types of BP are required locally to detect $p$ and $\bar{p}$ as well as new types of COD to correct the closed orbit around each energy inside the insertions. Furthermore the working points and the low beta configurations can change from one experiment to the other.

For synchrotrons having such an irregular and variable configuration, it becomes necessary to define a software package independent of the configuration. Fast algorithms are needed to correct the closed orbit (on-line). These algorithms should take into account the finite strength of the COD's and the fact that at some places there are not enough COD's available for the beam bump method at high energy.

Software Package

The software package is made by three main software units:

- off-line programs for generating the closed orbit (CO) data bases and for extracting from them the working file corresponding to the desired machine configuration,
- on-line programs for acquiring and correcting the CO from the SPS control room,
- locally distributed real-time routines for controlling the CO hardware.

A BP consists of two electrodes, the difference of the two signals ($E$) and their sum ($I$) being transmitted to the acquisition via two electronics channels. Having individual electronics chains by BP, terminated by a hardware analog or digital memory allows one to obtain a CO acquisition all around the synchrotron at a given predetermined timing. Using fast real time drivers to transfer data from the hardware memories to a local computer allows several acquisitions at different timings to be carried out systematically during each machine cycle. On request, a pretreatment is made in the local computer such as computing the position in mm from the raw data ($E$ and $I$) and from a calibration factor depending on the type and diameter of the BP.

This parallel processing provides a local data compression and faster execution. The same applies for the CO correction where the reference voltages of the COD power supplies are obtained from hardware memories modified by the local computers, when a new correction has been calculated at the control centre. Among the COD's those which are used for CO correction at any energy are pulsed. The corresponding function of time is generated by hardware from digital tables. When a new correction is needed, new tables are sent in user format (amplitude-duration) converted and distributed locally. At this level all the routines are mainly written in assembler or in compiled language but they have headers allowing them to be linked by an interactive language, called NODAL, into application programs which can be executed from the SPS control room.

There, the CO acquisition and correction programs are executed on-line. Their skeleton is written in NODAL for distributed execution in local computers. To speed up acquisition and correction calculations, they use subroutines written in FORTRAN. The embodied algorithms for CO correction are described in the following sections.

To be independent of machine configuration these programs use a "working file" containing the parameters of the CO elements needed for the current status of the accelerator.

This working file is obtained from two data bases. A "hardware data base" deals with all the CO elements installed around the accelerator in increasing order of $\mu$. It contains some basic information such as the responsible local computer, pointers, etc. This data base is unique and only updated if the physical layout changes, for instance, insertion of new detectors or correctors. A "lattice data base" giving the $B$ and $\mu$ values of each CO element. This data base depends on the focusing configuration; it is generated by off-line programs for any new working point or low beta insertion value and stored in a file on a library computer.

Knowing the accelerator status and the corresponding CO elements, the working file is generated on-line from those two data bases. Within one SPS cycle, the focusing configuration can change, for instance, when the low beta values are squeezed on the high energy flat-top. Therefore even within the cycle the working file is automatically updated.

Fast implementation of the least squares method

A frequently used method to correct the closed orbit is based on the least squares criterion.

It is particularly attractive when the number of deflectors available or desired is, or should be, less than the number required by the beam bump method. However the exact solution, which exists in theory, requires solving a large linear system. This may be quite time consuming and numerically unstable. An iterative strategy based on the successive solution of linear systems, which are larger from one step to the next, has been developed and successfully implemented2).
The method proposed here is also iterative, but avoids altogether the task of solving large linear systems. At each stage it finds the best deflector which minimizes the sum of the squared deviations of the predicted closed orbit obtained by simulating the effect of the kick computed in the previous iteration. Let \( \mathbf{y}_n(j) \) be a vector representing the \( N \) closed orbit positions (in either the horizontal or the vertical plane) obtained at the \( n \)-th stage, by adding only one kick located in the \( j \)-th position to the closed orbit vector \( \mathbf{y}_{n-1} \) of the previous stage:

\[
\mathbf{y}_n(j) = \mathbf{y}_{n-1} + \mathbf{x}_n(j) \mathbf{a}_j
\]

where \( \mathbf{x}_n(j) = \frac{p_j}{2} \sin \pi Q \)

