A Linear Framework for Orbit Correction in the High-Luminosity Large Hadron Collider

Joel Andersson

Master's thesis
2019:E61

LUND UNIVERSITY

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

CENTRUM SCIENTIARUM MATHEMATICARUM

A Linear Framework for Orbit Correction in the High–Luminosity Large Hadron Collider

Joel Daniel Andersson
joel.dan.andersson@gmail.com

December 12, 2019

LUND UNIVERSITY

Master's thesis work carried out at CERN with support from the HL–LHC project.

Supervisors: Alexandros Sopasakis, sopasak@maths.lth.se
Davide Gamba, davide.gamba@cern.ch

Examiner: Eskil Hansen, eskil.hansen@math.lth.se

Abstract

In a circular accelerator the closed orbit can be viewed as the mean position of particles in a beam. The closed orbit is perturbed by machine errors and can be manipulated by dedicated corrector magnets. This thesis introduces a linear algebra framework for closed orbit perturbation and correction, its implementation as a Python package and its use for three studies in HL–LHC: orbit corrector budget, orbit feedback expected performance analysis and specifications for new beam position monitors. The orbit corrector budget is formulated as a convex optimization problem and solved for the current iteration of HL–LHC. Results based on a simplified model for the orbit feedback are presented, showcasing its inefficacy in maintaining collision on its own and the inherent stability in LHC. Necessary short-term beam position monitor stability for adequate position-based correction of beam separation is estimated to be under one micrometer. Finally, optimizing over linear correction strategies is offered as an interesting venue for further research.

Keywords: CERN, LHC, HLLHC, HiLumi, orbit correction, error correction, knob implementation, linear algebra, matrix, closed orbit, accelerator physics, Twiss, beam dynamics, dynamical systems, applied mathematics, optimization, convexity, pseudoinverse, SVD, BPM, Python, MSc

Acknowledgements

I would like to thank my supervisor at CERN, Davide Gamba, the hands-down biggest contributor to this thesis whose name is not on the front cover. The idea behind the framework, as well as a proof-of-concept suite of MATLAB scripts, are his making and in every meaningful way the foundation on which this thesis is built. In addition to his direct contribution to the project's foundation, Davide's mentoring has been indispensable. I arrived at CERN with no formal education in accelerator physics, and was subsequently taught the notion of emittance and particle injection based on Davide playing with crumbs in the cafeteria –– successfully. Thank you Davide.

I would also like to thank Riccardo De Maria for interesting discussions about, and interest in, the framework applied to the orbit corrector budget.

In the same vein, I want to thank Jörg Wenninger for enlightening me about the orbit feedback system used in LHC and operational experience from using it.

I would like to to thank my home institute supervisor Alexandros Sopasakis for providing swift support and feedback when requisite, and for trusting me in the interim.

I thank Tobias Persson for supervising me in the summer of 2018 for my CERN openlab project. Without Tobias, there is little reason to believe that I would have later come to CERN as a Technical Student, nor that I would have considered myself a worthy candidate for it. Thank you for the support Tobias.

The typical work environment for non-staff personnel at CERN is shared offices, and with this in mind I would like to thank my office mates in 9/R-014. Thanks for your guidance, opinions on, and distractions away from my work, it has all been much appreciated.

For much the same reason, I want to thank my classmates, corridor co-habitants and friends in general for the duration of my studies in Lund. Thanks for keeping me sane.

I would like to thank my parents who humour my scholarly ambitions and who would have been about as happy with me if I went into plumbing. Thank you for making me stubborn, keeping my ego in check and caring deeply.

Lastly, I would like to thank my little brother Alexander for being two years my junior, following my footsteps and keeping even pace. You make me very proud Alex.

Contents

Chapter 1
Introduction

Coming up with a good subject for a Master's thesis is hard work. By the paradox of choice, it follows that it is the Engineering Physics student that has to work the hardest to find one. For this reason, I was very thankful for Davide's offer to come work at CERN on orbit correction for HL–LHC. However, I was not only spared the burden of finding a subject to work on, I was also offered a project which epitomized what Engineering Physics is all about: multi-disciplinary work. This thesis ties many different fields together, but the overarching theme is applied mathematics; mathematics applied to the study of first-order closed orbit perturbation and correction in HL–LHC.

Having said this, the subject matter is technical and heavy in jargon. To accommodate for this, the choice was made early to make the thesis self-contained. In retrospect I think this has been largely successful, but the page count had to suffer as a result. All the necessary theory, most notably transverse beam dynamics, is given in some detail in Chapter 2. Technical jargon is introduced ad hoc with a Glossary and a Nomenclature at the end of the thesis. I hope that the reader identifies that while transverse beam dynamics is integral to the application, and the jargon necessary to aptly describe it, all the actual analysis is based on linear algebra, statistics and a little convex optimization.

1.1 CERN and the HL–LHC Project

The full thesis was carried out at CERN, Meyrin, Switzerland, and its subject matter is heavily intertwined with CERN's mission and facilities. As such, it is only reasonable that a short summation of CERN follows. The European Organization for Nuclear Research, CERN (abbreviation in French: Conseil Européen pour la Recherche Nucléaire) is the world's largest particle physics laboratory. CERN was founded in 1954 as a means of pooling European nations' resources together for the purpose of conducting frontier nuclear research in Europe, and as a peace project in the aftermath of the second world war.

To pursue this research, CERN was to become more than a theory workshop, it was to

become a de facto laboratory, and the instrumentation of choice for this was particle accelerators. Simply put, by accelerating charged particles to high energy and have them interact with other particles, information could be retrieved about their nuclei. As time progressed and the understanding of nuclear physics accrued, the focus shifted from nuclear to subnuclear physics, or more broadly put, particle physics. The constituents of atoms are well understood, but what are the constituents of the constituents?

In parallel with this research trend towards the understanding of and relation between more and more fundamental constituents of matter, the accelerators were put on a trajectory towards achieving the corresponding particle energies necessary for probing this region. Greater energies means beams are harder to bend, which to a large extent explains the scale of new circular accelerators.

Today the latest and biggest accelerator at CERN is the Large Hadron Collider (LHC) completed in 2008 [1]. It is a 27-kilometer ring of well over a thousand super-conducting magnets kept at a temperature below three Kelvin. Somewhat simplified, two beams of particles are accelerated through the LHC in parallel but opposite direction and made to collide at fixed interaction points to study their collision. A schematic of this is presented in Figure 1.1

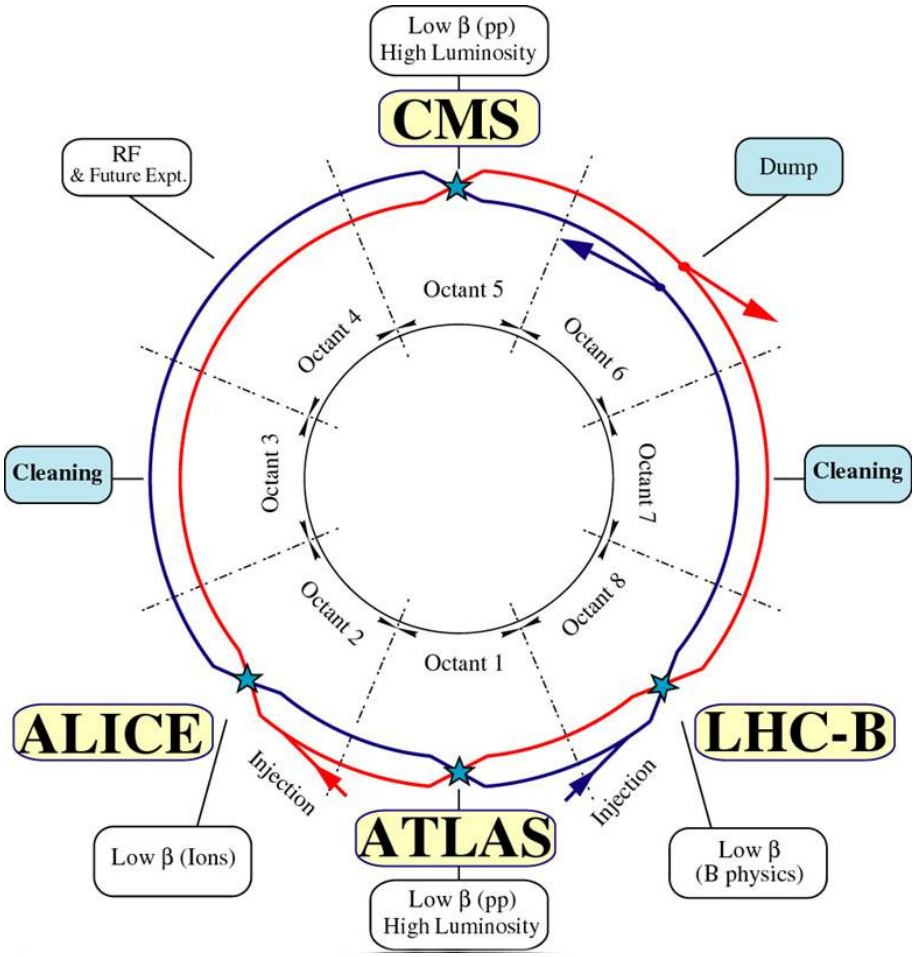


Figure 1.1: Schematic of LHC with interaction points [1].

