Beam threading for the MAX IV accelerators

Georgi Georgiev

Thesis submitted for the degree of Master of Science
Project duration: 4 months

Supervised by Magnus Sjöström and Sverker Werin

Department of Physics
MAX IV Division
May 2018

Abstract

Large particle accelerators are difficult to setup. One of the first problems that can be encountered, before production configuration is found, is the beam not passing through the accelerator. The process of guiding the beam through the accelerator is called beam threading. Depending on the accelerator complexity it can take days before the beam passes for the first time.

Currently the beam threading at MAX IV is performed manually. In addition to the long average time spent, the procedure of manual beam threading is labor-intensive and tiresome. An automated solution offers the advantage of greatly speeding up this process saving both time, effort and resources.

For this project an automated beam threader was developed for the 3 GeV and 1.5 GeV storage rings at MAX IV. The algorithm optimizes the available corrector magnets and uses beam position monitors or an oscilloscope to get an instant information on its progress. This way only minor effort by control room personnel is needed.

The typically low signal to noise ratio in the used diagnostics made noise filtering necessary. A successful decrease of the noise levels was achieved with spectral filters. Robust counting of the number of completed turns by the beam during threading from the filtered signal is developed. It is used by the automated beam threader as a metric of progress.

The automated beam threader successfully functions on the MAX IV storage rings and in simulation. It performs significantly faster than the manual threading procedure and in early attempt it outperformed the fastest manual threading for half of the time of the latter.

Contents

Chapter 1

Introduction

The main product of particle accelerators are beams of high energy particles, which can be used in wide range of scientific fields. They have influenced studies from the fundamental interactions of matter by particle collisions to the understanding of natural processes and chemical structure by using synchotron light.

The operation of accelerators is not without interruption. Upgrades or corrections to the accelerators are occasionally needed during maintenance periods. In that time the scientific studies performed with that accelerator are not possible.

After a maintenance is complete, it is not unusual that the accelerator is unable to accelerate beam due to unknown defects mistakenly introduced. The beam then may not even pass through the accelerator, but it can be possible to guide it.

1.1 Beam threading introduction

Beam threading is a general name for procedures that correct the beam trajectory, so the accelerated particles can pass further through a linear accelerator or a circular accelerator, before being lost. It is often needed during the commissioning of an accelerator or after some unwanted and unknown accelerator hardware change prevents the beam from passing through with the last known good configuration.

The goal of the automated beam threading is to deliver equivalent results faster to those of manual beam threading with minimal human interaction. This may be accomplished by a method that optimizes the corrector magnet configuration until a single turn is completed, then increases the number of turns by optimizing the corrector magnet configuration further and finally captures a beam with accelerating structures turned on. This method is the one used in manual beam threading as well.

There are extra conditions for the automated beam threader (ABT). First, it is advisable to use a black box optimization algorithm. This makes the algorithm independent from the accelerator optics, because the black box optimizer does not rely on the accelerator model. Therefore the automated beam threader may be used on many different accelerators. Additionally it allows the ABT to be used successfully on unknown accelerator models and hardware. The accelerator that is threaded has unknown changes from its model and methods relying on calculations from the model may not work. An example of automated beam threading procedure that uses the accelerator model, was developed for the large hadron collider at CERN [1].

As a second condition, a robust method that can be applied with high levels of noise is required. Because of the expected low intensity of the beam signal, the background

noise becomes a significant problem. The filtering and the handling of this noise are a must to minimize the fluctuations of the objective that is optimized. Signal processing is one approach. Another one is choosing an optimization algorithm that is robust. These solutions are not mutually exclusive and both can be used in conjunction.

Finally, the implementation of the automated beam threader has to be developed on Matlab with use of the Matlab middle layer software package. This requirement ensures that the code can be integrated in the MAX IV control system.

A major constraint on the project was the time. The time to fully develop and perform an exhaustive study on an automated beam threader is longer than what was allowed. Furthermore, the run time is limited and mostly reserved to specific activities relevant to the goal of MAX IV facility to deliver synchrotron light to users. To adapt the project to this major constraint, focus was put on the development of automated beam threader exclusively for the storage rings at MAX IV. Later, as the development progressed, simulation and ultimately experimental studies on specific cases were performed.

Chapter 2

Accelerators

2.1 General introduction

The accelerators are human made sources of high energy particles with very well defined properties. Naturally, relativistic particles are produced in ways that are not reliable or tunable. For precise and demanding tasks artificial sources of particle rays or beams are needed. This demand can be fulfilled with accelerators.

The accelerators are often classified by their design principles. First there are two main types of accelerators: linear accelerators and circular accelerators. In linear accelerators, linacs for short, the accelerated particle travels once along a relatively straight trajectory. The particle trajectory in a circular accelerator is a spiral or a closed orbit, meaning that the particle travels along a curved path and can encounter the same accelerator structures multiple times.

Depending on the kind of particles being accelerated, the accelerators can be divided into low mass particle accelerators and high mass particle accelerators. Low mass particles are usually electrons, while high mass particles are protons or ions. The particle mass influences significantly the dynamics of acceleration. Low mass particles become highly relativistic at much lower energies than high mass particles, so the relativistic dynamics apply early during their acceleration. Also low mass particles emit significant power as synchrotron radiation or synchrotron light, an effect that is covered in more detail later in this chapter.

2.2 Synchrotrons

The synchrotron is a circular accelerator with the main advantage of being capable of providing very high beam energies. The trajectory of the accelerated particle is a fixed orbit. The energy of the particle and the bending magnetic field increase synchronously in a synchrotron, thus the name of the accelerator. In principle, a synchrotron can increase the energy of a particle until some limit on the magnetic field or on the power loss from synchrotron light is reached.

The arrangement of the magnetic components and the accelerating structures of an accelerator is called a lattice. The lattices of synchrotrons commonly have multiple symmetry points and consist of repeated structures of magnetic components. These structures are called achromats [2, 3].

Some synchrotrons are used to accumulate and store a beam of particles for long

periods of time at constant energy and they are called storage rings [2]. Accumulating a stable beam is used to achieve high currents. The storage rings require precisely controlled magnetic fields, while accelerating structures are only needed if the power loss from synchrotron radiation must be compensated.

2.3 Theory

This section of the report presents part of the fundamental accelerator physics theory and proceeds to explain relevant effects.

2.3.1 Coordinate system and transverse dynamics

The coordinate system used to describe the particle and beam dynamics is a co-moving coordinate system. The origin of the coordinate system travels along the ideal trajectory. An idealized particle model, called the reference particle, is always at the origin of the co-moving coordinate system.

Bending magnetic fields are used in circular accelerators to bend the beam trajectory into a spiral or a closed orbit. The coordinate system follows the bent trajectory or the reference orbit. The linear equations of motion in a co-moving coordinate system in the absence of kicks are [2]

$$\begin{aligned} x''(s) - k(s)x(s) &= 0 \\ y''(s) + k(s)y(s) &= 0 \end{aligned} \tag{2.1}$$

with s as the distance along the reference trajectory, x'' and y'' as the second derivatives over s of the horizontal and vertical position respectively. The focusing strength k is also function of s. By convention $k < 0$ corresponds to focusing and $k > 0$ to defocusing. The equations have the form of Hill's differential equation of motion. The solution for the horizontal position gives [2]

$$\begin{aligned} x(s) &= \sqrt{\epsilon}\sqrt{\beta(s)}\cos[\Psi(s) + \phi] \\ x'(s) &= -\frac{\sqrt{\epsilon}}{\sqrt{\beta(s)}}\Big(\alpha(s)\cos[\Psi(s) + \phi] + \sin[\Psi(s) + \phi]\Big) \,, \end{aligned} \tag{2.2}$$

where ϵ is called emittance, $\beta(s)$ is the beta function, $\alpha(s) = -\beta'(s)/2$, $\Psi(s)$ is the phase and ϕ is an initial phase. The beta function is also called amplitude function and together with the emittance gives the position dependent amplitude of the oscillation $A(s) = \sqrt{\epsilon\beta(s)}$. The function x' for relativistic particles has the meaning of an angle relative to the reference trajectory. Equations (2.2) describe a characteristic oscillation in the horizontal plane called betatron oscillation. The solution for the vertical position is analogous to that of the horizontal with new functions $\beta(s)$, $\Psi(s)$ and values ϵ, ϕ.

The phase $\Psi(s)$ advances according to [2]

$$\Phi(s) = \int_0^s \frac{\mathrm{d}\sigma}{\beta(\sigma)} \tag{2.3}$$

and the beta function $\beta(s)$ represents the effect of the focusing $k(s)$ over the transverse motion. Therefore the phase and $\beta(s)$ are parameters of the accelerator. The number of

betatron oscillations completed in a single turn around a circular accelerator is called the tune Q of the accelerator. It is defined as [2]

$$Q = \frac{\Delta\Psi}{2\pi} = \oint \frac{\mathrm{d}s}{\beta(s)} \ , \tag{2.4}$$

where $\Delta\Psi$ is the phase advance in one turn.

In synchrotrons, the curvature of the trajectory of a relativistic particle with charge q and momentum p is [2]

$$R = \frac{p}{qB} \tag{2.5}$$

and it becomes clear that particles with different momentum and therefore energy have different radius of the curvature R of their trajectory if the bending field B is kept constant. This effect is called dispersion. The linear equations of motion with this dependence are [2]

$$\begin{aligned} x''(s) + \left(\frac{1}{R^2(s)} - k(s)\right) x(s) &= \frac{1}{R(s)} \frac{\Delta p}{p} \\ y(s) + k(s) y(s) &= 0 \ , \end{aligned} \tag{2.6}$$

where $\Delta p/p$ is the relative difference in momentum to the nominal. The bending radius $R(s)$ is also a function of s to acknowledge the fact that not the entire orbit must be bent, for instance in a straight section and the bending is assumed to be entirely in the horizontal plane. The solution is [2]

$$x(s) = \sqrt{\epsilon}\sqrt{\beta(s)}\cos[\Psi(s) + \phi] + D(s)\frac{\Delta p}{p} \ , \tag{2.7}$$

where $D(s)$ is the dispersion function.

2.3.2 Fields

The force acting on a charged particle in electric and magnetic fields is described by the Lorenz formula [4]

$$\vec{F} = q(\vec{E} + \vec{v} \times \vec{B}) \ , \tag{2.8}$$

where $\vec{F}$ is the force vector, q is the charge of the particle, $\vec{E}$ is the intensity of the electric field, $\vec{v}$ is the velocity of the particle and $\vec{B}$ is the magnetic induction.

Generally, in high energy accelerators a particle is accelerated with electric fields and its trajectory is guided with magnetic fields. The reason behind not using electric fields to steer is a practical limitation on the maximum achievable intensity of the electric field without causing harmful electric discharges. Indeed, creating magnetic fields that are strong enough to curve the trajectory of a relativistic particle is significantly easier than creating the equivalent electric fields. On the other hand, the magnetic fields can not change the energy of the particle, because the force from them is perpendicular to the velocity as presented in (2.8). Therefore the roles of the electric and magnetic components of the fields acting on a charged particle in accelerators are usually separated.

The magnetic field can be expanded into Taylor series. The first few terms of the Taylor series of the vertical component of the field parallel to the y-axis as function of the position along the horizontal x-axis are [2]

$$B_y(x) = B_{y0} + x\frac{\mathrm{d}B_y}{\mathrm{d}x} + \frac{1}{2!}x^2\frac{\mathrm{d}^2 B_y}{\mathrm{d}x^2} + \dots \tag{2.9}$$

where x = 0 corresponds to the horizontal position of the nominal trajectory. The expansion is analogous for the horizontal component of the field over the vertical axis with y = 0 defined at the nominal trajectory. When we multiply (2.9) with the particle's charge over its momentum we obtain [2]

$$\frac{q}{p}B_y(x) = \frac{1}{R} + kx + \frac{1}{2}mx^2 + \dots \tag{2.10}$$

where $1/R$ is the bending radius from the dipole field, k is the focusing strength of the quadrupole field and m is the sextupole field strength. These values are momentum invariant.

The first two terms are used to describe the linear beam optics. Dipole fields are either used as bending fields or to steer the beam relative to the nominal trajectory. The quadrupole fields steer the beam with linear dependence on the position of the latter. Essentially they act just as a lens in light optics. The difference is that a quadrupole field that is focusing in the horizontal plane, is a defocusing field in the vertical plane and vice versa. Therefore to focus the beam of particles in both planes a pair of quadrupole magnets focusing in alternative directions are required.

Higher order magnetic fields introduce non-linear effects in the particle dynamics. Until this part of the report attention was given only to the linear magnetic fields of dipoles and quadrupoles and the particle dynamics in such fields. The lowest order non-linear magnetic fields are the sextupole fields. It is mainly used to compensate the chromaticity that is a dispersive effect in the beam focusing. For a quadrupole field the magnitude of the focusing strength k decreases as the momentum of the relativistic particle increases. As a consequence, the tune of the ring becomes energy dependent. The chromaticity is defined as [2]

$$\xi = \frac{\Delta Q}{\Delta p/p} \, . \tag{2.11}$$

The inclusion of non-linear fields results in non-linear equations of motion that do not have analytical solution. These forces are often negligible at the reference orbit and therefore the equations (2.1) and (2.2) are valid within a small middle region. As the transverse distance of a particle increases the non-linear effects increase until it is lost. The acceptance region, where particles are not lost because of non-linear forces, is called dynamic aperture.

2.3.3 Synchrotron radiation

Synchrotron radiation is an important phenomenon in circular electron accelerators. As it is well known in electrodynamics, every charged particle radiates when it is accelerated [4]. The power emitted as electromagnetic wave from a non-relativistic particle is described by the Larmor formula [2]

$$P_s = \frac{e^2}{6\pi\epsilon_0 m_0^2 c^3}\left(\frac{\mathrm{d}\vec{p}}{\mathrm{d}t}\right)^2 \, , \tag{2.12}$$

where e is the charge of the particle, m_0 is the rest mass of the particle, $\vec{p}$ is its momentum, ϵ_0 is the electric permeability of vacuum and c is the speed of light. For heavy particles as protons and ions with large m_0 this effect is usually negligible, but electrons and other light particles can emit significant amounts of synchrotron radiation. In accelerators it is

dominantly present when the particle trajectory is curved rather than when the particle energy is increased [4].

In circular accelerators the particle trajectory is curved by a centripetal force and is therefore accelerated. The power of the synchrotron radiation of a highly relativistic particle in that case is [2]

$$P_s = \frac{e^2 c}{6\pi\epsilon_0 (m_0 c^2)^4} \frac{E^4}{R^2} \ , \tag{2.13}$$

where E is the energy of the particle and R is the bending radius. It is evident that electrons in accelerators with larger circumference and therefore larger R will lose less energy, however the dominant parameter is the energy E. For electrons with energies above 10 GeV the construction and running costs become disproportionally large, because of the required energy restoration. For example, in the MAX IV 3 GeV electron storage ring a beam current of 500 mA will lose 500 kW of power emitted as synchrotron radiation [5].