\( p_j \) being the best deflection which the \( j \)-th corrector can provide, \( Q \) the betatron frequency and \( \mathbf{a}_j \) the \( j \)-th column of the matrix \( \mathbf{A} \) with elements

\[
a_{ij} = \frac{\sin((Q-\mu_j-\mu_i))}{\sin((\mu_j-\mu_i))}
\]

At each iteration we should find the best corrector among the \( I = \ell(n) \) available, \( \mathbf{x}_n = \mathbf{x}_n(n) \) which minimizes the sum \( S \) of the squares of the \( N \) closed orbit positions

\[
S[\mathbf{x}_n(n)] = \sum_{j=1}^{N} (\mathbf{y}_n(j) - \mathbf{y}_n(n))^2
\]

where \( \mathbf{y}_n^T \) is the transpose of \( \mathbf{y}_n \).

Deflections obtained for the same corrector at different iteration stages are simply added up by the virtue of the linearity of (2.1). It is easy to prove that for each corrector at the \( j \)-th location, \( S[\mathbf{x}_n(j)] \) has a minimum for:

\[
\mathbf{x}_n(j) = -\mathbf{W}(j,j)^T \mathbf{y}_n(j)
\]

where \( \mathbf{W}(j,j)^T \) is the transpose of \( \mathbf{W} \).

The location \( \mathbf{x}_n(j) \) corresponds to the index \( j \) for which

\[
S[\mathbf{x}_n(j)] = \min \ S[\mathbf{x}_n(n)]
\]

where \( \mathbf{C} \) maps the set of the available correctors. Let us note that \( \mathbf{x}_n \) will also minimize

\[
- \left[ \mathbf{T}_n(j) \right]^2 / \mathbf{W}(j,j).
\]

Using (2.1) \( \mathbf{T}_n(j) \) may be expressed recursively:

\[
\mathbf{T}_n(j) = \mathbf{T}_{n-1}(j) + \mathbf{x}_{n-1}(j) \mathbf{W}(j,n-1)
\]

Thus for each \( j \) identifying a deflector only a multiplication and a sum has to be computed at each iteration stage provided \( M \) is small enough to permit storing the matrix \( \mathbf{W} \).

It is also easy to show that

\[
S[\mathbf{x}_n(n)] - S[\mathbf{x}_n(j)] = S[\mathbf{x}_n(n)] - S[\mathbf{x}_n(j)]
\]

which proves the convergence of the method because

\[
\mathbf{T}_n(j) \mathbf{x}_n(n) \]

is always negative by virtue of (2.4).

The first step in the iteration (2.1) is obtained by identifying \( \mathbf{y}_0 \) with the measured closed orbit in either the horizontal or vertical plane.

The iteration is stopped according to two criteria, whichever comes first. These are:

- the residual sum \( S[\mathbf{x}_n(n)] \)
- or the difference \( S[\mathbf{x}_{n-1}(n)] - S[\mathbf{x}_n(n)] \)

are less than some prefixed values. The second criterion is very useful in sparing lengthy computations to obtain only a very small decrease in the residual sum. Of course the number of iterations is also limited to avoid endless looping.

**Constrained beam bump method**

As we have shown, it is in principle possible to obtain a corrected closed orbit which is very close to the theoretical least square solution.

However, it is much faster to use the beam bump method extended over as many deflectors as are available, wherever the machine configuration permits it.

The classical beam bump method may strongly perturb the closed orbit between beam position monitors if the correctors have a very different phase advance from the associated monitors. This may happen even in a machine designed to have the correctors very close to the monitors, when a corrector is out of service or when the phase advance function is strongly modified locally to meet special requirements, as in the low beta insertions.

Furthermore, \( \beta \) and \( \mu \) may be such that the strength required from a corrector to fully correct the local orbit displacement is not available.