LHC is divided into eight octants, half of which host collisions between the two beams at designated positions called *interaction points*. The interaction points are `IP1` (ATLAS), `IP2`

(ALICE), `IP5` (CMS) and `IP8` (LHC-B), where the experiments at each interaction point are given in parenthesis. In 2012, the ATLAS and CMS experiments conducted at the LHC reported the detection of the Higgs Boson, leading to the 2013 Nobel Prize in Physics being awarded to François Englert and Peter W. Higgs for their formulation of it.

Today, the evolution of CERN's accelerator complex continues. Most notably, the LHC will undergo a major upgrade during the 2020s to extend its operability another decade, thereby giving it the new name High-Luminosity Large Hadron Collider (HL–LHC or Hi-Lumi for short) [2]. As the name gives away, one of the goals of the HL–LHC is to increase its *luminosity* (and thereby its rate of collisions), and this by a factor of five beyond the original design value of the LHC, all the while keeping the same structure and tunnel.

The LHC being already a highly complex machine, any upgrade must be studied to make sure the design remains stable and functional. This typically involves simulation studies, probing everything from element placement to magnet design to strategies for the correction of errors. This is the context in which this thesis exists, it is a part of the simulation studies for the HL–LHC. More specifically, this thesis introduces a framework for machine error induced orbit perturbations, correction of these orbit perturbations and the implementation of *orbit knobs*, where an orbit knob is taken as a shorthand for any routine that manipulates the beam orbit in a controlled manner, typically for a specific operation scenario. Having implemented this framework, three studies will be conducted inside it. A more detailed explanation of the primary study is given next in prose, with the actual physics and mathematical formalism of the underlying problem postponed to Chapter 2, 3 and 4.

1.2 Primary Thesis Objective in Prose

A circular accelerator necessitates a set of components which are indispensable. In the first order, they are:

1. Bending magnets. If beams of particles are to perform a circular motion then some force is necessary to bend the beam. Bending magnets are used for this.

2. Quadrupoles. If a laser pointer is targeting a wall far away, then the laser dot will have become larger than the pointer's aperture by the time it hits the wall. Analogously, beams of particles will spread radially as they travel in an accelerator if no mediating force acts on them; let that go on for long enough and the beam will eventually hit the inside of the beampipe. What is needed is some form of *focusing*, something that stabilizes the beam to keep its cross-section from expanding. Quadrupoles achieve this.

3. Orbit correctors. If a beam needs to be fine-steered in the accelerator, e.g. there is a pipe which the beam has to pass through which is slightly off the path the beam otherwise would have gone, one can power orbit correctors to steer the beam into it.

4. Beam Position Monitors (BPMs). In order to steer a beam it is necessary to know its position. BPMs are sensors that measure the position of beams within the beampipe.

As it happens, bending magnets and quadrupoles installed into an accelerator are prone to have errors, which albeit small, have noticeable effect on beams in the accelerator. One such effect is beam orbit perturbation. To correct for it, one powers orbit correctors which

counteract the effects of the errors as measured by the BPMs. This is called *orbit error correction*. In addition to this, the orbit correctors have to be used for various fine-steering routines, e.g. moving the beam at some point or have it form a certain angle at some other. These routines are called *orbit knobs*. Finally, there is a limit put on how much each orbit corrector can be used. With this in mind, a natural question comes to mind: *is there enough orbit corrector strength to both correct for machine errors and to implement all desired knobs?* This is the primary objective of this thesis, to design and implement a framework for answering the question of whether the *orbit corrector budget* holds.

1.3 Clarification of Objectives

Leaving the realm of prose, the objective can be more technically enunciated. The primary objective of this thesis is to develop a Python framework for HL–LHC capable of:

1. Computing the orbit perturbation effects of machine element imperfections and misalignments.
2. Computing the orbit response of orbit correctors.
3. Implementing knobs.
4. Performing orbit error correction.
5. Verifying the integrity of the orbit corrector budget.

The aim is for the framework to be used continuously in the verification process of HL–LHC optics; verifying that the error correction and knob implementation are feasible and satisfy the orbit corrector budget. This is the primary objective, the secondary objective is to put the framework to use for three studies as part of the HL–LHC Work Package 2 (WP2) group at CERN. These studies are:

1. Verifying that the v1.5 HL–LHC optics satisfy the orbit corrector budget.
2. Investigating the performance of the Orbit FeedBack system (OFB) employed in LHC for maintaining collision and how it transfers to HL–LHC.
3. Estimating the necessary BPM stability for employing local IP correction during collision.

1.4 Previous work

Had it not been for prior work and investigations into this problem of finding feasible corrector configurations, this would not have been a feasible Master's thesis; it would have been incompatible with the time budget. It is with this in mind that I owe a good deal to my CERN supervisor Davide Gamba and his colleague Riccardo De Maria, among others, for their previous work on this subject (e.g. [3]).

Dr. Gamba had already developed a set of MATLAB scripts for accomplishing similar results by directly interfacing with the accelerator design software MAD-X [4]. This was a proof-of-concept and was estimating perturbations caused by errors and the effects of orbit correctors by modifying one element at a time and retrieving the difference in orbit. In doing so, his framework could build up the matrices necessary for this formalism by repeatedly calling MAD-X. This work differs in three major respects:

1. The framework takes input in the form of `.tfs` tables, a MAD-X file format, which contain all the relevant information for a linear treatment of the problem, and to formulaically compute the effects on the orbit. This allows for a speed-up in analyzing optics and decouples the framework from direct interaction with MAD-X.

2. Optimization over orbit corrector configurations is included for error correction and knob implementation. This was not present before.

3. Three additional studies based on the framework are conducted.

Chapter 2
Theory

In this chapter the theory that was previously withheld will now be given. It will be split into three parts:

1. Transverse Beam Dynamics
2. Linear Algebra
3. Optimization

The notions introduced in this chapter will be implement to provide the framework described in Chapter 3, subsequently applied to the problems treated in this thesis.

2.1 (Transverse) Beam Dynamics

In this section the essentials of Transverse Beam Dynamics necessary for this thesis will be accounted for. For a more comprehensive review, see Wolski's Beam Dynamics in High Energy Particle Accelerators [5] which has inspired the approach taken in this section. Our review will be incremental, starting with single particle analysis.

2.1.1 Particle Dynamics

Consider a Cartesian coordinate system and a charged particle with mass m and charge q. Assuming a force $\mathbf{F}$ is applied to the particle with momentum $\mathbf{p}$, Newton's second law states:

$$\frac{d\mathbf{p}}{dt} = \mathbf{F}\,. \tag{2.1}$$

In the case of it being in a electromagnetic field, the Lorentz force is given by

$$\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B})\,. \tag{2.2}$$

One approach to computing the position of a single particle moving through the accelerator would be to directly solve the equations posed in (2.1) and (2.2). This is possible, but not the typical approach in beam dynamics [5, p. 60]. One of the primary reasons is that the integration with respect to time would require the electromagnetic fields to be expressed as functions of time. In beam dynamics the electromagnetic fields are more easily defined as functions of the position along the accelerator, which instead warrants integration with respect to path length. An alternative treatment that allows for this is Hamiltonian mechanics, which is the approach used for this thesis and beam dynamics in general. Hamilton's equations can be stated as:

$$\begin{aligned} \frac{dx_i}{dt} &= \frac{\partial H}{\partial p_i}, \\ \frac{dp_i}{dt} &= -\frac{\partial H}{\partial x_i}, \end{aligned} \tag{2.3}$$

where x_i and p_i correspond to the i^{th} dimension for the position and momentum of the particle, respectively, and H is a function, the Hamiltonian. As Newton's second law and Hamilton's equations are both able to describe the same system, the force in the latter case must be encoded in the Hamiltonian. For example, a one-dimensional harmonic oscillator would in Newtonian mechanics be described by the force

$$F = -kx\,, \tag{2.4}$$

whereas in Hamiltonian mechanics the corresponding system would be defined by

$$H = \frac{p^2}{2m} + \frac{k}{2}x^2\,. \tag{2.5}$$

For accelerator physics, the relevant force is the Lorentz force in (2.2), hence the equivalent Hamiltonian has to be derived. First, we recall how the scalar potential ϕ and vector potential $\mathbf{A}$ are related to the electromagnetic fields:

$$\begin{aligned} \mathbf{E} &= -\nabla\phi - \frac{\partial \mathbf{A}}{\partial t}, \\ \mathbf{B} &= \nabla \times \mathbf{A}\,. \end{aligned} \tag{2.6}$$

In the non-relativistic case, this resolves to the Hamiltonian:

$$H = \frac{(\mathbf{p} - q\mathbf{A})^2}{2m} + q\phi\,. \tag{2.7}$$

Observe that $\mathbf{p}$ in (2.7) is not the mechanical momentum but the *canonical momentum*. These are not guaranteed to be equal, as this particular case demonstrates by applying (2.3) to this Hamiltonian:

$$v_i = \frac{dx_i}{dt} = \frac{\partial H}{\partial p_i} = \frac{p_i - qA_i}{m} \implies \mathbf{p} = m\mathbf{v} + q\mathbf{A}\,. \tag{2.8}$$

In essence, Hamiltonian mechanics will predict the same orbit but its momentum, called the canonical momentum, is not necessarily the same as the mechanical momentum. To resolve the naming conflict, $\mathbf{p}$ is taken to be the canonical momentum and $\mathbf{p}_{mech}$ the mechanical momentum.