The energy lost is restored with radio frequency (RF) cavities. These cavities are resonators in which an electromagnetic wave oscillates. The electric field is co-linear with the nominal trajectory of a particle and depending on the phase of the oscillation relative to the particle passage it can accelerate or decelerate the particle. Because of this fact, the accelerated beam is structured by the electric field as bunches of particles that are synchronized with the period of the electric field.

2.3.4 Dipole field perturbations

The orbit distortions from magnetic field errors are corrected with special dipole magnets called steering coils or corrector magnets. The corrector, in short from corrector magnet, can steer the beam to a new angle relative to the reference trajectory. Because of the short length, the steering can be approximated as a kick with strength k_p at point p. The kick changes the direction of the beam from $x'(s_p)$ to $x'(s_p) + k_p$, with $x'(s)$ defined in (2.2). The correctors can create local orbit bumps, that are a change in the transverse position of the beam within limited region of the orbit. The simplest form of the local orbit bump is the 180° bump. The formula describing this type of bump is [2]

$$k_2 = \pm\sqrt{\frac{\beta_1}{\beta_2}} k_1 \ , \tag{2.14}$$

with k as the kick strength in angle and β as the beta function. The lower indices indicate the values as position s_n, where n is 1 at the first corrector and 2 at the second. This type of bump is only possible if the phase change between the correctors is an integer times π. For odd integers the sign is $+$ and for even it is $-$.

Errors in the magnetic fields can result in a dipole kick [2]. If we assume that k_1 is the strength of a kick from an error and there is a corrector with $n\pi$ phase difference, the error can be corrected with a kick k_2 from the corrector calculated from (2.14).

To correct a kick from erroneous fields with arbitrary phase change at least two correctors are needed. Again the theory for local orbit bumps can be used. A local orbit bump without limitation to specific phase difference between the steering coils can be

created with three kicks. The relations between the kicks are [2]

$$\begin{aligned} k_2 &= \sqrt{\frac{\beta_1}{\beta_2}}\frac{\sin(\Psi_3-\Psi_1)}{\sin(\Psi_3-\Psi_2)}k_1 \\ k_3 &= \sqrt{\frac{\beta_1}{\beta_3}}\left(\frac{\sin(\Psi_3-\Psi_1)}{\tan(\Psi_3-\Psi_2)}-\cos(\Psi_3-\Psi_1)\right)k_1 \end{aligned} \tag{2.15}$$

with lower indices again indicating the position along the orbit. Analogously to the 180° bump case, when the first kick is a result from magnetic field error, the relations in (2.15) give the required strength of the two correctors following the kick to return the beam back to the reference orbit.

In all discussed forms of orbit bumps, a required kick can be outside of the technical limits of a corrector. Selecting another corrector can solve this problem as the strength depends on the local values of the beta functions, but it is possible that this is not achievable with any corrector in the machine. If a single correction is not enough to return the beam at the reference orbit, subsequent corrections with different sets of correctors can be applied until the distortion is eliminated.

There is also another reason to use many correctors to correct the kicks from magnetic field errors. The orbit distortion is usually also a superposition of many errors with different magnitudes thorough the accelerator. A single correction can be immediately followed by a kick from an error and thus introducing the need for another correction. Therefore all correctors in an actual accelerator are involved in correcting the trajectory of the beam.

There are specialized methods, in example the effective corrector and inverse response matrix, described in more detail in [2]. These methods require a stable beam and precise beam position measurements to function. During beam threading there is no stored beam and therefore they can not be used.

2.3.5 Beam position monitors

Position sensors, referred as beam position monitors (BPMs), measure the transverse beam position [2]. There are different types of BPMs. Two types using similar principle are the capacitive button BPM and the strip line BPM, presented in Figure 2.1.

The capacitive button BPM has four metal buttons isolated from the wall of the accelerator chamber that serve as electrical pickups. Two variants of button BPMs are shown in Figure 2.1. The capacitive buttons of a BPM detect electric fields from passing beam bunches. The intensity I of the signal of a button is dependent on many factors, such as the charge in the beam and most importantly the distance between the beam and the button. Based on the last, the signals from the four buttons are used to determine the position of the beam relative to the symmetry axis of the BPM with equations [2]

$$\begin{aligned} \Delta x &= a_x\frac{(I_2+I_3)-(I_1+I_4)}{\sum_i I_i} \\ \Delta y &= a_x\frac{(I_1+I_2)-(I_3+I_4)}{\sum_i I_i}\ , \end{aligned} \tag{2.16}$$

where Δx and Δy are respectively the horizontal and vertical position of the beam, $a_{x,y}$ are sensitivity factors and I_i is the intensity of the signal at the ith button with

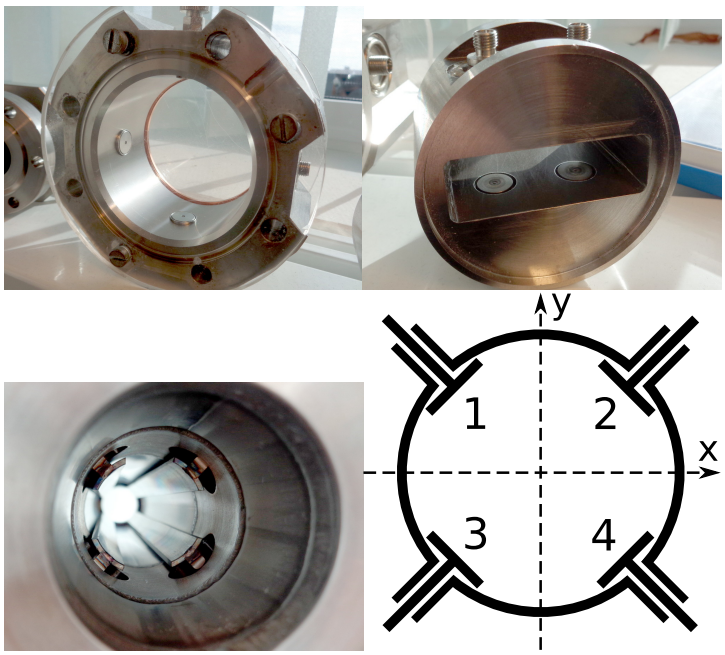

Figure 2.1: Photographs of different kinds of BPMs and an schematic. Top-left: a MAX II capacitive button BPM. Top-right: a MAX III capacitive button BPM. Bottom-left: a strip-line BPM of the MAX IV linac. Bottom-right: An a BPM schematic illustrating the position of the pickups. Pictures by author.

numbering as in Figure 2.1 bottom right. The center of a BPM is aligned to coincide with the transverse position of the reference orbit, so the results of equations (2.16) are the transverse position of the beam relative to the nominal trajectory. Strip line BPMs use the same concepts as capacitive button BPMs to measure the position of the particle beam, but generally provide more intense signal in similar conditions.

2.4 MAX IV facility

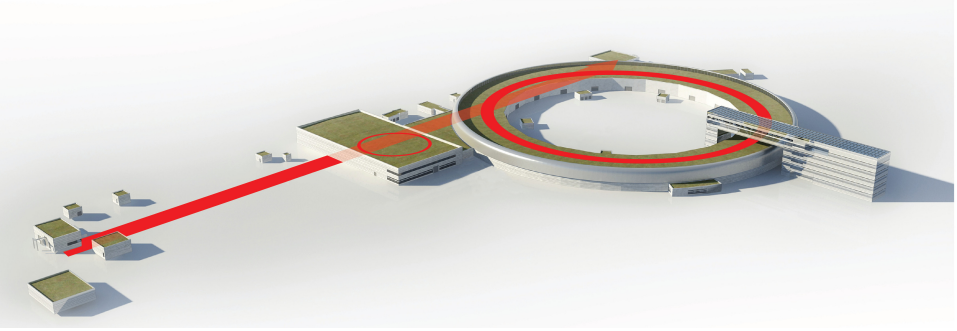

Figure 2.2: Illustration of the MAX IV facility. The electron beam is created at the far left and is accelerated by the linac (straight line). The 1.5 GeV (small circle) and 3 GeV (large circle) storage rings and their buildings are visible. Picture by Johny Kvistholm.

The MAX IV synchrotron light sources consist of two electron storage rings and a linac that also serves as a full-energy injector for the rings [3, 5], as shown in Figure

2.2. The main purpose of the facility is to serve as a synchrotron light source for user experiments. The MAX IV 3 GeV storage ring has achromats allowing lower emittance within the size and energy constraints than previous accelerators [5]. Some characteristics of the MAX IV accelerators are presented in Table 2.1 and are discussed in the following text.

Table 2.1: Parameters of the MAX IV storage rings [3, 5].

Parameter	3 GeV ring	l.5 GeV ring
Circumference [m]	528	96
Number of achromats	20	12
Horizontal tune Q_x	42.2	11.22
Vertical tune Q_y	16.28	3.14
Lattice radiation losses per turn [keV]	363.8	117.2
Revolution time [ns]	1760	320
Momentum acceptance	±4.5%	±3.0%

2.4.1 MAX IV 3 GeV storage ring

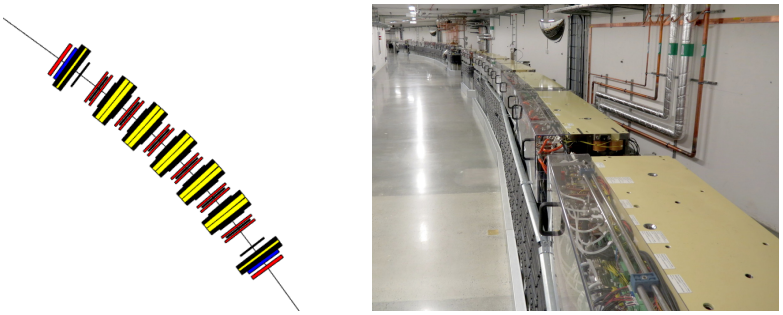

Figure 2.3: Left: Diagram of a MAX IV 3 GeV ring achromat. The yellow rectangles represent parts of the dipole magnets, the red and blue are quadrupole magnets. The long line between the achromats is a straight section, where a transfer line and insertion devices for synchrotron light production can be placed. Right: Photograph of a section of the MAX IV 3 GeV ring. The iron blocks of the cells are on top of concrete support. Pictures by author.

The MAX IV 3 GeV storage ring has a distinct approach in its design that has proved to be a successful concept for synchrotron light sources [5, 6, 7]. This new concept is the use of multi-bend achromat instead of the widely used double-bend achromat. As the name suggests, the double-bend achromat has two bending magnets and magnetic optics in its structure [2]. The multi-bend achromat on the other hand has multiple bending magnets, there are seven bending magnets in the MAX IV 3 GeV ring [8].

Seven cells of optics, one for each bending magnet within the achromat, are manufactured from pairs of iron blocks. These contain the iron yokes of the magnetic components in the cell. They are the yellow painted blocks in Figure 2.3. The two blocks of a pair form the top and bottom half of a cell and limit the misalignment between the components of the cell, as they are part of the same pair of blocks [3, 5].

There are a number of components for orbit correction in each achromat. Each achromat is equpped with 10 circular aperture capacitive button BPMs. The peak sensitivity of the BPMs is at 500 MHz that is the fifth harmonic of the RF. There are 10 horizontal and 9 vertical corrector coils in each achromat with maximum kick strength of 0.38 mrad by design [9]. In total there are 200 BPMs, 200 horizontal correctors and 180 vertical correctors. The large number of the BPMs are required to allows good sampling the betatron oscillations in both planes [5, 10].

The electrons are injected at full 3 GeV energy from the linac through a transfer line and with miltipole injection kicker. The transfer line is an optical structure of magnetic components that connects the ring with the linac. There are BPMs and corrector coils for both directions inside the transfer line. At the end of the transfer line there is a vertically deflecting magnet that bends the injected electrons, but shields the already present beam from that bending field [2]. The pulsed multipole injection kicker allows new electrons to be stacked without significant disturbances of the stored beam, process referred as top-up [11].

2.4.2 MAX IV 1.5 GeV storage ring

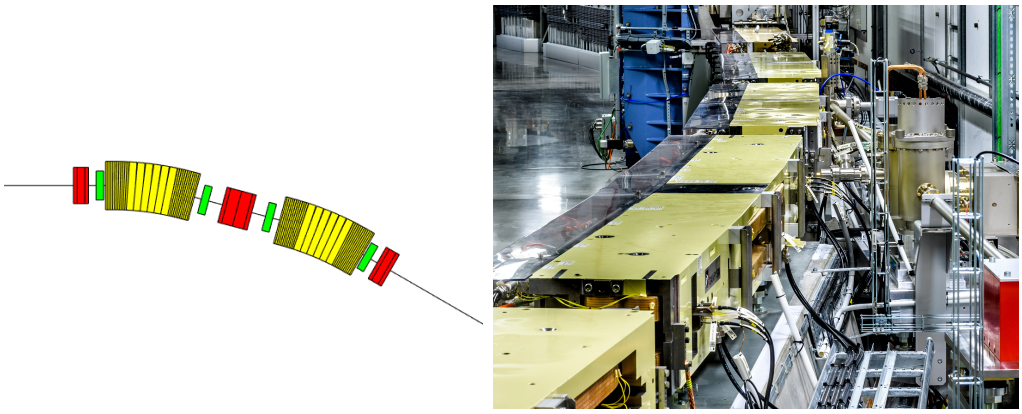

Figure 2.4: Left: Diagram of a MAX IV 1.5 GeV ring achromat. The yellow rectangles represent parts of the dipole magnets, the green are sextupole magnets and the red are combined function quadrupole and sextupole. Right: Photograph of a section of the MAX IV 1.5 GeV ring. Photograph by Kennet Ruona.

The MAX IV 1.5 GeV storage ring uses the established double bend achromat lattice. Each achromat is made out of two nearly identical optical structures with a single bending magnet that are mirrored with respect to the achromat center. Each cell is machined from a pair of iron blocks, similarly to the 3 GeV ring [3].

The betatron oscillations are sampled with 36 BPMs in total, 3 in each achromat. The BPMs are of capacitive button type with rhombus aperture [12]. Right after each BPM there is a combined function sextupole with extra corrector coils for both directions. The maximum kick a corrector coil can give to the beam is 0.25 mrad [3].

The injection of electrons in the storage ring is accomplished with a single dipole injection kicker [13]. The transfer line to the ring starts at the extraction point in the linac, where the incoming electrons have 1.5 GeV energy. Just as the transfer line of the large ring it has focusing magnets, diagnostics, corrector magnets and ends with a septum magnet in the ring.