First of all it is easy to show that the building block of the beam bump method, i.e. the fact that the orbit displacement detected by a monitor can be fully compensated by three adjacent correctors, can be extended to an arbitrary number \( N \) of correctors and orbit displacements \( M \), provided that \( N \geq M+2 \). The linear system to be solved will be of \( N \)-th order instead of third order. The closed orbit outside the section containing \( N \) correctors will also not be perturbed. The linear system can be reduced to \( L=N-2 \) equations by a simple modification to the elements of the system matrix:

\[
\mathbf{A} = \begin{pmatrix} \mathbf{A} \end{pmatrix} - \begin{pmatrix} \mathbf{W} \end{pmatrix}
\]

where \( \mathbf{W} \) and \( \mathbf{y} \) are vectors representing respectively the desired deflections and the measured orbit displacements and \( \mathbf{A} \) is a symmetric matrix with elements

\[
a_{ij} = \frac{(\cos(\mu_i-\mu_j))(\sin(\mu_j-\mu_i))}{(\sin(\mu_i-\mu_j))}
\]

\[
\mathbf{A} = \begin{pmatrix} \mathbf{A} \end{pmatrix} - \begin{pmatrix} \mathbf{W} \end{pmatrix}
\]

where \( \mu_i \) and \( \mu_j \) are respectively the phase advances of the first and last correctors in the considered section of the accelerator.

In practice the achievable deflections are bounded, i.e.

\[
|K(j)| \leq \| \mathbf{E} \|
\]
For a missing deflector $K_j$ is null. Let us consider the case when one or more $K_j(K_0)$ obtained from (3.1) do not satisfy (3.3). Let $K_n(K_{0n})$ be the largest of them.

The system (3.1) may also be written as

$$A_1 \tilde{K}_1 = - \tilde{y} - A_1 K_0(K_0)$$

where $A_1$ is the matrix $A_0 - \alpha$ from which the $\theta_0$-th column has been removed and $\tilde{K}_1$ a vector of $L - 1$ dimension. It is easy to prove that the norm of the residual vector is

$$S_1(K_1) = \left| (A_1 \tilde{K}_1 + y - C_0(K_1) - K_0(K_0)) \right|^2$$

where $C_0$ depends on which solution of (3.4) is taken. To minimize $S_1(K_1)$, we should choose $K_1(K_0)$ as close as possible to $K_0(K_0)$, i.e.

$$K_1(K_0) = \tilde{K}(K_0) \text{ sgn} (K_0(K_0))$$

and choose the least square solution of (3.4) which is given by solving

$$A_1^T A_1 K_1 = - A_1^T (\tilde{y} + K_1(K_0) \tilde{w}_1)$$

where $A_1^T$ is the transpose of $A_1$. Noting that

$$A_1^T A_1 K_0 = - A_1^T (\tilde{y} + K_0(K_0) \tilde{w}_1)$$

we may also write

$$A_1^T A_1 (K_1 + K_0) = - A_1^T (\tilde{w}_1 (K_1 - K_0))$$

The procedure may be iterated until all deflections comply with (3.3). At each step $n$ the $K_n$ of dimension $L - n$ vector is obtained by solving the system

$$A_{n-1}^T A_{n-1} (K_{n-1} - K_{n-1}(K_{n-1})) = - A_{n-1}^T (\tilde{w}_n (K_{n-1} - K_{n-1}(K_{n-1})))$$

where $A_{n-1}$ is the matrix $A_n - \alpha$ from which the $\theta_{n-1}$-th column has been removed and $K_{n-1}(K_{n-1})$ being the largest deflection for which $|K_{n-1}(K_{n-1})| > \tilde{K}(K_{n-1})$.

Experimental results

The strategy for orbit correction described in the previous section has been very successful in achieving very small deviations inside the low beta insertions while avoiding large orbit excursions outside. Fig. 1 shows the bare vertical orbit and the one obtained after using the two methods in succession. Fig. 2 shows in detail the small orbit deviations inside an insertion for both planes. The mimic diagram at the bottom shows the relative locations of the quadrupoles for each position monitors. The acquisition of the whole beam position is done within one SPS cycle. The constrained beam bump algorithm over 30 correctors takes less than two machine cycles and the best kick method over the same number of correctors and 108 beam position monitors about 1 min.

Acknowledgements

We are indebted to A. Faugier, J. Miles and H. Michel for their contributions to the software package.

References