Note that the Hamiltonian in (2.7) is not relativistic, whereas the system of analysis (particle beams in HL–LHC) is, hence a relativistic expression for the Hamiltonian needs to be derived. Recall from special relativity that a particle in free space (i.e. no electromagnetic fields present) has a total energy given by

$$E_{free}^2 = \mathbf{p}_{mech}^2 c^2 + m^2 c^4 \,, \tag{2.9}$$

where m is the rest mass and c the speed of light in vacuum. With the same notation, the following relations are valid:

$$\begin{aligned} E_{free} &= \gamma m c^2 \,, \\ \mathbf{p}_{mech} &= \beta \gamma m c \,, \\ \beta &= \mathbf{v}/c \,, \\ \gamma &= \frac{1}{\sqrt{1-\beta^2}} \,. \end{aligned} \tag{2.10}$$

To get the total energy E and the canonical momentum $\mathbf{p}$ in presence of electromagnetic fields, the contribution from the fields has to be added:

$$\begin{aligned} E &= E_{free} + q\phi \,. \\ \mathbf{p} &= \mathbf{p}_{mech} + q\mathbf{A} \,. \end{aligned} \tag{2.11}$$

To find the Hamiltonian we note that for some systems the Hamiltonian is equal to the total energy of the particle. Looking back on (2.5), it proved correct for the harmonic oscillator. With that in mind, if the total energy can be expressed in the canonical momentum then it is possible that the expression could correspond to the Hamiltonian. This can be achieved by combining (2.9) and (2.11):

$$(E - q\phi)^2 = E_{free}^2 = \mathbf{p}_{mech}^2 c^2 + m^2 c^4 = (\mathbf{p} - q\mathbf{A})^2 c^2 + m^2 c^4 \,, \tag{2.12}$$

which upon solving for E gives a guess for a possible Hamiltonian:

$$H = E = c\sqrt{(\mathbf{p} - q\mathbf{A})^2 + m^2 c^2} + q\phi \,, \tag{2.13}$$

and this turns out be a correct Hamiltonian for a relativistic particle in a electromagnetic field [5, p. 63].

2.1.2 Integrating with Respect to Path Length

First, consider a particle moving in one dimension. We can express this as a function of t where each point in time is described by a tuple $(x(t), p(t))$, where x and p are referred to as *dynamical variables*. Over the evolution in time, this particle will form a continuum of points in the two-dimensional space x, p called the *phase space*. For any such path (physical or not), one can compute the *action* S between any two times defined as:

$$S = \int_a^b [p\dot{x} - H] \, dt \,. \tag{2.14}$$

Given two points in phase space which a system passes through at two separate times, the actual path between them, as defined by Hamilton's equations, will minimize the action. The integral in (2.14) can be rewritten as an integral over x as per:

$$S = \int_a^b [p\dot{x} - H]\, dt = \int_{x(a)}^{x(b)} p\frac{dx}{dt}\frac{dt}{dx}dx - H\frac{dt}{dx}dx = \int_{x(a)}^{x(b)} \left[(-H)\frac{dt}{dx} + p\right] dx\,, \quad (2.15)$$

which, with the following variable substitution:

$$\begin{aligned} H' &= -p\,, \\ p' &= -H\,, \\ a' &= x(a)\,, \\ b' &= x(b)\,, \\ t' &= x\,, \\ x' &= t\,, \end{aligned} \quad (2.16)$$

becomes:

$$S = \int_a^b [p\dot{x} - H]\, dt = \int_{a'}^{b'} \left[p'\frac{dx'}{dt'} - H'\right] dt'\,. \quad (2.17)$$

In the new variables, a representation of the Hamiltonian system has been constructed that uses the coordinate x as the free variable and different canonical variables. They define the same action and hence minimizing one will minimize the other, which in turn means that they describe the same path. For clarity, the new equations for the system are written down in terms of the original variables:

$$\begin{aligned} \frac{dt}{dx} &= \frac{\partial p}{\partial H}\,, \\ \frac{dH}{dx} &= -\frac{\partial p}{\partial t}\,. \end{aligned} \quad (2.18)$$

As is visible from (2.18), this new system gives solutions to time and Hamiltonian as functions of x and they are derived from the original canonical momentum p. It is worth spending a moment interpreting the meaning of this. When the new system of equations is solved, $t(x)$ and $H(x)$ are retrieved, which means that for any given position, it is possible to predict the time of arrival and the total energy of the particle at that point. This is to be compared to the original system (2.3), where instead $x(t)$ and $p(t)$ are solved for, i.e. the position and canonical momentum at any given time.

Note that this was derived for the case of one degree of freedom, whereas a particle inside an accelerator has three. To extend the result, the derivation for the case of a accelerator segment that is straight and use a Cartesian coordinate system (x, y, z) where the z-axis runs parallel to the length of the segment follows. The action from (2.14) directly translates to:

$$S = \int_a^b [p_x\dot{x} + p_y\dot{y} + p_z\dot{z} - H]\, dt\,, \quad (2.19)$$

where everything has been defined analogously and H relates to the Hamiltonian for the new three-dimensional system, which was found by expanding the vectors in (2.13). It is

possible to apply the same trick as before in (2.15) to the position z, i.e. the position along the accelerator, to express the action as an integral over the path along the accelerator rather than time. Although the reader might already have guessed the outcome, the derivation follows for clarity:

$$\begin{aligned}\int_a^b \left[p_x\dot{x} + p_y\dot{y} + p_z\dot{z} - H\right] dt &= \int_{z(a)}^{z(b)} \left[p_x\dot{x}\frac{dt}{dz} + p_y\dot{y}\frac{dt}{dz} + p_z\dot{z}\frac{dt}{dz} - H\frac{dt}{dz}\right] dz \\ &= \int_{z(a)}^{z(b)} \left[p_x\frac{dx}{dz} + p_y\frac{dy}{dz} - H\frac{dt}{dz} + p_z\right] dz\,.\end{aligned} \tag{2.20}$$

Observe that by defining a new Hamiltonian $H' = -p_z$ and new canonical momentum in z as $-H$, the previous result from the one-dimensional case is replicated. H' can be computed from (2.13) by solving for p_z:

$$H' = -p_z = -\sqrt{\frac{(E - q\phi)}{c^2} - (p_x - qA_x)^2 - (p_y - qA_y)^2 - m^2c^2} - qA_z\,, \tag{2.21}$$

where we have used that $H = E$. From now on the previous Hamiltonian H will only be referred to as E and the newly derived H' will be referred to as the Hamiltonian H. This is done since the original formulation of the Hamiltonian is unwieldy and offers no utility other than as an expression for the total energy of the particle.

The new representation of the Hamiltonian system in (2.21) has replaced time as the free variable with z, and as a result solving Hamilton's equations will now produce a solution for the coordinate-momentum pairs (x, p_x), (y, p_y) and $(t, -E)$. This means that for any position along the beam, the transverse canonical momenta and position, the time of arrival and energy, are tracked. In this formulation of the Hamiltonian system, we have that

$$\frac{d(-E)}{dz} = \frac{\partial H}{\partial t}\,, \tag{2.22}$$

implying that a time-invariant system will result in the conservation of the total energy of the particle. For stable conditions in the accelerator, almost all fields are time-invariant, and therefore so is the Hamiltonian. If the Hamiltonian is the function determining the dynamics of the system then, trivially, the transverse variables (x, p_x) and (y, p_y) can be solved independently of time. This implies that if the subject of analysis is the transverse variables and all the electromagnetic fields are time-invariant, then a Hamiltonian treatment will reduce the dimensionality of the problem by one.

Hamiltionian for a Curved Reference System

The Hamiltonian just derived was for the case of a straight reference system, i.e. a linear accelerator. This thesis will however work with curved reference systems as HL–LHC is a circular accelerator, wherefore a different Hamiltonian has to be derived for curved reference systems. The idea is the same but deriving it in detail would take more effort than it is worth, as such it is provided below in the form that will be used for the remainder of the thesis. It is given as

$$H = \frac{\delta}{\beta_0} - (1+hx)\sqrt{\left(\delta + \frac{1}{\beta_0} - \frac{q\phi}{cp_0}\right)^2 - (p_x - a_x)^2 - (p_y - a_y)^2 - \frac{1}{\beta_0^2\gamma_0^2}} - (1+hx)a_s\,, \tag{2.23}$$

where

$$
\begin{aligned}
z &= \frac{s}{\beta_0} - ct\,, \\
\delta &= \frac{E}{cp_0} - \frac{1}{\beta_0}\,, \\
p_x &= \frac{\beta_x \gamma c + qA_x}{p_0}\,, \\
p_y &= \frac{\beta_y \gamma c + qA_y}{p_0}\,, \\
a_x &= \frac{A_x q}{p_0}\,, \\
a_y &= \frac{A_y q}{p_0}\,, \\
a_s &= \frac{A_s q}{p_0}\,.
\end{aligned}
\tag{2.24}
$$

Quantities indexed by a zero refer to a *reference particle*. The reference particle has the momentum p_0, the momentum the accelerator was designed for, and follows the *reference trajectory* with radius ρ (and curvature $h = 1/\rho$), the trajectory the reference particle was designed to follow. The reference system is shown in Figure 2.1

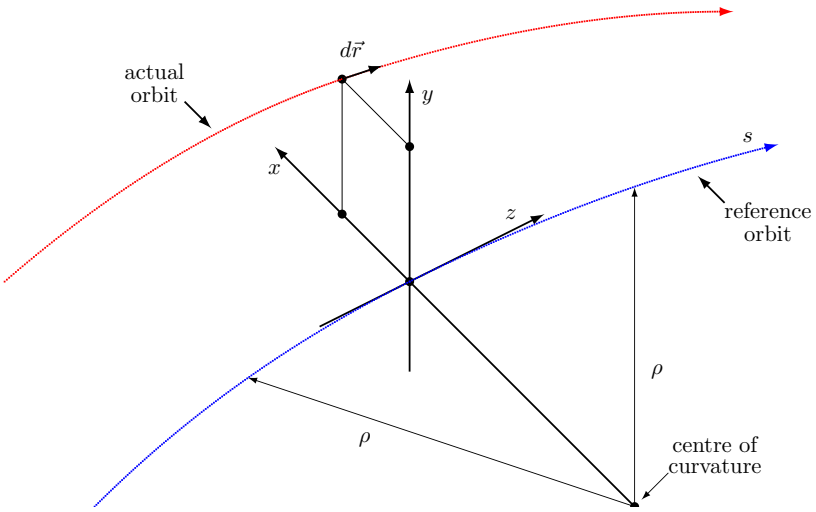


Figure 2.1: Graphical representation of the coordinate system to be used for this thesis [4].