2.4.3 MAX IV linac

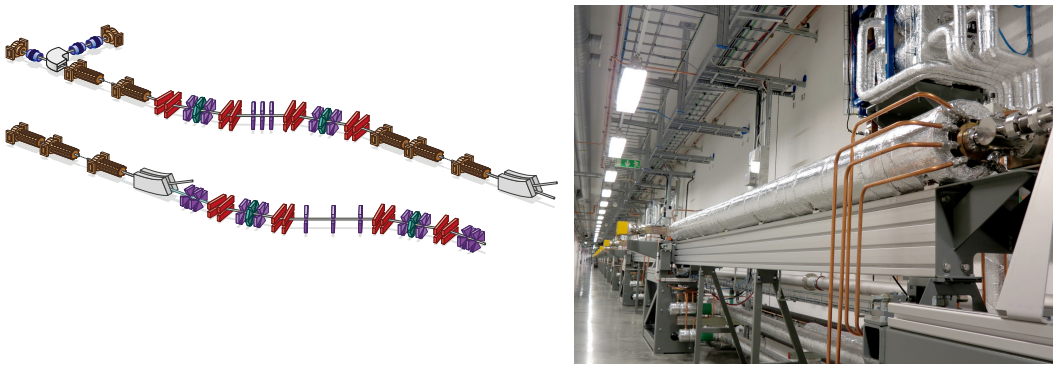

Figure 2.5: Left: Illustration of the linac with some of its components. On the top are the components until the small ring injection point and the rest of the linac is cut and shown below. The thermionic pre-injector is in the bend arm in the top left side. Picture by Johnny Kvistholm. Right: One of the 39 accelerating structures of the linac. Each structure consists of several 3 GHz RF cavities with accelerating gradient of 20 MV/m. Picture by author.

The MAX IV linac is used to accelerate the electrons up to the nominal energy for the corresponding storage ring. Most of its 250 m length is dedicated to accelerating structures as the ones shown in Figure 2.5. During injection to one of the rings the beam is kicked into the transfer line of the ring [3].

One of the components of the linac relevant to beam threading is the chopper used to form the bunch structure. It is part of the thermionic pre-injector [14]. In the pre-injector the electrons pass through the RF strip-line chopper before reaching an aperture. The electric field between the two strip-lines of the chopper deflects the beam. The RF signal applied to the strip-lines is chosen to allow short trains of 3 bunches of 100 pC with period of 10 ns in between to pass the aperture. The chopper can be used to form a beam with 500 MHz bunch structure. This could be very useful for beam threading, because the BPMs of the rings are most sensitive at 500 MHz.

2.4.4 Additional description of relevant components

In this section, the details of some components that are relevant to beam threading will be given.

The signal from a single button of the capacitive button BPMs in the MAX IV storage rings is filtered and then digitized by an analog to digital convertor (ADC) [15]. Each button has its separate ADC channel. The ADC takes a sample of the signal every 8.5 ns and can store up to 65000 samples in its memory buffer. The ADC channels can be synchronized with the incoming beam. The ADC buffer therefore holds the measured signal intensity at the corresponding button over time. From the values in the ADC buffer the BPM system calculates the beam position. The accuracy and reliability of the calculated position depends on the signal intensity with respect to the intensity of the background and electronics noise. The lower limit of the stored beam current, at which the BPMs can not anymore provide precise beam position measurements, is few

hundred microamperes. Also each ADC channel has to be calibrated to eliminate the systematic errors caused by misalignment or differences in the sensitivity between the buttons [10, 15].

The vacuum chamber of the storage rings defines the physical aperture of the accelerator. Practically, when the beam exits the physical aperture it hits the wall of the vacuum chamber and it is lost immediately. The physical aperture of the 3 GeV ring in general is circular with diameter of 22 mm. The 1.5 GeV ring has an ellipsoidal aperture with width of 40 mm and height of 20 mm [3].

Each storage ring has an oscilloscope. The oscilloscope has several channels, one of which is connected to the buttons of the first BPM. The signal received by the oscilloscope is the sum of the signals from the four BPM buttons. It can measure over far longer time range than the ADCs of the BPMs [3].

Chapter 3

Software and methods

3.1 The MAX IV control system software

The automated beam threader (ABT) is programmed entirely in Matlab and makes use of several software solutions available for Matlab. Matlab is used in the operation of the MAX IV accelerators and currently version 2013a of it is installed in the control room. The ABT will used in the control room and therefore it is compatible with that version. Some of the main software components in the MAX IV control system are described in the following text.

3.1.1 Tango controls

Tango Controls is a control system based on distributed objects called tango devices [16]. Each tango device is running in a Tango device server that is responsible for communication and handling of requests. A device is identified by its name and it can be a macro device with many sub devices that in turn can also be macro devices. This allows the control system to have a structure adapted to the hardware structure.

Tango Controls is widely used in the industry and science experiments and it is the control system of the MAX IV accelerators. Both beam control instrumentation and power supplies together with diagnostics and detectors are available as Tango devices. Its monitoring and logging services are integrated in the control room. A Matlab binding that allows Tango procedures to be executed from the Matlab environment is utilized.

3.1.2 Matlab middle layer

The Matlab Middle Layer (MML) is a widely used in accelerators code package that abstracts the hardware control from the Matlab environment. It has uniform function and naming conventions, both simplifying the program development and allowing exchange of software between accelerators. In MAX IV facility it is on top of and is communicating with Tango. The MML has an included simulator that is controlled in identical manner as the actual accelerator, because of the uniform function naming. The simulator uses the accelerator toolbox (AT) [17] for Matlab.

3.1.3 Accelerator toolbox

Accelerator toolbox is a collection of accelerator physics routines and applications for Matlab. It can be used to create accelerator models and to numerically study the properties of these models. Simulation and analysis of beam dynamics using tracking is also part of the AT. The tracking code is based on TRACY-2 [18].

There are several features of AT that are used to study beam threading. The six-dimensional particle tracking takes into account synchrotron radiation losses and compensation with RF cavity. As later will be discussed, the major part of the beam threading process is done without restoring the lost particle energy due to synchrotron light. The threading procedures are in part needed because of misalignment of components and AT has functions for applying specified misalignment and rotation to components relative to their design position.

3.2 Optimization introduction and methods

The optimization algorithms try to find the global minimum of a function as their goal. The minimized function is called objective function. The value of the objective function at given values of its variables is called an objective value. There are numerous optimization methods and algorithms. Three of them were tested in simulation for the automated beam threader (ABT). Their principles are explained below.

3.2.1 Simplex

The ABT optimizer algorithm uses the Nelder-Mead simplex method [19] to find a minimum of a multivariable objective function. It is a derivative free method. In N-dimensional space the simplex is a $N+1$ point shape that encloses a non-zero volume of that space. If each variable of the objective function corresponds to a single dimension, then the coordinates of each point of the simplex are an input for the objective function. An objective value can then be assigned to each point of the simplex.

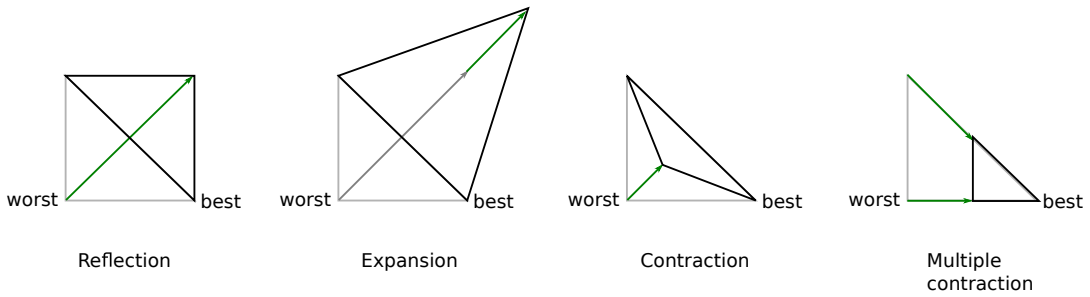

Figure 3.1: The simplex operations in 2D space. The initial simplex is the gray triangle and the final simplex is the black triangle. The green arrows show the movement of the points. The worst and best points indicated are for the initial simplex.

An initial simplex can be constructed from a single initial point by adding for each basis direction a new point that is offset in that direction. There are several operations performed in the Nelder-Mead simplex method. These are illustrated in Figure 3.1.

Once the initial simplex is formed the point with worst objective value is picked. First is the reflection. The worst point is reflected in the $N-1$ dimensional plane defined by

the N other points. If the reflected point is the best point, a further expansion in that direction is attempted. Then the best of the reflected point or the expanded point is the new point.

If the reflected point is still the worst point, a smaller step called contraction is attempted. If at least one of these two points is better than the previous worst, it is the new point. Otherwise the whole simplex is shrank with a multiple contraction towards the best point.

After a new simplex is formed by an operation, the worst point of this simplex is selected and the procedure is repeated again. In terms of the optimizer algorithm each finished operation in an iteration and each new point attempted is an objective function evaluation. The two most computation heavy iterations are the initial simplex formation with $N + 1$ evaluations and the multiple contraction with $N + 2$ evaluations.

The simplex algorithm is available in the Matlab optimization toolbox.

3.2.2 Particle swarm optimization

The particle swarm optimization (PSO) [20, 21] is a robust algorithm that is able to search for the global minimum of a multivariable function. It accomplishes that by simulating the motion of particle like agents in multidimensional space.

The goodness of any position in that space is defined by the objective value. The personal best position pb_i of the ith agent is recorded. The global best position gb that is the best of all personal best positions, is also recorded. Then the equations of motion for the ith agent are [21]

$$\begin{aligned} v_i &= w * v_i + c_1 r_1 (pb_i - x_i) + c_2 r_2 (gb - x_i) \\ x_i &= x_i + v_i \end{aligned} \tag{3.1}$$

with x_i being the position, v_i the velocity, r_1 and r_2 are random numbers in the interval $[0, 1]$, c_1 and c_2 are parameters and w is an inertia parameter. At every iteration the new velocity is calculated and the agent is moved. Eventually, most of the agents shall gather within a small distance away from the lowest minimum.

3.2.3 Robust conjugate directional search

The robust conjugate directional search (RCDS) [22] is a robust direct search algorithm using Powell's method. To explain the algorithm, conjugate directions shall be introduced first. The Hessian of a multivariable function f is defined as [22]

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} \ . \tag{3.2}$$

If the relation $\vec{u}^T \cdot \hat{H} \cdot \vec{v} = 0$ is true for two vectors u and v in the variable space, then they are conjugate [22]. Powell's method produces a set of mutually conjugate directions for any point in the variable space [23].

The algorithm optimizes with a robust line optimizer along a single direction. Initial set of points is sampled. A parabola is continuously fit to the points as points outside the noise limits are discarded. The best point is chosen from the minimum of the final fit. An optimization along single conjugate direction is independent from that of any other conjugate direction. Thus, every new direction improves upon the results from the previous.

3.3 Beam threading at MAX IV

In the MAX IV synchrotrons the beam threading is often performed after long period shutdowns or hardware interventions and including the first accelerator commissioning. After a long period shutdown the state of the synchrotron is changed compared to the one it had last during operation. Temperature changes and other conditions influence the lattice and its performance.

The automated procedure developed in this project is presented in section 3.5.

3.3.1 Manual procedure

This section describes the manual beam threading procedure and mentions previous attempt for automation at the end. The steps were recreated in a demonstration for this project. The power to the RF cavities is turned off during that process

Firstly, the transfer line to the storage ring is threaded if needed and the injection kicker angle is set. The procedure of beam threading through the transfer lines is identical to that of beam threading in the rings for a first turn.

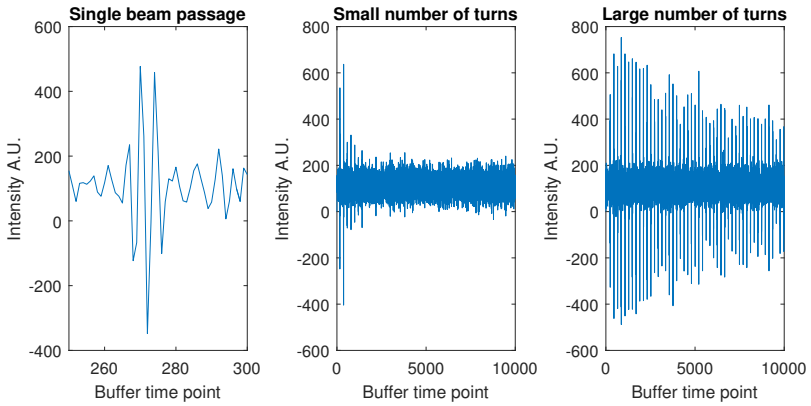

Figure 3.2: BPM ADC buffer samples from the 3 GeV storage ring. The 0 time point is synchronized with the beam entering in the ring. Left: a ringing from a single beam passage. Middle: ringing peaks from a small number of beam pasages, the dense peaks between 0 and 200 intensity are noise. The signal intensity is about the same order of magnitude. Right: ringing peaks from large number of beam passages.

The beam threading of the MAX IV synchrotrons to achieve the first complete turn around the orbit is performed mostly with correctors and BPMs. Because the position of the beam can not be determined, the ADC buffer of one channel of a single BPM is viewed directly in Tango. The steps performed during the threading are presented in Figure 3.3.

An example buffer readout is plotted in Figure 3.2, left. The beam of electrons creates a short ringing signal in the BPM button when it passes. This signal is used to determine the presence of the beam and also the estimated location where the beam is lost. An ADC buffer of a BPM after and near the loss location is viewed during the threading process. Correctors before and close to the loss location are tried first.

The described steps are repeated until the beam travels a complete turn around the accelerator. Then the multi-turn threading is performed that is essentially the same. If instead all correctors have been tried to find a good kick, but the beam does not proceed

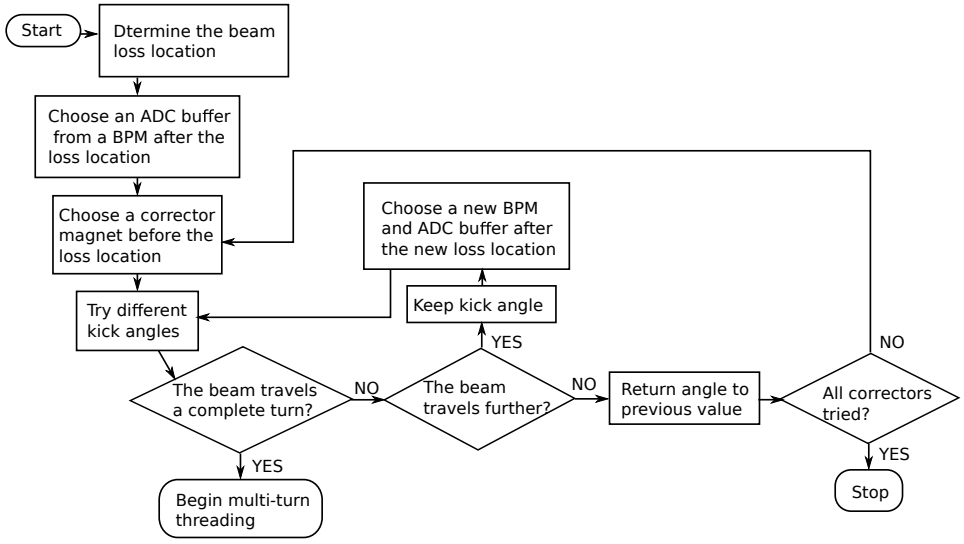

Figure 3.3: The single-turn manual beam threading procedure.

further in the accelerator, special actions must be taken. These may include, but are not limited to changing the strength of quadrupole magnets, changing the injection kicker angle, checking the state of the accelerator components with Tango or stopping the synchrotron and investigating the hardware.