The reference trajectory is parameterized by s, the free variable for this Hamiltonian. Everything indexed by an s refers to quantities pointing in the tangential direction of the reference orbit. What was previously the coordinate-momentum pair $(t, -E)$ has been transformed into (z, δ) and the canonical momenta have been scaled by the reference momentum. The reason for this change is numerics. Solving the Hamiltonian system exactly for a full accelerator is rarely, if ever, an option. It is therefore necessary to employ numerical methods for the simulations, which encourages conditioning of the dynamic variables. For example,

$(z, \delta) = (0, 0)$ for the reference particle which can be put in contrast to $(t, -E)$. The time of arrival for positions along the accelerator will increase indefinitely and so tracking the relative longitudinal distance of particles would involve taking differences between increasingly large numbers. Instead tracking quantities in relation to the reference particle is a valid approach to conditioning the variables, as great deviation from the reference particle will nevertheless result in lost particles.

2.1.3 Linear Transfer Maps

As was previously hinted at, the Hamiltonian in (2.23) will rarely translate into explicit solutions for a full circular accelerator. Alternatives will have to be devised, and from experience in the field of accelerator physics (and applied physics in general), a linear approximation is sufficient for many applications. The approach taken here is the canonical one. Summarized, the following is done per element in the accelerator:

1. Use the Hamiltonian in (2.23) where the electromagnetic fields and the curvature of the element have been entered.

2. Do an appropriate Taylor approximation and truncate at the third order (i.e. a second-order approximation of the Hamiltonian).

3. Setup Hamilton's equations, which will be linear relations.

4. Express the mapping through the element as a matrix.

If this is done for every element, then effectively one has retrieved a first order approximation of the entire accelerator. By applying a sequence of matrices to a given configuration of dynamical variables corresponding to a certain state of a particle in the accelerator, it is possible to compute its state further down or back in the accelerator. In this section linear approximations to the *drift space*, *bending magnet*, *orbit corrector dipole* and *quadrupole* will be computed. These elements are considered because:

1. They can all be well-approximated by a linear approximation.

2. The drift space is the simplest element there is, hence a good starting point.

3. Bending magnets and quadrupoles are the main source of closed orbit perturbation in an accelerator like HL–LHC.

4. Orbit corrector dipoles are the main tool for controlled manipulation of the closed orbit.

Drift Space

A drift space in an accelerator is simply free space, a region of the accelerator where no electromagnetic fields are in place. It makes little sense to have curved drift spaces as the

beam experiences no bending force inside them, hence $h = 0$. The Hamiltonian inside such an element is described by removing all electromagnetic fields from (2.23) producing

$$H = \frac{\delta}{\beta_0} - \sqrt{\left(\delta + \frac{1}{\beta_0}\right)^2 - p_x^2 - p_y^2 - \frac{1}{\beta_0^2\gamma_0^2}}\,. \tag{2.25}$$

This is the exact Hamiltonian inside a drift space. In this particular case there is also an exact closed solution, but for the sake of methodology a second order approximation of the Hamiltonian is derived. Note that the Hamiltonian is independent of z, hence δ must be constant. This implies that for a linear approximation the transverse dynamical variables are the only ones considered. The approximation is performed as follows:

$$\begin{aligned} H &= \frac{\delta}{\beta_0} - \sqrt{\left(\delta + \frac{1}{\beta_0}\right)^2 - p_x^2 - p_y^2 - \frac{1}{\beta_0^2\gamma_0^2}} \\ &= \frac{\delta}{\beta_0} - C\sqrt{1 - \frac{p_x^2 + p_y^2}{C^2}} \\ &\approx \frac{\delta}{\beta_0} - C + \frac{p_x^2 + p_y^2}{2C}\,, \end{aligned} \tag{2.26}$$

where $C = \sqrt{1 + \frac{2\delta}{\beta_0} + \delta^2}$, (recall that $\gamma^2 = \frac{1}{1-\beta^2}$) and it has been assumed $\frac{p_x^2+p_y^2}{C^2} \ll 1$. This assumption is valid only if particles do not deviate significantly in angle from the reference particle, which while not immediately obvious, is reasonable in practice. Further expanding C yields the final Hamiltonian:

$$H_2 = \frac{p_x^2}{2} + \frac{p_y^2}{2} + \frac{\delta^2}{2\beta_0^2\gamma_0^2}\,, \tag{2.27}$$

where the constant has been left out as it does not impact the dynamics. To produce the solution, Hamilton's equations can be applied to (2.27) for each set of canonical coordinates.

$$\begin{aligned} \frac{dx}{ds} &= \frac{\partial H_2}{\partial p_x} = p_x\,, \\ \frac{dp_x}{ds} &= -\frac{\partial H_2}{\partial x} = 0\,, \\ \frac{dy}{ds} &= \frac{\partial H_2}{\partial p_y} = p_y\,, \\ \frac{dp_y}{ds} &= -\frac{\partial H_2}{\partial y} = 0\,, \\ \frac{dz}{ds} &= \frac{\partial H_2}{\partial \delta} = \frac{\delta}{\beta_0^2\gamma_0^2}\,, \\ \frac{d\delta}{ds} &= -\frac{\partial H_2}{\partial z} = 0\,. \end{aligned} \tag{2.28}$$

It goes without saying that the equations in (2.28) are an approximation, which defines the dynamics inside a drift space in the first order. Given initial conditions and the length of

a drift space, the means of computing the evolution of any particle passing through it are provided. Using the notation

$$\mathbf{u}(s) = \begin{bmatrix} x(s) & p_x(s) & y(s) & p_y(s) & z(s) & \delta(s) \end{bmatrix}^\mathsf{T} , \tag{2.29}$$

for describing the state of a particle, the following matrix formalism can be employed:

$$\mathbf{u}(s_0 + L) = \mathbf{T}_{\text{drift}}\mathbf{u}(s_0) , \tag{2.30}$$

where s_0 denotes the start of a drift space, L is the length of it and $\mathbf{T}_{\text{drift}}$ is defined as:

$$\mathbf{T}_{\text{drift}} = \begin{bmatrix} 1 & L & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & L & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{L}{\beta_0^2\gamma_0^2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{2.31}$$

The first linear approximation of an element has been derived. The derivation for subsequent elements is similar but the bookkeeping will be kept less verbose.

Bending Magnet

Bending magnets are curved dipoles tasked with bending the reference trajectory by using a constant magnetic field perpendicular to the bending plane, e.g. for horizontal bending we have

$$\mathbf{B} = \begin{bmatrix} B_x & B_y & B_s \end{bmatrix}^\mathsf{T} = \begin{bmatrix} 0 & B_0 & 0 \end{bmatrix}^\mathsf{T} . \tag{2.32}$$

It is however not the magnetic field but the vector potential that enters the Hamiltonian, hence the magnetic field has to be expressed as a function of this quantity. Recall from (2.6), the magnetic field is the curl of the vector potential. In Cartesian coordinates this would be a straightforward computation, but as the coordinate system of choice for a bending magnet is curvilinear, i.e. $h \neq 0$, the curl of the vector potential is non-trivial. For the sake of brevity, a vector potential is directly provided which is valid in our coordinate system and computationally advantageous:

$$\mathbf{a} = \begin{bmatrix} 0 & 0 & -k_0 x + \frac{k_0 h x^2}{2(1+hx)} \end{bmatrix}^\mathsf{T} . \tag{2.33}$$

where $k_0 = \frac{q}{p_0}B_0$ is the *normalized field strength*. Inserting (2.33) into the Hamiltonian (2.23), we retrieve

$$H = \frac{\delta}{\beta_0} - (1+hx)\sqrt{\left(\delta + \frac{1}{\beta_0}\right)^2 - p_x^2 - p_y^2 - \frac{1}{\beta_0^2\gamma_0^2}} + k_0(1+hx)x - k_0 h\frac{x^2}{2} . \tag{2.34}$$

Using the same technique as before for approximating the Hamiltonian yields

$$H_2 = \frac{p_x^2}{2} + \frac{p_y^2}{2} + \frac{\delta^2}{2\beta_0^2\gamma_0^2} + \left\{k_0 - h\left(1 + \frac{\delta}{\beta_0}\right)\right\}x + k_0 h\frac{x^2}{2} . \tag{2.35}$$

Again the partial derivatives of H_2 are computed and expressed in a linear relation, but this time around there is an added constant term:

$$\mathbf{u}(s_0 + L) = \mathrm{T}_{\mathrm{bend}}\mathbf{u}(s_0) + \mathbf{d}_{bend}\,, \tag{2.36}$$

where

$$\mathrm{T}_{\mathrm{bend}} = \begin{bmatrix} 1 & L & 0 & 0 & 0 & 0 \\ -hk_0L & 1 & 0 & 0 & 0 & \frac{hL}{\beta_0} \\ 0 & 0 & 1 & L & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -\frac{hL}{\beta_0} & 0 & 0 & 0 & 1 & \frac{L}{\beta_0^2\gamma_0^2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{d}_{bend} = \begin{bmatrix} 0 \\ (h-k_0)L \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.37}$$

Note that depending on the value of the $h - k_0$, a particle entering on the reference trajectory could leave the bending magnet off the reference trajectory. However, by definition, the reference trajectory is the trajectory traversed by the reference particle, which enforces $h = k_0$ for a correctly designed and perfect accelerator. In such a case, the mapping simplifies to $\mathrm{T}_{\mathrm{bend}}$.

Orbit Corrector

The primary mean of changing the closed orbit in a controlled manner is through *orbit corrector dipoles*. An orbit corrector dipole is effectively a straight bending magnet with a variable k_0. It follows that a horizontal orbit corrector can be retrieved by taking the derived map of the bending magnet from (2.37) with $h = 0$. The resulting mapping is:

$$\mathbf{u}(s_0 + L) = \mathrm{T}_{\mathrm{hcorr}}\mathbf{u}(s_0) + \mathbf{d}_{hcorr}\,, \tag{2.38}$$

where

$$\mathrm{T}_{\mathrm{hcorr}} = \mathrm{T}_{\mathrm{drift}}\,, \quad \mathbf{d}_{hcorr} = \begin{bmatrix} 0 & \theta & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}}. \tag{2.39}$$

and $\theta = k_0L$. Since orbit correctors are only powered for intentional manipulation of the orbit, $\theta = 0$ when analyzing the stability of an accelerator, hence they are equivalent to drift spaces in a naive operational scenario. The analogous derivation holds for a vertical orbit corrector, where the only difference in the resulting mapping is that the orbit kick applies to the vertical plane.

Quadrupole

In accelerators quadrupoles are the primary tool for transverse focusing of the beam. They have a magnetic field

$$\mathbf{b} = \frac{q}{p_0}\mathbf{B} = \begin{bmatrix} k_1 y & k_1 x & 0 \end{bmatrix}^{\mathsf{T}}, \tag{2.40}$$

where by convention $k_1 = \frac{q}{p_0}B_1$. A vector potential achieving this is

$$\mathbf{a} = \begin{bmatrix} 0 & 0 & -\frac{k_1}{2}(x^2 - y^2) \end{bmatrix}^{\mathsf{T}}. \tag{2.41}$$

Inserting this vector potential into (2.23) with $h = 0$ (quadrupoles are usually straight elements) and no scalar potential yields the Hamiltonian for a straight quadrupole:

$$H = \frac{\delta}{\beta_0} - \sqrt{\left(\delta + \frac{1}{\beta_0}\right)^2 - p_x^2 - p_y^2 - \frac{1}{\beta_0^2\gamma_0^2}} - \frac{k_1}{2}(x^2 - y^2)\,. \tag{2.42}$$

Repeating the approximation procedure gives the second-order Hamiltonian

$$H_2 = \frac{p_x^2}{2} + \frac{p_y^2}{2} + \frac{\delta^2}{2\beta_0^2\gamma_0^2} + \frac{k_1}{2}(x^2 - y^2)\,. \tag{2.43}$$

Due to the interdependence in each pair of transverse variables, the expression is slightly more complicated than before, but nevertheless linear. In matrix formalism

$$\mathbf{u}(s_0 + L) = \mathrm{T}_{\mathrm{quad}}\mathbf{u}(s_0)\,, \tag{2.44}$$

where

$$\mathrm{T}_{\mathrm{quad}} = \begin{bmatrix} \cos(\omega L) & \sin(\omega L)/\omega & 0 & 0 & 0 & 0 \\ -\omega\sin(\omega L) & \cos(\omega L) & 0 & 0 & 0 & 0 \\ 0 & 0 & \cosh(\omega L) & \sinh(\omega L)/\omega & 0 & 0 \\ 0 & 0 & \omega\sinh(\omega L) & \cosh(\omega L) & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{L}{\beta_0^2\gamma_0^2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \omega = \sqrt{k_1}\,. \tag{2.45}$$

Assuming $k_1 > 0$ and realistic values values for k_1, L, any particle entering the quadrupole with a horizontal orbit will experience a kick to bring it back towards zero horizontal orbit, whereas a vertical orbit will be aggravate the situation by giving a kick *increasing* the deviation. Summarized, for $k_1 > 0$ the quadrupole is horizontally *focusing* and vertically *defocusing*, with the inverse for $k_1 < 0$.

Often it is sufficient to treat the quadrupole as a *thin lens*. The concept is the same as in optics: an incoming beam experiences a change in angle linearly proportional to the offset from the optical axis, and no change in position. The crux of a thin lens is essentially that it is so thin that any beam passing through it does not experience any change in transverse position while inside the lens. Formulaically, this is achieved by keeping $k_1 L$ constant while letting $L \to 0$, and this gives the thin-lens transformation matrix

$$\mathrm{T}_{\mathrm{quad(thin)}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -k_1 L & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & k_1 L & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{L}{\beta_0^2\gamma_0^2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.46}$$

where the focal length is $f = 1/(k_1 L)$. Observe that the mapping is valid for $k_1 < 0$, which inverts the effects in the transverse planes. In the first order, a thin-lens treatment is often sufficiently accurate, and it also has the benefit of providing a simple way of visualizing the transverse dynamics using linear optics.

2.1.4 Beam Dynamics

Having derived linear transformations for the essential building blocks of a circular accelerator, the next object of attention is to study the structures where such elements are put in sequence with a focus on the transverse plane. We make the following assumptions:

- All elements are without errors.
- All orbit correctors are powered off.
- The beam is monochromatic, i.e. all particles have the same energy as the reference particle, $\delta = 0$.

This is done to make all mappings strictly linear with no added constant, and the dynamics in the transverse planes decoupled from the dynamics in the longitudinal plane.

As discussed previously, any quadrupole is focusing in one and defocusing in the other transverse plane, which brings about the idea of the simplest structure for focusing a beam: the FODO lattice. The FODO lattice is a lattice where quadrupoles are alternatingly focusing and defocusing with some constant spacing between them, i.e. drift spaces. In a simple periodic FODO lattice, the transformation matrix for one cell can be written as:

$$\mathbf{T}_{\text{FODO}} = \mathbf{T}_{\text{drift}}\mathbf{T}_{\text{D}}\mathbf{T}_{\text{drift}}\mathbf{T}_{\text{F}}\,, \tag{2.47}$$

where $\mathbf{T}_{\text{F}}$ and $\mathbf{T}_{\text{D}}$ correspond to focusing and defocusing quadrupoles respectively with focal lengths $\pm f$ separated by drift spaces of length L. The decoupled dynamics implies that $\mathbf{T}_{\text{FODO}}$ is a block diagonal matrix as per:

$$\mathbf{T}_{\text{FODO}} = \begin{bmatrix} T_x & 0 & 0 \\ 0 & T_y & 0 \\ 0 & 0 & T_s \end{bmatrix}. \tag{2.48}$$

with transverse transfer matrices

$$\begin{aligned} \mathbf{T}_{\text{x}} &= \begin{bmatrix} 1 - \frac{L}{f}(1 + \frac{L}{f}) & \frac{L}{f}(2f + 1) \\ -\frac{L}{f^2} & 1 + \frac{L}{f} \end{bmatrix}, \\ \mathbf{T}_{\text{y}} &= \begin{bmatrix} 1 + \frac{L}{f}(1 - \frac{L}{f}) & \frac{L}{f}(2f - 1) \\ -\frac{L}{f^2} & 1 - \frac{L}{f} \end{bmatrix}, \end{aligned} \tag{2.49}$$

Note that the physical interpretation of applying the FODO transformation matrix is the transportation of the particle through one cell in a periodic machine. For the FODO matrix given in (2.47), repeatedly applying it to a state would produce a sequence of states measured at the entrance of the focusing quadrupole as the particle passes through multiple cells. Such a plot is shown in Figure 2.2 where the horizontal state is tracked through a FODO lattice each time it passes through the defocusing quarupole. Notice how for any starting point, the phase space plot assumes the shape of an ellipse, all with the same shape. This is a result built into how the problem was treated. To get into why this structure arises is technical but important, and the next subject.