The multi-turn threading has the following steps, presented in Figure 3.4.

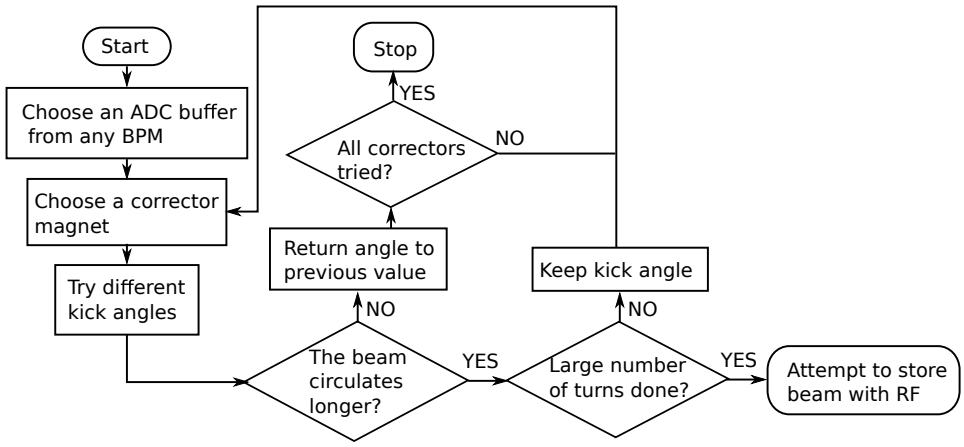

Figure 3.4: The multi-turn manual beam threading procedure.

After an orbit is already established, a kick in single corrector will have effect on all subsequent turns. The number of completed turns by the beam that is the metric being optimized, is estimated from the number of peaks in the ADC buffer from each beam pass, shown in Figure 3.2, middle and right.

Storing a beam current is attempted after a sufficiently high number of turns is achieved. The RF cavities are powered and their phases are adjusted. The injection kicker strength may also be changed at this stage. An orbit correction is performed with an amount of beam current that allows beam position measurements with the BPMs. If this is not the case, then special actions like the ones listed for first turn threading are performed.

A previous and unsuccessful attempt to automate the beam threading was made that

predated the automated beam threader that is subject of this report. It is programmed on Matlab and uses the Optimization toolbox and MML. This procedure was briefly tested at the 3 GeV ring and it did not thread the beam through the synchrotron [10].

3.3.2 Current issues with manual beam threading

There are known issues of the described manual beam threading. The first and main issue is the time length of the procedure [10]. The fastest manual threading took two hours and was achieved after the summer shutdown in 2017 on the 3 GeV storage ring including the transfer line. On some other occasions the threading continued over several days. It is generally correct to assume that a manual beam threading can take a day [10].

Notable issues are the costs. The run time and personnel are costly. It is a priority to use the expenses in efficient manner to produce more scientific results. Therefore the mean time to recover by beam threading must be minimized [10].

The manual beam threading is monotonous and tiresome. The person performing the beam threading has to repeat the same actions numerous times and must have high concentration. The exhaustion will slow down the threading and therefore it can not be done for prolonged periods of time. In addition, the slow progress or the absence of such is discouraging. This makes it more likely that the person will lose concentration. The outlined problems can increase the time needed to thread the beam [10].

3.4 Simulation model

The simulations were performed in Matlab using the MML and AT. The design lattice of either of the MAX IV storage rings in AT is used as a base. The physical aperture of the vacuum chamber is added and the RF cavity power is turned off. Misalignment to some components in the lattice are introduced as errors.

The simulations were mostly performed with the small ring model, because there was more run time available for threading experiments on the 1.5 GeV machine [10]. For the purpose of beam threading misalignment errors are introduced. In the design of the ring a Gaussian distribution with cutoff at 2σ is used. The transverse shift of the components is $\sigma = 100$ μm and the rotation is $\sigma = 100$ μrad [3].

The error model in the simulations is different. It uses the Irwin-Hall distribution [24] of two uniform random variables. The error distributions are centered at 0 and are rescaled using the design values. A random number is drawn from a distribution and is used as shift or rotation error accordingly.

The misalignment errors for the small ring model are applied to combined quadrupole and sextupole magnets. The error distribution is scaled until the number of turns the reference particle can make is between 10 and 100. Such a starting point promises that an improvement is possible and significant work is required by the threader. In practice Irwin-Hall distribution with limits at $\pm 5\sigma$ provides lattices suitable for testing of beam threading.

In some simulations an artificial noise was added to the number of completed turns evaluated with the tracking. The noise has two components both having an Irwin-Hall distribution of two uniform random variables. The first component is independent of the turn count and is within ± 200. It attempts to model a background noise. The second

component models an uncertainty in the counting of the number of turns N and has amplitude of $5\sqrt{N}$.

Finally some simulations include a 1 % chance that the linac will not deliver a beam to the ring. In that case the assumed number of turns is 0 without noise.

3.5 Automated beam threading for MAX IV

This section describes the features and explains the main steps of the algorithms that were developed for the automated beam threader. The results are presented in the next chapter.

3.5.1 Development notes

The development of the automated beam threader (ABT) made heavy use of simulation to minimize test time on the machine. The fundamental parts of the ABT were developed and tested in simulation. Starting without prior information made the development time consuming and often relied on trial and error techniques. Significant part of that stage was done on simulation. An already functioning prototype of the ABT was later tested and delivered results in the experiment quickly relative to the total development time.

The variety of diagnostics useful for the purpose of ABT is rather limited. The main sensors are the ADC channels of the BPMs of the storage ring. Their large number is an advantage as the random fluctuations in the optimized metric are decreased by averaging. The readings from the oscilloscope connected to the first BPM are another source of data applicable to measuring the performance of the ABT.

As explained earlier in this chapter the first turn and multi-turn beam threading at MAX IV have different methods. Therefore the ABT was also split into a single-turn and multi-turn beam threading routines. It was advisable, because the beam does not travel through the whole accelerator during the first turn threading. The multi-turn ABT was developed first as it was assumed to be quicker to complete.

3.5.2 Multi-turn Automated Beam Threader

The multi-turn automated beam threader (MABT) searches for the corrector magnet settings that allow optimal number of turns. It is used to thread the ring after the beam circulates for some turns in the synchrotron and until there are enough turns to try storing a current with RF cavity on.

The threader code is modular, because it is split into several internal subroutines and has few external functions that are specific to the accelerator. An example of the latter are the sensor functions that are briefly introduced later in this section. More details on the ABT code are provided in Appendix B and Appendix C.

Main Algorithm

The actions performed by the final version of the MABT and their order are presented in Figure 3.5. The threader starts by initializing global variables. It matches the machine it is running for and writes the specific sensor function and the number of corrector subsets in global variables.

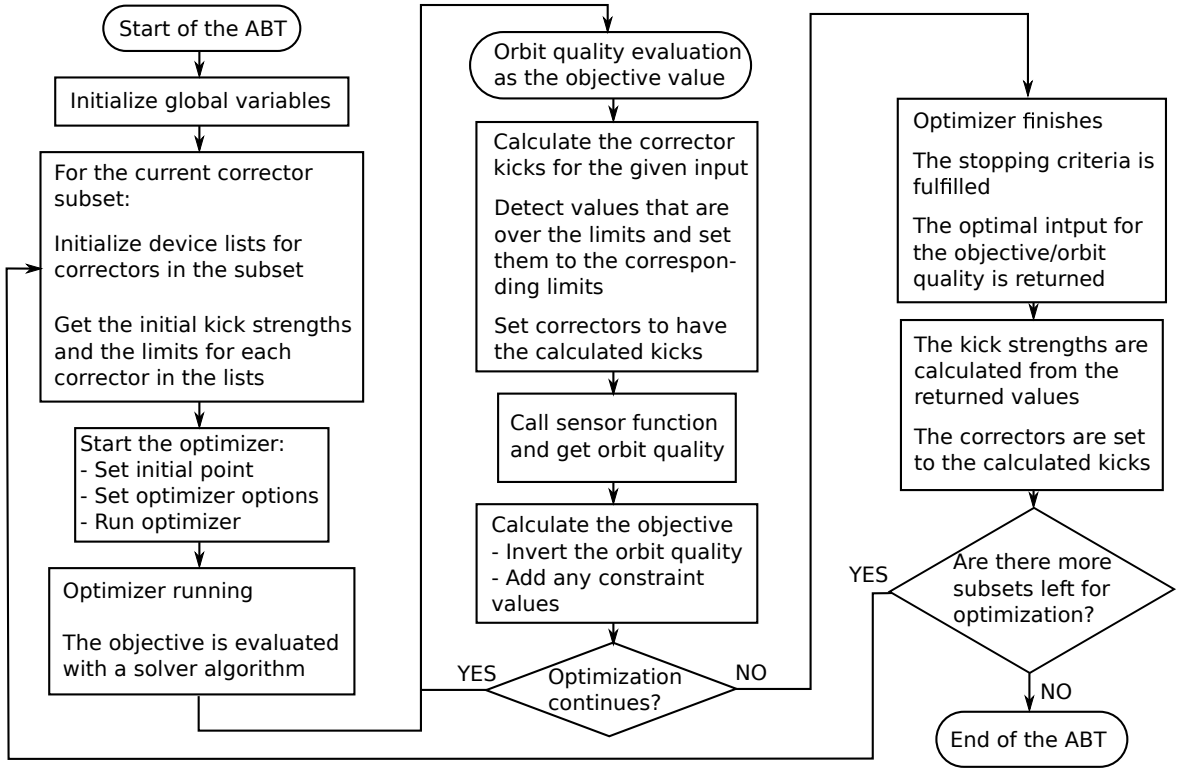

Figure 3.5: The multi-turn automated beam threader algorithm presented by a flowchart.

The corrector subsets are lists of unique corrector devices used to split the optimization problem of the whole synchrotron into smaller units. They consist of a similar number of corrector coils that are rather equally spaced throughout the ring. The use of subsets instead of all correctors at once decreases the run time of the MABT and allows execution points where the optimization can be securely paused or stopped. The former advantage is true for the Simplex algorithm and this is explained in the next chapter.

The main loop of the MABT prepares the required variables and uses the black box optimizer to obtain the best kick strengths for the current corrector subset. The corrector subset is identified with an integer number s, where $1 \leq s \leq total\ number\ of\ subsets$. The corrector subset and its current settings are loaded using the MML. Then the options for the optimizer algorithm including specific stopping criteria are set and optimization is started. While the optimizer is running it calls multiple times the objective function also referred here as the orbit quality function.

Skipping to the point when the optimizer finishes, the control is returned to the MABT algorithm. The optimal kick strengths that are obtained from the optimizer result, are set to the correctors in the current subset. The main loop continues repeating these steps until all subsets are optimized.

Orbit Quality Function

The task of this function is to give consistent quality value based on measurements of the electron beam. The optimizer algorithm calls this function to get objective values for different values of the variable list that it optimizes. The corrector strengths are calculated from the values of these variables, each variable corresponding to a single corrector. These strengths are checked against their limits and then the correctors are

set to them.

The objective value is calculated from the quality value returned by the sensor function. This function returns a positive value corresponding to the estimated number of completed turns by the beam by using given sensors. The sign of the quality value is changed, because by convention optimizers search for the minimum of the objective value. If any of the variables from the optimizer results in a kick angle outside the limits, a soft constraint value is calculated that is proportional to the fourth power of the maximum normalized difference over the limit for any angle.

The objective value is the sum of the always negative sign-inverted quality value and the always positive soft constraint. The constraint has minimum of 0 at the origin of the multi-variable space of corrector kick strengths and rapidly grows outside the latter. With such soft constraint the optimizer algorithm will move towards the same origin and within the corrector limits. Inside the limits the constraint value is 0 to avoid undesired effect on the optimization.

3.5.3 Single-turn Automated Beam Threader

The single-turn automated beam threader (SABT) can be used to guide the beam through a synchrotron. It requires that the transfer line is threaded in advance and the beam enters an achromat. It is based on the final MABT algorithm. The introduced changes handle the differences between the methods that were discussed before. The SABT algorithm is presented in Figure 3.6.

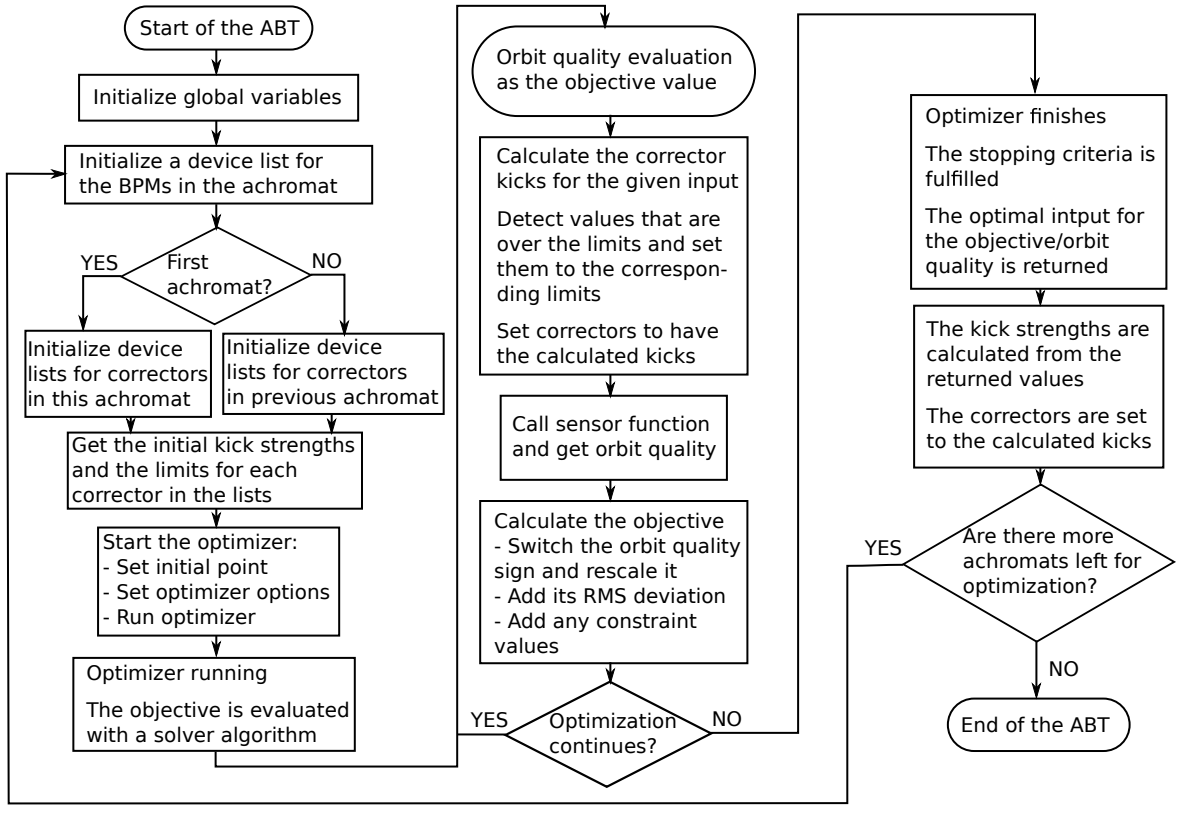

Figure 3.6: The single-turn automated beam threader algorithm presented by a flowchart.