First, the concept of *symplecticity* will be introduced. A $2n \times 2n$, n positive integer, matrix $\mathbf{A}$ is said to be symplectic if it satisfies

$$\mathbf{A}^{\intercal}\mathbf{S}\mathbf{A} = \mathbf{S}\,, \tag{2.50}$$

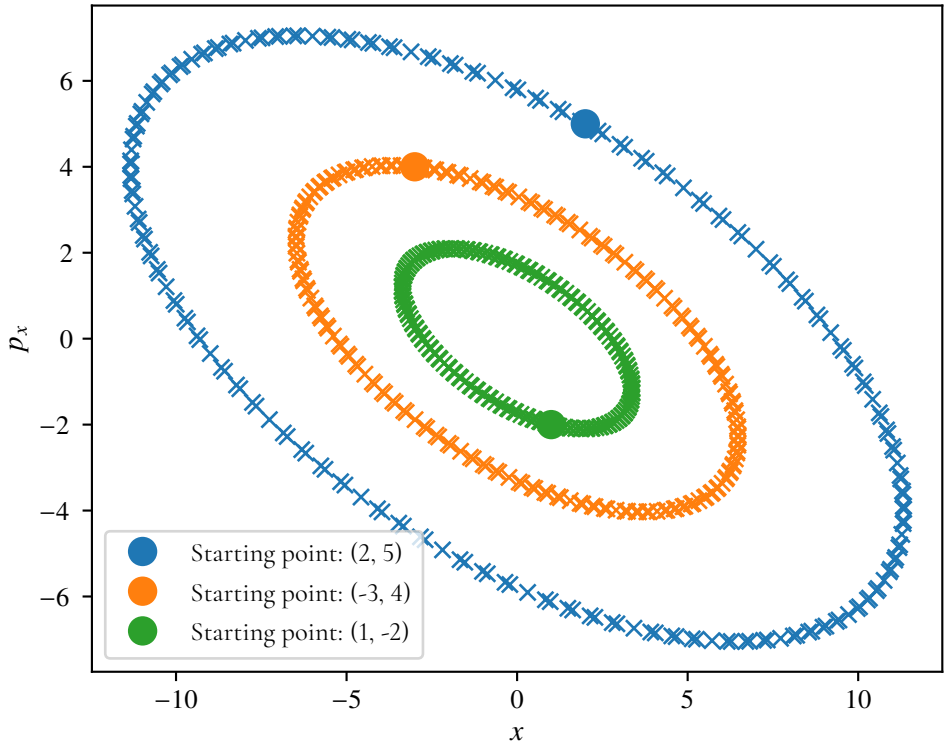


Figure 2.2: Phase plots for different initial phase space vectors in a FODO lattice where $L = 1$, $f = \sqrt{2}$. Each cross corresponds to the horizontal phase space vector measured after having passed through the defocusing quadrupole in such a periodic FODO lattice.

where $\mathbf{S}$ is the skew-symmetric matrix defined as

$$\mathbf{S} = \underbrace{\begin{bmatrix} \mathbf{R} & & \\ & \ddots & \\ & & \mathbf{R} \end{bmatrix}}_{n}, \quad \mathbf{R} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{2.51}$$

Any solution derived from Hamilton's equations is symplectic, and therefore so are all transfer matrices derived so far. This motivates why the approach taken for deriving each matrix was to approximate the Hamiltonian to second order, differentiate it and then solve Hamilton's equations. An alternative (and at first glance, just as good) approach would have been to differentiate the Hamiltonian, approximate to the first order and then solve the system. However, symplecticity is only guaranteed if Hamilton's equations are solved exactly, independently of what the Hamiltonian looks like. Another important observation is that any (valid) product of symplectic matrices is also symplectic. This can be proven as follows. Given two symplectic matrices, $\mathbf{A}_1$ and $\mathbf{A}_2$, the following holds:

$$(\mathbf{A}_1\mathbf{A}_2)^\intercal \mathbf{S} \mathbf{A}_1\mathbf{A}_2 = \mathbf{A}_2^\intercal \mathbf{A}_1^\intercal \mathbf{S} \mathbf{A}_1\mathbf{A}_2 = \mathbf{A}_2^\intercal \mathbf{S} \mathbf{A}_2 = \mathbf{S}\,. \tag{2.52}$$

This implies that any sequence of elements in the accelerator for which there are linear transformations (derived by approximating the Hamiltonian to second order), will in turn produce a map that is symplectic.

Importantly, symplecticity enforces constraints on the system. Consider a symplectic transfer map $\mathbf{T}$ for a cell in a periodic lattice. The cell could come from a FODO lattice, but could also be the product of all element transfer maps for one lap in any circular accelerator. We consider the horizontal part of the transfer map $\mathbf{T}_\mathrm{x}$, but the same derivation would hold for $\mathbf{T}_\mathrm{y}$. $\mathbf{T}_\mathrm{x}$ can be parameterized as follows [6]:

$$\begin{aligned}\mathbf{T}_\mathrm{x} &= \mathbf{I}\cos(2\pi Q_x) + \mathbf{R}\mathbf{A}_\mathrm{x}\sin(2\pi Q_x) \\ &= \begin{bmatrix} \cos(2\pi Q_x) + \alpha_x \sin(2\pi Q_x) & \beta_x \sin(2\pi Q_x) \\ -\gamma_x \sin(2\pi Q_x) & \cos(2\pi Q_x) - \alpha_x \sin(2\pi Q_x) \end{bmatrix},\end{aligned} \tag{2.53}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{R}$ is the skew-symmetric matrix from (2.51) and $\mathbf{A}_\mathrm{x}$ is defined as:

$$\mathbf{A}_\mathrm{x} = \begin{bmatrix} \gamma_x & \alpha_x \\ \alpha_x & \beta_x \end{bmatrix}. \tag{2.54}$$

This representation of $\mathbf{T}_\mathrm{x}$ has four free paramaters: Q_x, α_x, β_x and γ_x. However, recall that $\mathbf{T}_\mathrm{x}$ is symplectic. This is a matrix relation which here translates to

$$\beta_x \gamma_x - \alpha_x^2 = 1\,. \tag{2.55}$$

α, β and γ are known as the *Courant-Snyder parameters* or, as they will here be referred to, the *Twiss parameters*, and Q is the *tune*. Note that

$$\mathrm{Tr}(\mathbf{T}_\mathrm{x}) = 2\cos(2\pi Q_x)\,. \tag{2.56}$$

Using (2.56), one can efficiently compute the Twiss parameters and tune given the transfer map $\mathbf{T}_\mathrm{x}$ of a cell in the periodic lattice. Note that the Twiss parameters derived from such a cell is a function of the internal position in the cell, i.e. from which position one defines the start of the cell. Let the horizontal transfer map for a cell be

$$\mathbf{T} = \mathbf{T}_2\mathbf{T}_1\,, \tag{2.57}$$

where $\mathbf{T}_1$ and $\mathbf{T}_2$ are both products of an arbitrary number of element transfer matrices. Note that the start of the cell is the entrance of the first element in $\mathbf{T}_1$, analogous to the exit of the last element in $\mathbf{T}_2$. A different description of the same periodic lattice can be constructed by choosing a different cell:

$$\mathbf{T}' = \mathbf{T}_1\mathbf{T}_2 = \mathbf{T}_1\mathbf{T}\mathbf{T}_1{}^{-1}\,, \tag{2.58}$$

where $\mathbf{T}'$ instead starts with the entrance of the first element in $\mathbf{T}_2$. Note that

$$\mathrm{Tr}(\mathbf{T}) = \mathrm{Tr}(\mathbf{T}') = 2\cos(2\pi Q_x)\,, \tag{2.59}$$

since (square) matrix multiplication commutes under the trace operation. The tune is thus independent of the position in the cell.

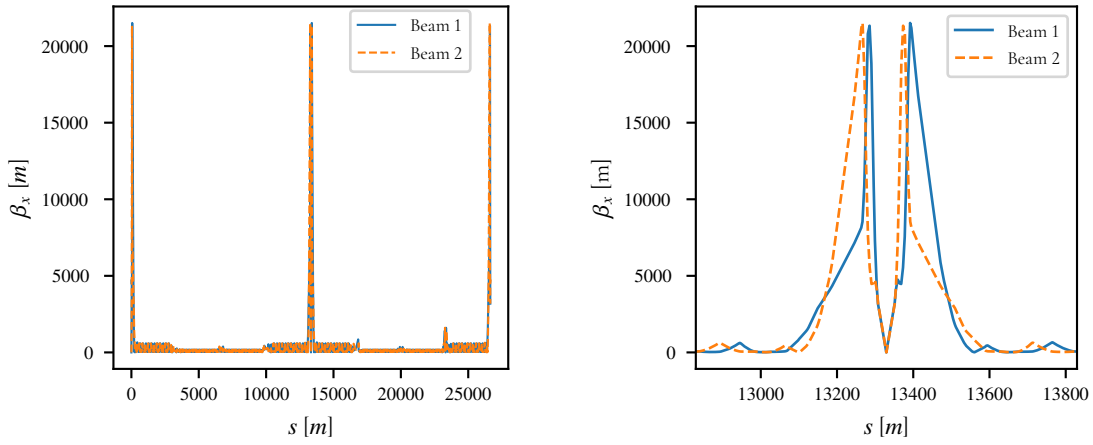
For the Twiss parameters, we have:

$$\begin{aligned} \mathbf{T} &= \mathbf{I}\cos(2\pi Q_x) + \mathbf{R}\mathbf{A}_\mathrm{x}\sin(2\pi Q_x)\,, \\ \mathbf{T}' &= \mathbf{I}\cos(2\pi Q_x) + \mathbf{R}\mathbf{A}_\mathrm{x}{}'\sin(2\pi Q_x) = \mathbf{T}_1\mathbf{T}\mathbf{T}_1{}^{-1} \\ &= \mathbf{I}\cos(2\pi Q_x) + \mathbf{T}_1\mathbf{R}\mathbf{A}_\mathrm{x}\mathbf{T}_1{}^{-1}\sin(2\pi Q_x)\,, \\ \implies \mathbf{R}\mathbf{A}_\mathrm{x}{}' &= \mathbf{T}_1\mathbf{R}\mathbf{A}_\mathrm{x}\mathbf{T}_1{}^{-1}\,, \\ \implies \mathbf{A}_\mathrm{x}{}' &= \mathbf{R}^{-1}\mathbf{T}_1\mathbf{R}\mathbf{A}_\mathrm{x}\mathbf{T}_1{}^{-1} = \mathbf{R}^\intercal\mathbf{T}_1\mathbf{R}\mathbf{A}_\mathrm{x}\mathbf{T}_1{}^{-1} = \\ &= (\mathbf{T}_1{}^{-1})^\intercal\mathbf{A}_\mathrm{x}\mathbf{T}_1{}^{-1}\,, \end{aligned} \tag{2.60}$$

where the last two steps make use of $\mathbf{R}^\intercal = \mathbf{R}^{-1}$ and a corollary to the symplecticity condition: $\mathbf{R}^\intercal\mathbf{T}\mathbf{R} = (\mathbf{T}^{-1})^\intercal$. Summarizing, the starting position for a given cell affects the Twiss parameters, but the tune stays the same. To make this statement more rigorous, let s_0, s_1 be longitudinal positions inside a cell where $\mathbf{T}(s_0, s_1)$ denotes the mapping from position s_0 to s_1. The Twiss parameter mapping can then in general be given as:

$$\mathrm{A_x}(s_1) = \left\{\mathbf{T}(s_0, s_1)^{-1}\right\}^\intercal \mathrm{A_x}(s_0)\mathbf{T}(s_0, s_1)^{-1}\,. \tag{2.61}$$