In principle the SABT can be used in place of the MABT. There is nothing that prevents the SABT from functioning after the first turn is completed. It may even

increase the number of completed turns by optimizing more achromats, such event was observed during the experiments presented in the next chapter. However, the SABT is limited to a single achromat. Using correctors distributed throughout the ring is usually much more time efficient in reducing the orbit distortion caused by magnetic components. This was also observed during the experiments.

Main Algorithm

Similarly to the MABT, the SABT starts by initializing variables. The synchrotron is detected and its corresponding sensor function is set. The synchrotron is divided into achromats instead of being divided into corrector subsets. By convention the first achromat is after the transfer line and the beam reaches it first.

The only detectors that are both providing information regarding the threading and are positioned along the whole ring are the BPMs. Because of this, the SABT works exclusively with BPMs.

The main loop starts with initializing a list of BPM devices that are in the current achromat. Then it sets a list for the corrector coils that are in the previous achromat with the exception of the first achromat. This approach was chosen, because by only using BPMs in achromat A it is not known if the beam reaches achromat $A + 1$ and it is not possible to optimize the latter without beam. When the beam is lost after the last BPM inside achromat A, it will not reach achromat $A + 1$ and this can not be detected in achromat A.

After the lists of devices for the current achromat are initialized, the algorithm continues as it does in the MABT: it sets the options for the optimizer algorithm and runs it. With the obtained optimal values it sets the corrector strengths and continues until all achromats are optimized.

Orbit quality function

The orbit quality function introduces a new evaluation of the objective value for the SABT. The quality value returned by the sensor function is multiplied by -10 and its standard deviation that is also returned by the sensor function, is added. This is done to artificially create small differences between the objective values for different points in the multi-variable space, where the optimizer searches for a minimum. If all points have the same objective value the optimizer algorithm may get stuck or finish immediately. After this a constraint value is added in the same manner as in the MABT.

3.6 Additional automated beam threader actions

The automated beam threader algorithm depends on sufficiently accurate objective value. The first step to obtain a reliable and consistent quality value is to isolate the signal from the noise in the measurements. The next step is to define a procedure to reliably evaluate the quality value from the filtered signal. In this section, the solutions for these two major tasks are described.

3.6.1 Noise filtering

The noise is a significant problem. The signal obtained from BPMs is often unusable. This is especially the case for the small ring. Two signal processing methods are used in the ABT. The best signal to noise ratio was achieved with selective frequency filtering and summing of the BPM ADC buffers. The other method sums the absolute of the difference from the mean value for each ADC channel buffer of a BPM.

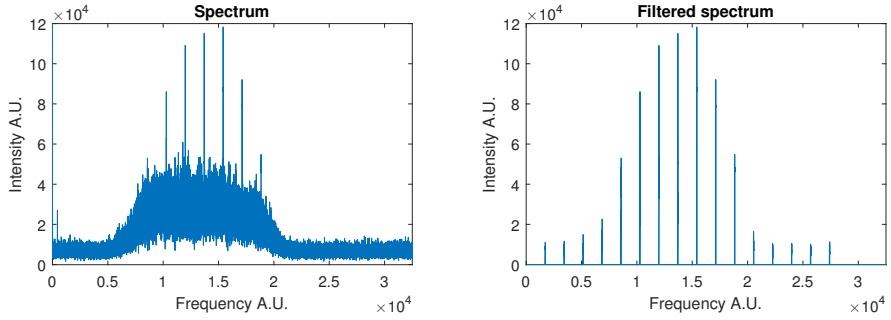

Figure 3.7: The spectrum of a sample signal from a small ring BPM with many completed turns by the beam, left. Right: the same signal after selective filtering.

The selective frequency filtering method, internally named Fourier transform peak absolute (FPA) filter, is specific to the hardware. The revolution frequency of the beam in the ring is the base harmonic in frequency space. The frequency space is obtained by fast Fourier transform presented in detail in [23], with frequency position dependent on the time size of the sample. Because the position of a given frequency in the result is dependent on the input time size, different hardware sampling rate or sample interval will change the position of the frequency.

The envelope of the peaks after a Fourier transform is transformed into the envelope of the signal harmonics [25]. By selecting only the beam signal harmonics within narrow frequency windows, most of the total noise intensity is removed, as shown in Figure 3.7. The the position of the frequency harmonics must be known precisely in advance to define the position of the windows.

Table 3.1: Summary of the empirical FPA parameters.

Source	Base harmonic	Window width
Small ring BPM	1713 points	31 points
Large ring BPM	314 points	11 points
Large ring oscilloscope	849.3 points	91 points

The filtered signal is obtained after an inverse fast Fourier transform. The absolute values of the signals from the four ADC channels are summed to get the final result. Because of the limited number of harmonics and some noise in the harmonic peaks, the filtered signal has distorted envelope. At low number of completed turns the signal peaks are smoothed. The last 10000 points of the time interval have to be ignored, because the envelope of filtered signal usually has a large distortion at the end.

The second method is internally named absolute sum (AS) filter. It does not have noticeable signal to noise ratio improvements, as is visible in Figure 3.8. It is used in single-turn threading, where the the distortion of the FPA is too destructive. It can

also be used to view the an BPM ADC channel signal to more easily assess the signal envelope by eye.

The AS filter extracts the individual peak amplitude in the ringing relative to the mean of the whole buffer. Then it sums the amplitude signals from the four buttons of a BPM and finally applies some smoothing to remove very sharp noise peaks.

3.6.2 Orbit quality evaluation

The procedure made to evaluate the quality value of the orbit is as follows: make a measurement with a sensor, filter the measured data, find the number of peaks with prominence above specific value, use the number of peaks as quality value. If there is more than one sensor, the average number of peaks is used. This procedure is performed by the sensor function that was introduced in section 3.5.2. There are several different sensor functions depending on the theading and the hardware. They are summarized in Table 3.2

Table 3.2: Threader and sensor combinations.

Threader	Sensor	Filter	Note
SABT small ring	BPMs	AS	Low sensitivity
MABT small ring	BPMs	FPA	
SABT large ring	BPMs	AS	
MABT large ring	Oscilloscope	FPA	Low sensitivity
MABT large ring	BPMs	FPA	Limited to 315 turns at most

It was already explained that each passage of the beam creates a ringing signal at the BPM button, shown in Figure 3.8 on the left. It is possible to count the number of beam passages from the ringing by eye, but it is not trivial task with computer code. The number of peaks differs between ringings and their amplitude gradually decreases.

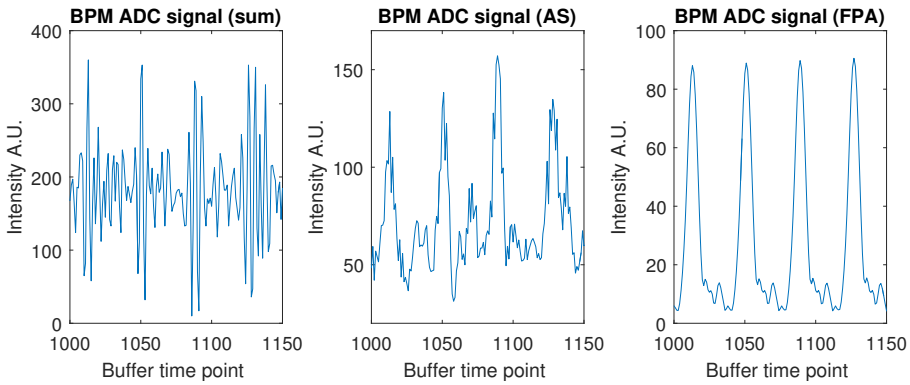

Figure 3.8: Signal of a beam passing in BPM ADC buffer from the small storage ring. The effects of the AS and FPA filters are visible.

The AS filtered signal is an approximation of the amplitude of the ringing. This follows from the previously explained details of the filter. The amplitude has a single distinct peak at each ringing, shown in Figure 3.8 middle. The number of beam passages then must be equal to the number of these peaks. Counting the number of completed turns is now simplified to peaks detection after the AS filter.

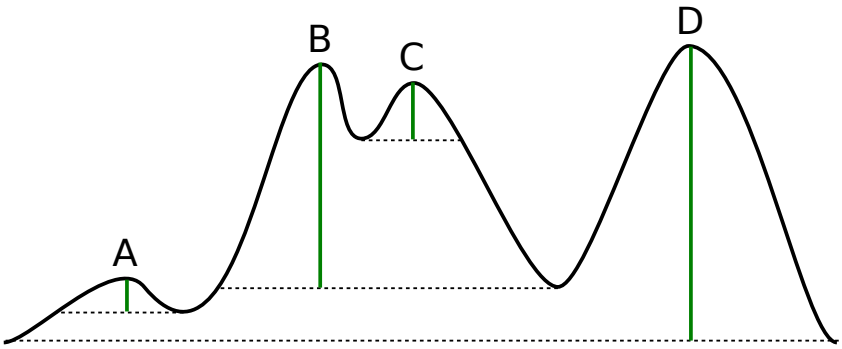

Figure 3.9: The peak prominence illustrated. The green line shows the peak prominence, the dotted line shows the peak region. The peaks are ordered by height from highest as D, B, C and A. The highest peak is the most prominent. The prominence of C is limited by B.

The peak counting must be robust. Finding peaks by prominence generally avoids noise peaks. The prominence of a peak measures its significance relative to other peaks. From the peak top in the left and right direction the lowest minimum is searched. The searching continues until a higher peak or an end of the data is encountered. The highest of the left or right minimum is selected. Then the prominence of the peak is the difference between its height and that minimum. Examples are shown in Figure 3.9. The rather small noise peaks on top of the signal peaks have low prominence.

Matlab comes with a peak detection function. Peaks can be selected by numerous criteria, one of which is the prominence. The issue is that the option to select peaks by prominence is not available in Matlab version 2013a. To overcome this issue a peak detection routine was developed that can provide identical results to the peak detection by prominence found in later versions. The disadvantage of the routine is the slow execution speed. By sacrificing little accuracy a significant speed-up was achieved in the routine.

The peak detection by prominence is still useful in FPA filtered signal. Such signal is shown in Figure 3.8 right, but it is not always the case. For instance, the oscilloscope signal still has significant noise peaks left after filtering. The FPA filtering efficiency depends on the window width and as was shown in Table 3.1, it is relatively large for the oscilloscope.

Chapter 4

Results

4.1 Simulation results

As previously stated, a major part of the development and testing of the automated beam threader (ABT) was aided by simulations. The simulation results guided several design decisions, such as the introduction of corrector subsets and the choice of optimizing algorithm.

Some terms can refer to both the simulated model and the real machine. For clarity the terms lattice and model will refer to the virtual physics model used in the simulation. The terms machine and hardware will refer to the actual hardware and associated equipment.

The first simulation results are simple tracking of a reference particle without RF cavity. Without errors the particle travels for limited number of turns, because of the synchrotron radiation losses. For the large ring the number of completed turns is 2427 before the particle is lost and it is 3334 for the small ring.

It is worth noting that the number of completed turn is significantly lower in models without non-linear fields. The large ring can only hold the particle for 924 turns. The significant difference is caused by the optimization during the design of the ring. The sextupole magnets were designed to allow large dynamic aperture and momentum acceptance [3, 5].

4.1.1 Simplex

Neither particle swarm optimization or robust conjugate directional search outperformed the simplex method with similar number of objective function evaluations in the simulation. Therefore the MABT continued to make use of the Simplex method as its optimizer.

Execution speed

The first simplex simulation results were made, before the corrector subsets were introduced in the MABT code. The simulation was without noise on the large ring model that has 380 corrector magnets. The initial simplex formation requires 381 particle trackings. This can take half an hour on ordinary computer, before any optimization with other operations is performed.

On top of that, the algorithm operates on the worst point first. A lot of time can be spent operating on the same points in a bad region and gradually improving the simplex

until a new best point and therefore a new result is obtained. Even worse, if the operation is a multiple contraction, another 382 evaluation are performed.

The chosen solution is to split the correctors into subsets as described in the previous chapter. With a 19 corrector subset substantial improvements were achieved, such as starting with only 7 turns and finishing with 686 turns in 86 iterations and 150 evaluations. The position error of the components was within $\pm 2\sigma$.

The simulation studies continued on the small ring model with $\pm 5\sigma$ misalignment. The corrector coils were divided into 6 subsets of 12 correctors each. Using all subsets and starting from 16 turns, the MABT was able to finish at 707 turns after 120 iterations and 285 evaluations. It must be noted that the particle tracking and the simplex method algorithm are both deterministic and give the same result for the same initial conditions in absence of random noise.

Performance with noise

The performance of the optimizer with random noise using few subsets was investigated. Using two corrector subsets and starting at 17 turns, without noise 1409 turns were achieved, but with $\pm 5\sqrt{N}$ signal noise model 1438 turns were reached in one instance. This observation was repeated in some later simulations too. This rather unexpected results are assumed to be caused by the convergence of the simplex to a local minimum. If the local minimum is too shallow, the noise might "kick" the simplex outside this minimum. Therefore the simplex might find better minimum in the presence of noise.

In the same setup with signal and background noise the simplex performed significantly worse. It reached 293 turns after the first subset, but regressed to 266 turns after the second. With two more subsets it reached 1272 turns. Other simulations confirmed this observation. The conclusion made is that the simplex is likely to fail to find a good solution when the background noise is dominant in the result.

Later a new noise model with $\pm 0.3N$ turn count noise and ± 10 background noise was used to better simulate the experimental results after filtering. The simplex method performed well under these conditions. The mentioned before 1 % chance that the linac does not deliver beam did not had noticeable effect on the results.

Finally, a soft constraint was added to the objective function. With this constraint the simplex was forced inside the region within the limits of the corrector coils in the machine. The simulations confirmed that finding a good solution was still possible.

The stopping criteria of the optimizer was set to 20 iteration and 44 evaluation maximum for single subset of 12 correctors. It was noticed that when the simplex method struggles to find a solution it uses multiple contractions and the number of evaluations grows faster relative to the number of iterations.

Overall, the corrector optimization with the simplex method proved to be successful. The method proved to be sufficiently robust and rather fast with a small subset of correctors and stringent stopping criteria. The MABT algorithm prototype was deemed ready to be tested on the accelerator hardware.

4.1.2 Other optimization methods

Particle swarm optimization

The particle swarm optimization (PSO) was tested on simulation as an alternative algorithm to replace the simplex method in the MABT. The simulation used a small

storage ring model starting at 55 completed turns. Soft constraints, $\pm 0.3N$ turn count noise and ± 10 background noise were used. A single subset of 12 correctors was optimized.

The simplex method achieved a final number of completed turns in the interval 100 and 190 after 100 iterations and 240 to 270 evaluations. The results from PSO for 100 iterations with 10 agents were between 70 and 170 completed turns after 1012 evaluations. With 20 agents after 200 iterations and 4021 evaluations the final number of turns were 90 to 330.

The PSO algorithm does not converge as quickly to a minimum as the simplex method in the simulations. The swarm of agents requires a long period of time to evolve that is equivalent to a large number of iterations. At each iteration, every agent is moved to a new position and the objective function at that position is evaluated. The number of evaluations is thus the number of iterations multiplied by the number of agents.

Robust conjugate directional search

Although the robust conjugate directional search (RCDS) promises superior robustness, it has the same issue as PSO. In the limited simulation tests of single subset of 12 correctors the RCDS used over 870 objective evaluations just at the start. Perhaps the calculation of the Hessian and the noise evaluation are the cause, that is if the configuration was correct. The lack of time did not allow further investigation.

4.2 Experimental results

My results from the noise filters and the automated beam threader algorithms described in the previous chapter are presented in this section.

4.2.1 Noise filtering

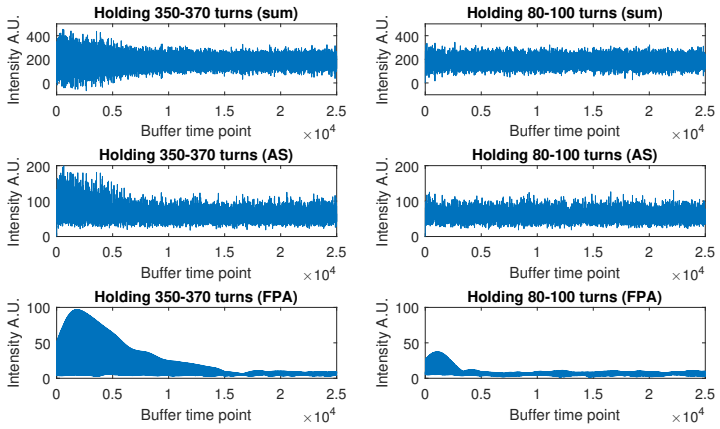

Figure 4.1: Comparison between filters on the small storage ring with BPM ADC data. Left side: signal of beam circulating for 350-370 turns at correctors with injection configuration. Right side: signal from beam circulating for 80-100 turns at worse corrector settings. The FPA filter separates the noise from the signal successfully, but introduces noticeable distortion in the amplitude.

Significant amount of work was put into separating the signal from the noise in the BPM ADC buffers and the oscilloscope. Early in the project samples of BPM ADC buffers were acquired from the 1.5 GeV storage ring. At this time it became apparent that the signal intensity was near the noise levels.

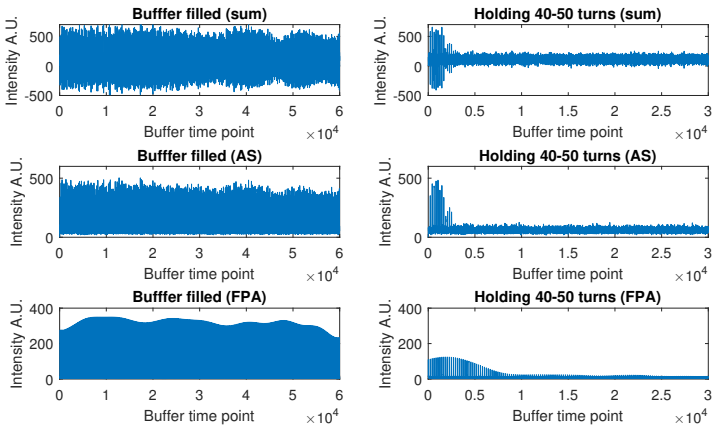

Figure 4.2: Comparison between filters on the large storage ring with BPM ADC data. Left side: signal of beam circulating for over 320 turns at correctors with injection configuration. Right side: signal from beam circulating for 40-50 turns at worse corrector settings. Despite holding half the turn number than the small ring in Figure 4.1 left, the signal in the large ring is clearly separated from the noise. The envelope distortion from the FPA filter is more pronounced.

The signal intensity in the BPM ADC buffers in the 3 GeV ring is significantly higher than the background noise. This is clearly visible in Figure 4.2 compared to Figure 4.1. This was first observed during the large ring commissioning week.

It can be noticed from Figure 4.2, the BPM ADC buffers can be completely filled with signal peaks. In the interval measured in the ADC buffer only about 315 turns can fit at most. This is caused by the larger revolution time in the large ring. The oscilloscope can measure longer intervals than the ADC buffer.

The oscilloscope in the large storage ring requires processing with FPA. In Figure 4.3 it can be seen that the signal to noise ratio of the oscilloscope is low. It is practically unusable without FPA filter. Even after filtering, significant fraction of the signal peaks can not be separated from the noise. This is discussed in more detail in the following section.

4.2.2 Orbit quality evaluation

The consistency of the quality value from experimental measurements is discussed. For the same corrector configuration the same result is expected.

The standard deviation in the number of turns counted by different BPMs is often above 30, but the beam to beam deviation is important for the reliability of the orbit quality value. The standard deviation of the number of turns for different beam shots is usually below 10 turns, as illustrated and commented in Figure 4.4. This includes the uncertainty in the counted number of peaks and the background noise, which is nearly eliminated with the used prominence.

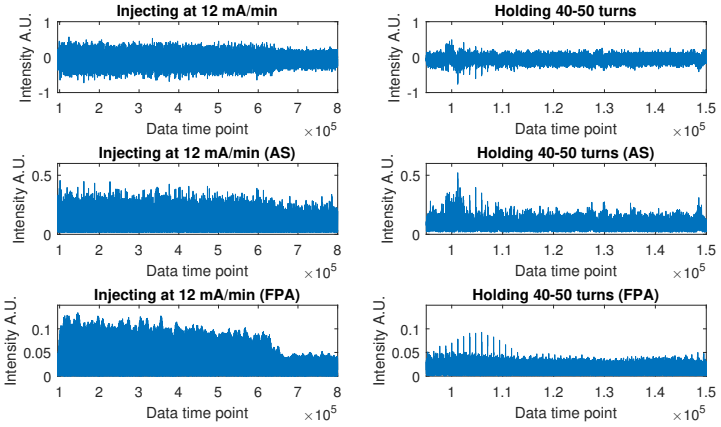

Figure 4.3: Comparison between filters on the large storage ring with oscilloscope data. Left side: signal of beam circulating for over many turns at correctors with injection configuration. Right side: signal from beam circulating for 40-50 turns at worse corrector settings. At low number of turns some fraction is not detected. The envelope distortion from the FPA filter is also noticeable at low number of turns.

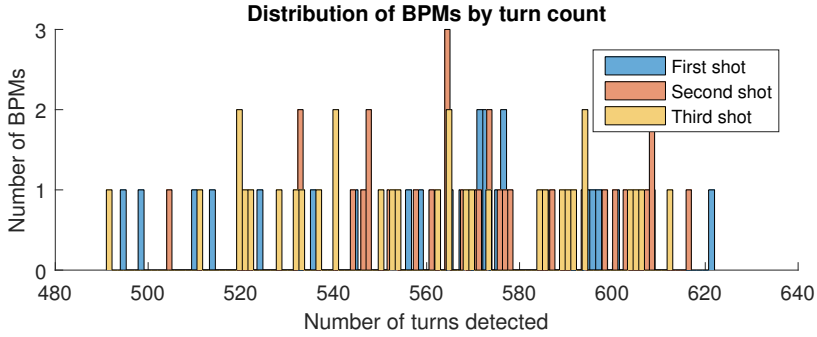

Figure 4.4: Distribution of BPMs by individual counted turns for three different beam shots in the small ring. The interval, where the BPMs are distributed, is over 100 turn difference long. In comparison, the difference in the average number of turns between the shots is within 12 turns.

The oscilloscope has low sensitivity. The number of turns counted in the oscilloscope is usually three to twenty times lower than the number of turns counted by the BPMs. This is the main reason to have a sensor function counting with BPMs for both single and multi-turn threading for the large ring. Regardless, the number of turns that is counted with the oscilloscope, changes consistently with the number of turns reported by the BPMs.

4.2.3 Single-turn algorithm

The single-turn automated beam threader (SABT) accomplished its task to guide the beam in the small and the large storage rings. The noise presence influenced the performance of the SABT and it required some initial correction on the small ring.

The best corrector magnet configuration is known as the injection settings. Starting from this configuration the strength of some correctors is adjusted until the beam is lost

after a small number of turns.

3 GeV storage ring

Because of the limited run time and technical problems, the performance of the final SABT algorithm version was measured only once on the large ring. To save time during that single-turn threading attempt, every third achromat was optimized instead of all achromats.

Starting from zero strength on all correctors in less than 30 minutes the single-turn threading was completed. Initially the beam was lost before completing a turn, then after the first optimized achromat it was able to complete a turn. The threading continued and after the fifth optimized achromat it was able to complete four turns. Finally, over 50 turns were counted after the seventh optimized achromat.

The single attempt was sufficient to prove that the SABT can successfully thread the large ring. The corrector magnets were set to zero strength, the equivalent to starting without previous knowledge. Furthermore, the threading was completed with the limitation of optimizing only a third of all achromats. Finally, the single-turn threading finished with multiple turns completed.

Later experiments provided execution speed measurements. The time spent for optimizing a single achromat is between 200 to 400 seconds. In that case, the estimated time to optimize all 20 achromats of the large ring is 1 to 2 hours. The performance of the threader could not be measured due to mistiming of the BPM ADC channels.

1.5 GeV storage ring

The same SABT algorithm was tested on the small ring. The single-turn threading could not be accomplished after setting the corrector strengths to zero. Instead the production values that are the best known settings, of the corrector strengths were decreased to some point.

The single-turn threading was completed by the algorithm with as low as 5 % for horizontal and 10 % for vertical corrector strengths relative to the production values. The threading was completed by only optimizing the second achromat, which is remarkable. Initially the beam was lost during its first turn and after the 184 seconds long optimization over 400 turns were completed. An injection rate of 40 mA/min was achieved after turning on the RF cavities. The injection rate with production configuration was 50 mA/min.

The performance of the SABT on the 1.5 GeV ring was mixed compared to that on the 3 GeV ring. The small initial correction that is required, keeps the beam from losing too many electrons during its passage through the accelerator to have a signal sufficiently above the noise level. At one or few completed turns a missed signal peak or an additional noise peak will result in a large relative error in the measurement. On the other hand with the given small initial correction the SABT was able to complete the threading procedure of the small ring in less than five minutes.

The execution speed is better on the 1.5 GeV ring. The main reason is the smaller number of correctors to optimize. A single achromat is usually optimized for 160 to 300 seconds. By estimation, the whole small ring can be optimized for about an hour or less.

4.2.4 Multi-turn algorithm

The SABT and the multi-turn automated beam threader (MABT) are capable of threading the 3 GeV ring successfully from zero initial corrector strength. This was achieved in less than an hour. The SABT and MABT are capable of threading the 1.5 GeV ring with very small initial correction.

The MABT was tested on rings with corrector strength derived from the production settings, similarly to the testing of the SABT. On few occasions the MABT was started directly after a SABT.

3 GeV storage ring

The multi-turn beam threading on the large ring was mostly done with the oscilloscope as a sensor. The limited testing time and technical issues did not allow very detailed study. Despite that there are results proving the ability of the MABT to thread the beam.

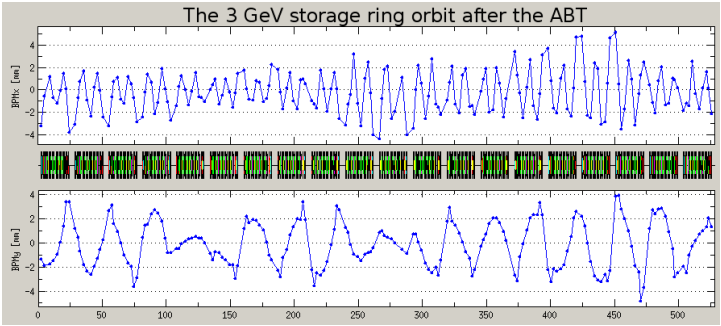

Figure 4.5: Large ring orbit after the ABT finished as reported by the BPMs along the ring. The top is the horizontal and bottom is the vertical beam position. The RMS orbit amplitude is 1.8 mm in both planes. This is very large amplitude.

The beam in the 3 GeV ring was threaded starting with 35 % for horizontal and 70 % for vertical corrector strengths relative to the production values. The initial number of turns completed by the beam was 50. After 226 seconds long optimization of the first corrector subset, 540 completed turns were reached and 9 mA/min injection rate was possible with RF cavity on. After the second subset optimization for 285 seconds, the beam completed over 680 turns. The injection rate improved to 11 mA/min.

The threading from the one successful SABT attempt at the large ring, described in the previous section, was continued with the MABT. Starting with about 59 turns, after the first subset optimization the beam was completing over 240 turns. During the second corrector subset optimizing a configuration reaching 320 turns was found, but the oscilloscope malfunctioned and the threading crashed at state of 240 completed turns.

Storing a beam was attempted directly afterwards. Only 50 μA were stored. This amount of current was still very low and an accurate orbit measurement was not possible. After some adjustments in the transfer line and the injection kicker done by an operator a current of 220 μA was stored. Higher values were not possible, because the injection kicker was kicking away part of the stored beam.