If the transfer map for each element of an accelerator is known, then the Twiss parameters at any position can be propagated along the accelerator using (2.61), thus providing a method for computing the Twiss parameters as a function of s. As an example, Figure 2.3 shows β_x in HL–LHC for a given *optics*, that is, for Twiss parameters generated by a certain setting of quadrupole strengths. For now, we only remark that the β-function assumes large values



(a) Full machine. **(b)** Around `IP5`.

Figure 2.3: β_x in HL–LHC shown for both beams over the full machine **(a)**, and zooomed in around `IP5` **(b)** for collision optics. $s = 0$ corresponds to `IP1`.

around interaction point 1 and 5, and relatively small values at the points themselves.

An invariant property along the accelerator is the *emittance*, ϵ_x. The emittance is defined as

$$\epsilon_x = \begin{bmatrix} x & p_x \end{bmatrix} \mathrm{A_x} \begin{bmatrix} x \\ p_x \end{bmatrix} = \mathbf{x}^\intercal \mathrm{A_x}\mathbf{x}\,. \tag{2.62}$$

To prove its constancy, let an arbitrary point $\mathbf{x}$ in horizontal phase space get mapped along the beamline via a transfer map $\mathbf{T}$. The emittance at the new position ϵ_x' thus satisfies

$$\epsilon_x' = (\mathbf{x}')^\intercal \mathrm{A_x}'\mathbf{x}' = \mathbf{x}^\intercal\mathbf{T}^\intercal\left\{(\mathbf{T}^{-1})^\intercal \mathrm{A_x}\mathbf{T}^{-1}\right\}\mathbf{T}\mathbf{x} = \mathbf{x}^\intercal \mathrm{A_x}\mathbf{x} = \epsilon_x\,, \tag{2.63}$$

Note that 2.62 defines an ellipse in phase space. This can be made more explicit by expressing it on the form:

$$\epsilon_x = \gamma_x x^2 + 2\alpha_x x p_x + \beta_x p_x^2\,, \tag{2.64}$$

which can also be graphically reppresented as in Figure 2.4. If the emittance is a constant

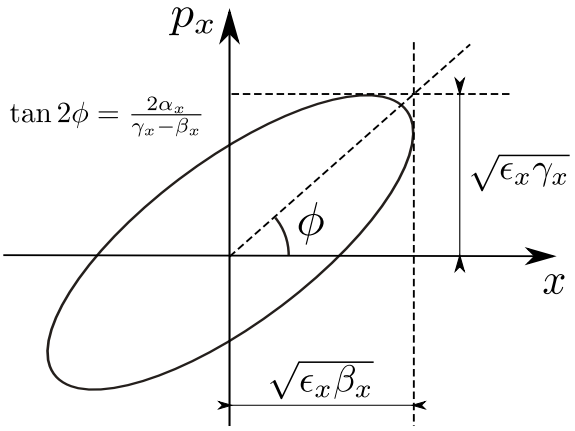


Figure 2.4: Parameterization of a phase space ellipse using the Twiss parameters.

for a particle traversing an accelerator, it then follows that a particle passing through a point multiple times in an accelerator will assume values on the ellipse defined at said point. This is the explanation for the ellipses shown in Figure 2.2.

Note that the Twiss parameters predict on which ellipse a given particle will be found on for each s, but gives no information about how a given point on a phase space ellipse translates to the next. To resolve this, we introduce the *angle variable*:

$$\tan \varphi_x = -\beta \frac{p_x}{x} - \alpha_x \,. \tag{2.65}$$

Given the angle variable, it is possible to solve for x and p_x in (2.64):

$$\begin{aligned} x &= \sqrt{\epsilon_x \beta_x} \cos \varphi_x \\ p_x &= -\sqrt{\frac{\epsilon_x}{\beta_x}} (\sin \varphi_x + \alpha_x \cos \varphi_x) \end{aligned} \tag{2.66}$$

Observe that knowing the value of φ_x as a function of s, together with the emittance and Twiss parameters, gives a complete description of a particle's trajectory. To achieve this end, we introduce the *phase advance* μ_x:

$$\mu_x(s_0, s_1) = \varphi_x(s_1) - \varphi_x(s_0) = \int_{s_0}^{s_1} \frac{ds}{\beta_x(s)} \,. \tag{2.67}$$

The phase advance is the increment in angle-variable between two points in an accelerator, and is determined solely by the β_x function. This implies that it is sufficient to know the initial point in phase space for a particle, together with the Twiss parameters for the accelerator, to determine its full evolution in phase space. This observation is made rigorous in (2.68):

$$\mathrm{T}(s_0, s_1) = \begin{bmatrix} \sqrt{\frac{\beta_x(s_1)}{\beta_x(s_0)}} \Big\{ \cos \mu_x + \alpha_x(s_0) \sin \mu_x \Big\} & \sqrt{\beta_x(s_0) \beta_x(s_1)} \sin \mu_x \\ \frac{\alpha_x(s_0) - \alpha_x(s_1)}{\sqrt{\beta_x(s_0) \beta_x(s_1)}} \cos \mu_x - \frac{1 + \alpha_x(s_0) \alpha_x(s_1)}{\sqrt{\beta_x(s_0) \beta_x(s_1)}} \sin \mu_x & \sqrt{\frac{\beta_x(s_0)}{\beta_x(s_1)}} \Big\{ \cos \mu_x - \alpha_x(s_1) \sin \mu_x \Big\} \end{bmatrix}, \tag{2.68}$$

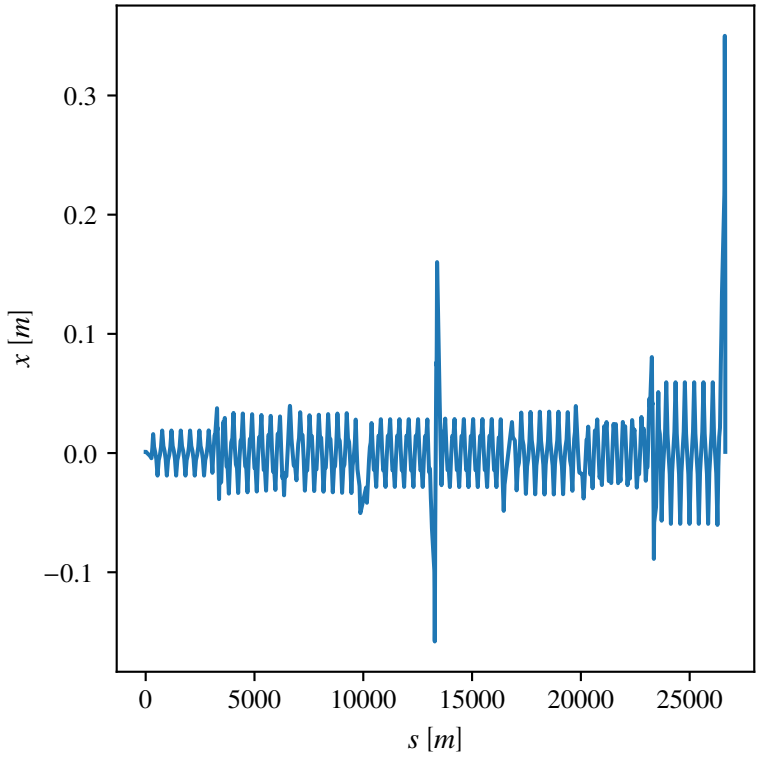


Figure 2.5: Plot of the horizontal trajectory of a particle for one turn in HL–LHC, starting at `IP1` with $x = 1.0\,\text{mm}$ and $p_x = 0.0\,\text{rad}$.

where μ_x is the phase advance between s_0 and s_1. Using this transfer mapping, Figure 2.5 shows the horizontal position of a particle for one turn in the HL–LHC. Observe that the particle oscillates, that it assumes greater values where β_x is at its greatest and that the oscillation is driven by the phase advance.