The orbit was obtained with the current of 220 μA. It is presented in Figure 4.5, where it is visible that the amplitude of the distortion is at points over 2 mm and it is also high at the start of the ring, where the injection kicker is. By having a stable orbit

that can be measured, other correction procedures such as inverse response matrix can be started.

The MABT can increase the number of completed turns rather quickly to a point where beam can be stored. A single subset takes 200 to 300 seconds to optimize when using the oscilloscope. With BPMs as sensors it takes 250 to 400 seconds. This is the only result with BPMs, due to the previously mentioned mistiming. The whole large ring can be optimized in less than two hours.

1.5 GeV storage ring

The lowest possible initial corrector strengths that allow the MABT to succeed in threading the beam are 20 % for horizontal and 30 % for vertical corrector strengths relative to the production values. The threading was completed with a single subset of correctors out of six subsets. From just a few completed turns over 500 turns were reached for 200 seconds. The injection rate was 56 mA/min.

More subsets further improved the threading result. After the MABT finished optimizing the whole ring, 680 turns were completed by the beam. The injection rate further improved to 58 mA/min. For comparison with production configuration the injection rate was 46 mA/min. This is explained with the optimizer implicitly optimizing the accepted charge efficiency, because the sensors are sensitive to the beam charge circulating during the threading. The orbit is shown in Figure 4.6.

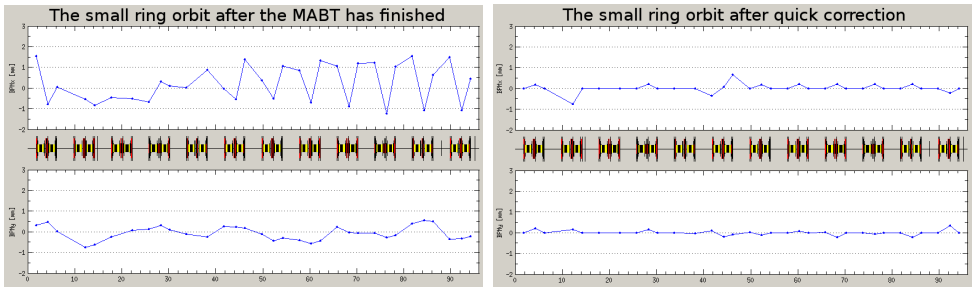

Figure 4.6: Left: the small ring orbit directly after beam threading as reported by the BPMs along the ring. The top is the horizontal and bottom is the vertical beam position.. Right: the orbit after 15 minute correction with stored beam.

Starting directly after the SABT at 5 % and 10 % horizontal and vertical corrector strengths respectively, three subsets were optimized with the MABT. The number of completed turns reached 530 and the injection rate was 53 mA/min. The injection rate with production settings was 50 mA/min that day. After the beam was stored, its orbit can easily be corrected, the result is presented in Figure 4.6. The MABT can thread a single subset of correctors for 200 to 320 seconds or optimize the whole ring in about half an hour.

4.3 Conclusion

The goal of developing an automated beam threader (ABT) for the MAX IV storage rings is completed. The recovery time was a main issue of the manual procedure. The automated threader is significantly faster and does not require nearly as much human labor.

The ABT may not have to optimize the whole ring. By estimation, the ABT execution takes more than an hour to optimize one of the MAX IV synchrotrons completely. Therefore if a beam can be stored, the ABT may be interrupted after optimization of a single unit, whether the unit is an achromat or a subset of correctors. The experimental results indicate that this is indeed the case. Often one of the storage rings can be threaded in less than twenty minutes.

The 3 GeV storage ring was threaded completely without previous correction using the automated threader in less than an hour with potential for more speed up. This is half the previous best manual beam threading time of two hours.

The decision to separate the automated threader into single-turn and multi-turn algorithms proved beneficial. Depending on the beam threading needed, both or the latter are used. The multi-turn algorithm had better time efficiency in increasing the number of completed turns as motivation to skip single-turn threading to save time.

The ABT algorithms are generally hardware independent. As a proof they function correctly on two synchrotrons with different lattices and parameters that are the 3 GeV and 1.5 GeV storage rings in the MAX IV facility. Yet there are some specific parameters in the ABT that need to be empirically determined.

The first such parameter is the number of corrector subsets. A good estimate for the number of subsets is when no less than three and no more that twenty corrector magnets are in a single subset. Secondly, the minimum peak prominence in the sensor function that is used when counting the number of turns comleted by the beam has to be changed. Different hardware has different noise levels and the prominence value is related to them. Finally, the FPA filter is filtering based on the revolution frequency in hardware units. The frequency window width in FPA is also dependent on the hardware response to a passing beam.

The noise in the signal remains an issue. The automated threading is robust and proved to function at poor signal to noise ratio. However, its performance and reliability degrade as the signal level approaches the noise level. With some initial correction the signal might be improved. Good initial correction in the previous production configuration is usually available.

The project time frame and limited run time did not allow more complete measurements on the performance of the ABT and extending its algorithms.

Chapter 5

Outlook

Further extensions of the automated threader are possible. To begin with, the determination of the minimum peak prominence may be automated by selecting the lowest prominence that eliminates noise in absence of signal. Next, it is technically possible to perform beam threading on the transfer line with the ABT using the single-turn method. For example, the algorithm can be extended to use the corrector magnets in the transfer line during the beam threading of the first achromat. The RF cavity phases can also be optimized to store beam if that is not possible initially. Beam threading the MAX IV linac with an ABT is another possibility that was not accomplished in this project.

The ABT was not tested with beam having 500 MHz structure formed by the thermionic pre-injector chopper. With the BPMs being most sensitive at that frequency, the signal to noise ratio may be improved [10]. Then the small ring may possibly be threaded without any correction from previous known configuration.

A single-pass beam position might also be measured with 500 MHz beam structure [10]. During threading the beam position relative to the reference orbit can be used in the objective value together with the number of completed turns. This may increase the efficiency and the execution speed of the ABT. Some optimizing methods like particle swarm optimization (PSO) can perform multi-objective optimization. The two objectives will be the beam position and the number of completed turns.

Returning to the topic of optimization methods, the PSO performance is dependent on its internal parameters. Speed improvements can be gained in PSO by adapting to the specific function that is optimized [21, 26]. The time constraints did not allow searching for optimal values for the PSO algorithm.

In case good parameters are found, the PSO might be used as a backup method during single-turn threading. The agents in PSO have wandering motion and may encounter isolated minima in otherwise flat objective function. In contrast, the simplex method tends to converge quickly even if no good objective value is found. However, the execution speed might remain an issue of PSO compared to simplex.

Acknowledgements

I would like to thank my supervisors, Magnus Sjöström and Sverker Werin, for the great feedback they provided. I am thankful to David Olsson for providing for explaining the PSO algorithm and giving me his implementation of it. I am also grateful to Mathias, Robin, Martin, Jacob, Viktor, David, Filip and the rest of the operator team for their support during the experiments. I am also thankful to Josefin Reftlér and Johanna Paulsson of the radiation safety team for explaining me the safety procedures and the reasoning behind them. Thanks to Johnny Kvistholm for making nice illustrations for my thesis and presentation. I am grateful to Georgia Paraskaki and Johan Eckdahl for the nice company and their advices. Finally, I want to thank the accelerator development group for giving me feedback on my presentations and the rest of the MAX IV staff for having such a nice laboratory.

Bibliography

[1] Hans Grote. Beam threading in the lhc. Technical report, CERN, 1998.

[2] Klaus Wille. *The Physics of Particle Accelerators: An Introduction*. Clarendon Press, 2001.

[3] MAX IV Facility. Detailed design report. Technical report, MAX IV Facility, 2010.

[4] John David Jackson. *Classical Electrodynamics Third Edition*. Wiley, 1998.

[5] Pedro F. Tavares, Simon C. Leeman, Magnus Sjöström, and Åke Andersson. The max iv storage ring project. *Journal of synchrotron radiation.*, 2014.

[6] Mikael Eriksson, J. Friso van der Veen, and Christoph Quitmann. Diffraction-limited storage rings – a window to the science of tomorrow. *Journal of Synchrotron Radiation*, 21(5):837--842, aug 2014.

[7] Les Dallin and Ward Wurtz. Towards a 4th generation storage ring at the canadian light source. Author(s), 2016.

[8] Dieter Einfeld, J Schaper, and M Plesko. Design of a diffraction limited light source (difl). In *Particle Accelerator Conference, 1995., Proceedings of the 1995*, volume 1, pages 177--179. IEEE, 1995.

[9] Martin Johansson, Bengt Anderberg, and Lars-Johan Lindgren. Magnet design for a low-emittance storage ring. *Journal of Synchrotron Radiation*, 21(5):884--903, aug 2014.

[10] Magnus Sjöström. Private communication.

[11] S. C. Leemann. Pulsed sextupole injection for sweden's new light source MAX IV. *Physical Review Special Topics - Accelerators and Beams*, 15(5), may 2012.

[12] Arkadiusz Kisiel, Wojciech Kitka, Lukasz Żytniak, Adriana Wawrzyniak, Lukasz Dudek, Piotr Goryl, and Maciej Kopec. Performance of the beam position monitor system in solaris synchrotron. 2016.

[13] S.C. Leemann. Injection with a single dipole kicker into the MAX IV storage rings. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 693:117--129, nov 2012.

[14] D. Olsson, F. Lindau, M. Eriksson, J. Andersson, and L. Malmgren. A chopper system for the MAX IV thermionic pre-injector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 759:29--35, sep 2014.

[15] Robert Lindvall. Private communication.

[16] A Götz, WD Klotz, and J Meyer. Object oriented programming techniques applied to device access and control. Technical report, 1992.

[17] A. Terebilo. Accelerator modeling with matlab accelerator toolbox. In *PACS2001. Proceedings of the 2001 Particle Accelerator Conference (Cat. No.01CH37268)*, volume 4, pages 3203--3205 vol.4, 2001.

[18] Johan Bengtsson, E Forest, and H Nishimura. Tracy-2 user's manual. *SLS Internal Document, February*, 1997.

[19] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308--313, jan 1965.

[20] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*. IEEE.

[21] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pages 69--73, May 1998.

[22] Xiaobiao Huang et al. Development and application of online optimization algorithms. Technical report, SLAC National Accelerator Laboratory (SLAC), 2017.

[23] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing, Second Edition*. Cambridge University Press, 1992.

[24] Philip Hall. The distribution of means for samples of size n drawn from a population in which the variate takes values between 0 and 1, all such values being equally probable. *Biometrika*, 19(3/4):240, dec 1927.

[25] Bahaa EA Saleh, Malvin Carl Teich, and Bahaa E Saleh. *Fundamentals of photonics*, volume 22. Wiley New York, 1991.

[26] David K. Olsson. Private communication.

Appendix A

User Manual

This chapter contains the user manual for MAX IV facility operators and researchers that want to use the ABT.

Preparation

The current ABT code was developed to perform beam threading after the beam successfully enters the synchrotrons. The beam threading of the linac and the transfer lines must be completed in advance. The injection kicker should also be set to presumably good strength.

Depending on the sensor used during the threading process, the BPMs or the oscilloscope have to be set up for the measurements.

The setup of the BPMs is done with the commands `enableADCChannels('BPM')` and `disableADCChannels('BPM')`. The first command allows consistent reading of the ADC buffers for beam threading, while the second reverts the changes from the first command by putting the BPMs in production settings.

The setup of the oscilloscope is done with the command `mabtR3Osciloscope.m`. It sets the oscilloscope to specific time range and interval between each point measured that are expected in the code responsible for the measurements with the oscilloscope.

Single-turn beam threading

The single-turn automated beam threader is started with the command `SingleTurnABT`.

After each achromat optimization is complete, the threader stops and waits for a key press. This is the best point to interrupt the threader or try to store a beam, if needed. Other adjustments can also be made at this point.

The algorithm outputs information to track its progress. For each evaluation of the objective that is equivalent to a measurement, the orbit quality, the standard deviation between BPMs and the constraint are printed in a single line. For each finished iteration of the optimization algorithm the number of the iteration, the number of evaluations, the current best value and the iteration operation are shown in a single line.

At the end after optimizing the corrector kicks the algorithm prints the total numbers of iterations and evaluations made.

Multi-turn beam threading

The single-turn automated beam threader is started with the command `MultiTurnABT`. It acts essentially the same way as the single-turn threader.

After each subset optimization is complete, the threader stops and waits for a key press. This is the best point to interrupt the threader or try to store a beam, if needed. Other adjustments can also be made at this point.

The algorithm outputs information to track its progress. For each evaluation of the objective that is equivalent to a measurement, the counted turns, the standard deviation between BPMs or 0 for the oscilloscope and the constraint are printed in a single line. For each finished iteration of the optimization algorithm the number of the iteration, the number of evaluations, the current best value and the iteration operation are shown in a single line.

At the end after optimizing the corrector kicks the algorithm prints the final number of counted turns, the total numbers of iterations and evaluations made.

Development guides

The ABT code depends on functions in the Matlab Middle Layer. If they are available for the new hardware, no major changes in the ABT code are needed.

The first step in extending the ABT to new machines or components is defining a device list to be optimized and a device list to be used for the measurement.

The next step is writing and testing a hardware specific filtering function that gives an objective value to be optimized.

The key global variables must be updated with the new hardware, possibly with new functions. Most importantly the corrector magnet lists and their limits must be set. They are presented in more detail in Appendix B.

The main loop selects the achromat/subset by an integer. By assigning an integer in the range of the loop to the extension it can be easily included in the ABT procedure. An example is the first achromat in SABT, that is treated differently.

Appendix B

Code Documentation

Multi-turn ABT

MABT variables

The Multi-turn automated beam threader makes use of 8 global variables: `mabt_qualitycall`, `mabt_iter`, `mabt_init`, `mabt_scale`, `mabt_sensor`, `mabt_limits`, `mabt_hcorr`, `mabt_vcorr`.

mabt_qualitycall Used to report some performance data. It is increased every time an objective value is calculated.

mabt_iter Used to report some performance data. It stores the total number of iterations done by the optimizer.

mabt_init Stores the initial settings of the combined device list of the horizontal and vertical corrector magnets in `mabt_hcorr` and `mabt_vcorr`. The initial settings are used as a starting point for the optimizer.

mabt_scale Stores a value that is used to scale the values given by the optimizer algorithm when calculating the settings for the corrector magnets in a subset. It is an optimizer specific value.

mabt_sensor A handle to a function that returns an orbit quality value that is used in the calculation of the objective value. It is an external function that is specific to the hardware that it uses.

mabt_limits Stores the design limits of the combined device list of the horizontal and vertical corrector magnets in `mabt_hcorr` and `mabt_vcorr`. The limits define where the optimizer constraint starts.

mabt_hcorr Device list of the horizontal corrector magnets. Only the correctors that are part of the current subset are in the list.

mabt_vcorr Device list of the vertical corrector magnets. Only the correctors that are part of the current subset are in the list.

Table B.1: MABT Functions

Name	Arguments	Returns	Description
`MultiTurnABT`	-	-	main function
`multiTurnOptimizer`	-	best objective function parameters	prepares and runs the optimization
`orbitQuality`	vector of objective function parameters	objective value	serves as the objective function
`setCorrectors`	vector of corrector strengths	-	sets the kicks of corrector coils in the subset to the given values
`getCorrectorSpace`	-	-	writes the `mabt_init` and `mabt_limits` variables
`correctorSubset`	subset number, amount of sunsets	-	writes the `mabt_hcorr` and `mabt_vcorr` variables

MABT functions

The function `MultiTurnABT` handles the machine detection and contains the main loop. In the loop the corrector subsets and corresponding variables are set and the optimized values are applied.

The function `orbitQuality` serves as an objective function for the optimizer. It uses the scaled values from the optimizer with the `mabt_init`, `mabt_limits` and `mabt_scale` variables to calculate the actual corrector kicks they correspond to. The correctors are set to these kicks and a number of completed turns is taken by calling the sensor function. A linearly growing saturation value is obtained from linear transformation in which -1 is a lower limit and $+1$ is an upper limit for each corrector strength. The corrector value furthest out these limits is picked as a saturation value and later is used to calculate the constraint.

Single-turn ABT

SABT variables

sabt_qualitycall Used to report some performance data. It is increased every time an objective value is calculated.

sabt_iter Used to report some performance data. It stores the total number of iterations done by the optimizer.

sabt_init Stores the initial settings of the combined device list of the horizontal and vertical corrector magnets in `sabt_hlist` and `sabt_vlist`. The initial settings are used as a starting point for the optimizer.

sabt_scale Stores a value that is used to scale the values given by the optimizer algorithm when calculating the settings for the corrector magnets in a subset. It is an optimizer specific value.

sabt_sensor A handle to a function that returns an orbit quality value that is used in the calculation of the objective value. It is an external function that is specific to the hardware that it uses.

sabt_limits Stores the design limits of the combined device list of the horizontal and vertical corrector magnets in `sabt_hcorr` and `mabt_vcorr`. The limits define where the optimizer constraint starts.

sabt_hlist Device list of the horizontal corrector magnets. Only the correctors that are part of the currently optimized achromat are in the list.

sabt_vlist Device list of the vertical corrector magnets. Only the correctors that are part of the currently optimized achromat are in the list.

sabt_achromat The number of the achromat where the beam sensor is located.

SABT functions

Table B.2: SABT Functions

Name	Arguments	Returns	Description
`SingleTurnABT`	-	-	main function
`singleTurnOptimizer`	-	best objective function parameters	prepares and runs the optimization
`orbitQuality`	vector of objective function parameters	objective value	serves as the objective function
`setCorrectors`	vector of corrector strengths	-	sets the kicks of corrector coils to the given values
`getCorrectorSpace`	-	-	writes the `sabt_init` and `sabt_limits` variables
`famfilter`	device family name, achromat number	device list	returns the devices of a family in an achromat

The function `SingleTurnABT` handles the machine detection and contains the main loop. In the loop the corrector subsets and corresponding variables are set and the optimized values are applied.

The function `orbitQuality` serves as an objective function for the optimizer. It uses the scaled values from the optimizer with the `sabt_init`, `sabt_limits` and `sabt_scale` variables to calculate the actual corrector kicks they correspond to. The correctors are set to these kicks and a number of signal peaks is taken by calling the sensor function. A

linearly growing saturation value is obtained from linear transformation in which -1 is a lower limit and $+1$ is an upper limit for each corrector strength. The corrector value furthest out these limits is picked as a saturation value and later is used to calculate the constraint.

Additional programs

count_ach_r1 This program returns the total number of signal peaks detected by all BPMs in a given achromat with prominence above fixed value. It is used during single-turn beam threading. This is the 1.5 GeV ring variant.

count_ach_r3 This program returns the total number of signal peaks detected by all BPMs in a given achromat with prominence above fixed value. It is used during single-turn beam threading. This is the 3 GeV ring variant.

get_count_r1bpm This program returns the average number of counted turns from all BPMs. It is used during multi-turn beam threading. This is the 1.5 GeV ring variant.

get_count_r3bpm This program returns the average number of counted turns from all BPMs. It is used during multi-turn beam threading. This is the 3 GeV ring variant.

get_count_r3bpm_fast This program returns the average number of counted turns from one tenth of all BPMs. It is used during multi-turn beam threading. This is a 3 GeV ring variant that executes faster by reducing accuracy.

get_count_r3scope This program returns the number of counted turns from the oscilloscope. It is used during multi-turn beam threading. It is useful for the 3 GeV ring only.

snfilter_as This program performs AS filtering on the signal in a Tango Controls reply from a single BPM.

snfilter_fpa_all This program performs FPA filtering on the signal in a Tango Controls reply from a single BPM. This is the 1.5 GeV ring variant.

snfilter3_fpa_all This program performs FPA filtering on the signal in a Tango Controls reply from a single BPM. This is the 3 GeV ring variant.

wvfilter3_fpa_all This program performs FPA filtering on the signal in a Tango Controls reply from the oscilloscope. This is the 3 GeV ring variant.

sabt_prominence This program performs peak search by prominence. To increase the execution speed some accuracy is sacrificed. This variant handles AS filtered signal during single-turn threading.

mabt_prominence This program performs peak search by prominence. To increase the execution speed some accuracy is sacrificed. This variant handles FPA filtered signal during multi-turn threading.

Appendix C

Source Code

MABT

```
function MultiTurnABT
% Multi-turn Automated Beam Threader

global mabt_qualitycall mabt_iter mabt_init mabt_scale mabt_sensor mabt_limits

switch getsubmachinename()
    case 'R1'
        disp('MAX_IV_R1_Multi-turn_threader')
        mabt_sensor = @get_count_r1bpm;% ALL BPMS
        dividecount = 6;% subset divisions
    case 'R3'
        disp('MAX_IV_R3_Multi-turn_threader')
        mabt_sensor = @get_count_r3scope;% OSCILLOSCOPE
%         mabt_sensor = @get_count_r3bpm;% ALL BPMS
%         mabt_sensor = @get_count_r3bpm_fast;% 1/10 BPMS
        dividecount = 20;% subset divisions
    otherwise
        disp('Unknown_machine')
        return
end

mabt_iter = 0;
mabt_qualitycall = 0;
mabt_scale = 0.1/250e-6;

for setselect = 1:dividecount
    correctorSubset(setselect, dividecount);
    getCorrectorSpace();
    opt_val = multiTurnOptimizer();
    setCorrectors(opt_val .* (mabt_scale*(mabt_limits(:,2)-mabt_limits(:,1))) + mabt_init);
    pause()
end

fprintf('Turns:_%5.1f;_Iter:_%5.1f;_Eval:_%5.1f\n', mabt_sensor(), mabt_iter, mabt_qualitycall);

end

function X = multiTurnOptimizer()
global mabt_iter mabt_init

% R1 6 sets: 44 evaluastions
% R3 20 sets: 58 evaluations
maxobjeval = 52;

x0 = zeros(size(mabt_init));

OPTIONS = optimset('Display','iter','MaxIter',20,'MaxFunEvals',maxobjeval,'PlotFcns',@optimplotfval);
[X,~,~,OUTPUT] = fminsearch(@orbitQuality,x0,OPTIONS);
mabt_iter = mabt_iter + OUTPUT.iterations;

disp('Finished_subset')

end

function Turns = orbitQuality(CorrVals)

global mabt_qualitycall mabt_init mabt_scale mabt_sensor mabt_limits

mabt_qualitycall = mabt_qualitycall + 1;

% transform input into real kick strengths
CorrVals = CorrVals .* (mabt_scale*(mabt_limits(:,2) - mabt_limits(:,1))) + mabt_init;
% determine the oversaturation
S = max(abs((CorrVals - mabt_limits(:,2))./(mabt_limits(:,2) - mabt_limits(:,1))*2 + 1));

setCorrectors(CorrVals);

[Turns,Noise] = mabt_sensor();
```

```
Turns = -Turns;

% Constraint
Cons = 0;
if S > 1; Cons = 2000*((S-0.9)/0.1)^4; end

fprintf('Turns: %5.1f; SDev: %6.2f; Constraint: %5.1f\n', -Turns, Noise, Cons);
Turns = Turns + Cons;
end

function setCorrectors(CorrVals)

global mabt_limits mabt_hcorr mabt_vcorr

% hold values within limits
CorrVals(CorrVals > mabt_limits(:,2)) = mabt_limits(CorrVals > mabt_limits(:,2),2);% upper limit
CorrVals(CorrVals < mabt_limits(:,1)) = mabt_limits(CorrVals < mabt_limits(:,1),1);% lower limit

NbrOfCorrs.HCM = size(mabt_hcorr,1);
NbrOfCorrs.VCM = size(mabt_vcorr,1);

HorVals = CorrVals(1:size(mabt_hcorr,1),1);
VerVals = CorrVals(NbrOfCorrs.HCM+(1:NbrOfCorrs.VCM),1);

setsp('HCM',HorVals,mabt_hcorr);
setsp('VCM',VerVals,mabt_vcorr);
pause(3)% NEEDED !!!
end

function getCorrectorSpace()

global mabt_hcorr mabt_vcorr mabt_init mabt_limits
HorVals = getsp('HCM',mabt_hcorr);
VerVals = getsp('VCM',mabt_vcorr);
mabt_init = [HorVals; VerVals];% initial kick values before the optimization

HorLim = getfamilydata('HCM','Setpoint','Range', mabt_hcorr);
VerLim = getfamilydata('VCM','Setpoint','Range', mabt_vcorr);
mabt_limits = [HorLim; VerLim];% the limits for each individual corrector
end

function correctorSubset(setselect, setcount)
% create a subset of correctors
persistent DevHCM DevVCM
if isempty(DevHCM); DevHCM = family2dev('HCM'); end
if isempty(DevVCM); DevVCM = family2dev('VCM'); end
global mabt_hcorr mabt_vcorr
mabt_hcorr = DevHCM(setselect:setcount:end,:);
mabt_vcorr = DevVCM(setselect:setcount:end,:);
end
```

SABT

```
function SingleTurnABT
% Single-turn Automated Beam Threader

global sabt_qualitycall sabt_iter sabt_hlist sabt_vlist sabt_achromat sabt_limits sabt_init sabt_scale sabt_sensor

switch getsubmachinename()
    case 'R1'
        disp('MAX_IV_R1_Single-turn_threader')
        achromat_count = 12;
        sabt_sensor = @count_ach_r1;
        disp('R1_Threader')
    case 'R3'
        disp('MAX_IV_R3_Single-turn_threader')
        achromat_count = 20;
        sabt_sensor = @count_ach_r3;
        disp('R3_Threader')
    otherwise
        disp('Unknown_machine')
        return
end

sabt_iter = 0;
sabt_qualitycall = 0;
sabt_scale = 0.4/250e-6;

step = 1;
for ach = 1:step:achromat_count
    if ach == 1
        sabt_hlist = famfilter('HCM', ach);
        sabt_vlist = famfilter('VCM', ach);
    else
        sabt_hlist = famfilter('HCM', ach-1);
        sabt_vlist = famfilter('VCM', ach-1);
    end
    getCorrectorSpace();
    sabt_achromat = ach;
    setCorrectors(sabt_init);
    opt_val = singleTurnOptimizer();
    setCorrectors(opt_val .* (sabt_scale*(sabt_limits(:,2) - sabt_limits(:,1))) + sabt_init);
    pause()
end
```

```
fprintf('Iter: %5.1f; Eval: %5.1f\n', sabt_iter, sabt_qualitycall);

end

function X = singleTurnOptimizer()
global sabt_iter sabt_init

S = @orbitQuality;

x0 = zeros(size(sabt_init));

OPTIONS = optimset('Display','iter','MaxIter',20,'MaxFunEvals',32,'PlotFcns',@optimplotfval);
[X,~,~,OUTPUT] = fminsearch(S,x0,OPTIONS);
sabt_iter = sabt_iter + OUTPUT.iterations;

disp('Finished achromat')

end

function Turns = orbitQuality(CorrVals)

global sabt_qualitycall sabt_achromat sabt_limits sabt_init sabt_scale sabt_sensor

sabt_qualitycall = sabt_qualitycall + 1;

% transform input into real kick strengths
CorrVals = CorrVals .* (sabt_scale*(sabt_limits(:,2) - sabt_limits(:,1))) + sabt_init;
% determine the oversaturation
S = max(abs((CorrVals - sabt_limits(:,2))./(sabt_limits(:,2) - sabt_limits(:,1))*2 + 1));

setCorrectors(CorrVals);

[Turns,Noise] = sabt_sensor(sabt_achromat);

Turns = -Turns * 10 + Noise;% the noise addition is a trick to avoid simplex convergence

Cons = 0;% constraint
if S > 1, Cons = 2000*((S-0.9)/0.1)^4; end

fprintf('Quality: %5.1f; SDev: %6.2f; Constraint: %5.1f\n', -Turns, Noise, Cons);
Turns = Turns + Cons;
end

function setCorrectors(CorrVals)

global sabt_hlist sabt_vlist sabt_limits

% hold values within limits
CorrVals(CorrVals > sabt_limits(:,2)) = sabt_limits(CorrVals > sabt_limits(:,2),2);
CorrVals(CorrVals < sabt_limits(:,1)) = sabt_limits(CorrVals < sabt_limits(:,1),1);

DevList.HCM = sabt_hlist;
DevList.VCM = sabt_vlist;
NbrOfCorrs.HCM = size(DevList.HCM,1);
NbrOfCorrs.VCM = size(DevList.VCM,1);

HorVals = CorrVals(1:NbrOfCorrs.HCM,1);
VerVals = CorrVals(NbrOfCorrs.HCM+(1:NbrOfCorrs.VCM),1);

setsp('HCM',HorVals,DevList.HCM);
setsp('VCM',VerVals,DevList.VCM);
pause(5)% NEEDED !!!
end

function DevList = famfilter(famname, achromat)
% filter devices by family and achromat
DevList = family2dev(famname);
DevList = DevList(DevList(:,1)==achromat,:);
end

function getCorrectorSpace()
global sabt_hlist sabt_vlist sabt_init sabt_limits
HorVals = getsp('HCM',sabt_hlist);
VerVals = getsp('VCM',sabt_vlist);
sabt_init = [HorVals; VerVals];% initial kick values before the optimization

HorLim = getfamilydata('HCM','Setpoint','Range', sabt_hlist);
VerLim = getfamilydata('VCM','Setpoint','Range', sabt_vlist);
sabt_limits = [HorLim; VerLim];% the limits for each individual corrector
end
```