If C is the length of a periodic cell (or circular accelerator), then $\mathbf{T}(s_0, s_0 + C)$ is equal to the transfer matrix for one turn as given in (2.53). This observation leads directly to the relation between the phase advance and tune:

$$Q_x = \frac{1}{2\pi}\mu_x(0, C) \tag{2.69}$$

This also provides an interpretation for the tune: it is the number of oscillations for one turn in a periodic accelerator. In HL–LHC, the tunes are equal across the two beams and assume values:

$$\begin{aligned} Q_x &= 62.31 \\ Q_y &= 60.32 \end{aligned} \tag{2.70}$$

Interestingly, the tunes differ for the two planes. From the perspective of a single particle passing through HL–LHC, the difference in tune implies that it will perform roughly two more oscillations in the horizontal plane than in the vertical plane per turn. Note however that the fractal part of the tune is almost equal across the planes. This implies that when considering the evolution of a particle's transverse position, at an arbitrary fix location s, from one turn to the next, the oscillations will occur with almost the same frequency in the two transverse planes.

2.1.5 Closed Orbit

The notion of a *closed orbit* will be very important for future analysis. It is only well-defined for a periodic lattice, from here on taken to mean a circular accelerator with circumference C. The closed orbit is represented by the point in phase space that closes in on itself after one turn in the accelerator:

$$\mathbf{u}_{co}(s) = \mathbf{u}_{co}(s + C)\,, \tag{2.71}$$

where $\mathbf{u}_{co}$ is a state vector as defined in (2.29). Monochromaticity is still assumed, and so all dynamics remain uncoupled. The transverse dynamics remains the primary interest, and so the analysis will be performed for the horizontal plane where $\mathbf{x} = \begin{bmatrix} x & p_x \end{bmatrix}^\mathsf{T}$, but the treatment of the vertical plane remains analogous. The one-turn map for a given s is here defined as:

$$f_s^1(\mathbf{x}) = \mathrm{T}_1(s)\mathbf{x} + \mathbf{d_1}(s)\,. \tag{2.72}$$

Note how in this treatment the one-turn map has a constant $\mathbf{d_1}(s)$. This differs from the previous treatment where $\mathbf{d_1}(s)$ was zero due to the assumption of no element errors nor powered orbit correctors. When discussing the closed orbit however, the analysis is only non-trivial if $\mathbf{d_1}(s) \neq \mathbf{0}$, wherefore it is assumed to be non-zero.

The following holds for the horizontal closed orbit $\mathbf{x}_{co}$:

$$\mathbf{x}_{co}(s) = f_s^1\{\mathbf{x}_{co}(s)\} = \mathrm{T}_1(s)\mathbf{x}_{co}(s) + \mathbf{d}_1(s)\,, \tag{2.73}$$

where $\mathrm{I} - \mathrm{T}_1(s)$ must be invertible for any operational machine. If it was not, then the notion of closed orbit would not exist for the machine, which in turn would render it inoperable as the notion of stability breaks down. It follows that

$$\mathbf{x}_{co}(s) = \left\{\mathrm{I} - \mathrm{T}_1(s)\right\}^{-1}\mathbf{d}_1(s)\,. \tag{2.74}$$

In the previous analysis, $\mathbf{x}_{co}(s) = \mathbf{0}$, i.e. the closed orbit was equal to the reference orbit. To see how the non-zero closed orbit enters the results, consider the one-turn map applied to an arbitrary point in phase space $\mathbf{x}$, at an arbitrary position s:

$$\begin{aligned} f_s^1(\mathbf{x}) &= f_s^1\{\mathbf{x} - \mathbf{x}_{co}(s) + \mathbf{x}_{co}(s)\} \\ &= \mathrm{T}_1(s)\{\mathbf{x} - \mathbf{x}_{co}(s) + \mathbf{x}_{co}(s)\} + \mathbf{d}_1(s) \\ &= \mathrm{T}_1(s)\{\mathbf{x} - \mathbf{x}_{co}(s)\} + \mathbf{x}_{co}(s)\,. \end{aligned} \tag{2.75}$$

Note how the one-turn map at a position s maps any point $\mathbf{x}$ based on its location in phase space relative to the closed orbit. It follows that the n-turn map satisfies

$$f_s^n(\mathbf{x}) = \mathrm{T}_1{}^n(s)\{\mathbf{x} - \mathbf{x}_{co}(s)\} + \mathbf{x}_{co}(s)\,. \tag{2.76}$$

We recall that $\mathbf{T}_1(s)$ is the one-turn mapping for the perfect machine, and therefore it follows that when the closed orbit is non-zero, the ellipse in phase space is shifted by the closed orbit, whereas its shape, determined by $\mathbf{T}_1(s)$, remains the same. Expressed differently, the closed orbit defines the center of the dynamics whereas the dynamics themselves are determined by the Twiss parameters. This is showcased in Figure 2.6: the distance and orientation of the starting point relative to the closed orbit define the scale of the dynamics, but the shape of the orbit is pre-determined by the Twiss parameters, i.e. the optics.

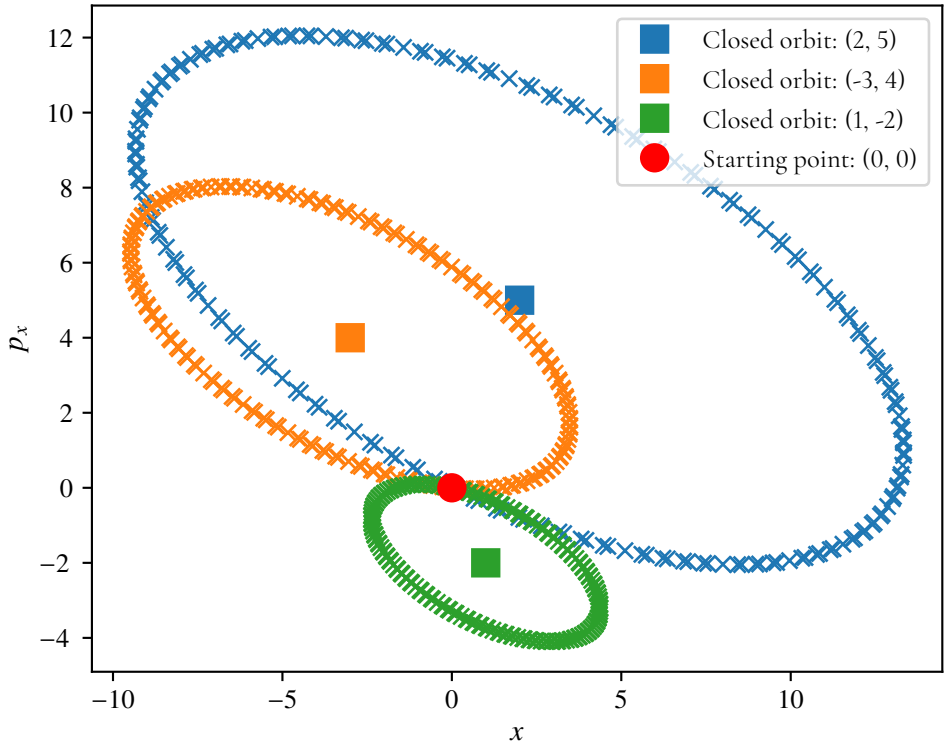


Figure 2.6: Phase-plots for a FODO lattice where $L = 1$, $f = \sqrt{2}$ and different closed orbits. Each cross in the graph corresponds to an iterate derived from the starting point $(0, 0)$ with a closed orbit coded with the same color.

Before concluding the discussion of the closed orbit as phenomenon and moving on to deriving how it can be computed efficiently, we will provide motivation for why the closed orbit plays an integral role in the operation of an accelerator. For the most part, it can be assumed that particle beams in an accelerator are enveloped by a circular pipe centered around the reference orbit, and that any particle reaching the pipe is lost. The shape of the dynamics is defined by the chosen optics and is therefore not, in the first order, impacted by machine dipolar imperfections. The closed orbit, however, is, and if the closed orbit is large enough at a point along the accelerator then the beam can hit the pipe. Such a situation is shown in Figure 2.7. In other words, if the closed orbit is not controlled, then the beam may be lost. Similarly, in a collider, such as HL–LHC, where two beams are to interact with each other, if the closed orbit is not controlled, then they might not enter collision, as the closed orbits have caused the beams to drift apart at the point of collision.

The next sections will show how the closed orbit can be perturbed by machine imperfections and how orbit correctors can have the same impact. Before that, we summarize the findings:

1. Accelerator design defines the Twiss parameters, and hence the dynamics around the closed orbit.

2. Machine errors and powered orbit correctors, together with the map of the perfect machine, determine the closed orbit.

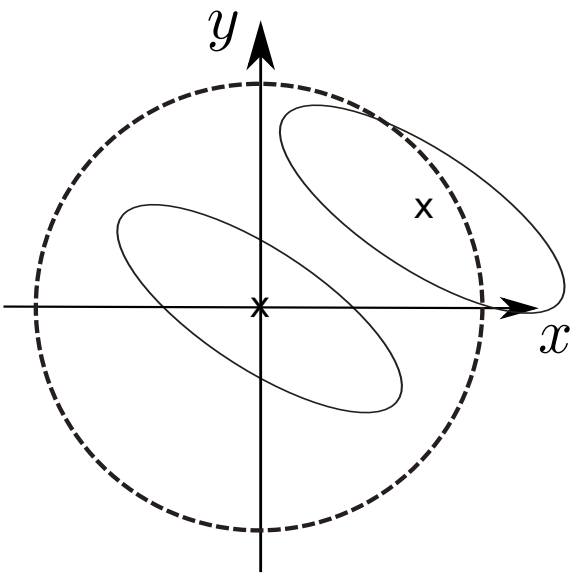


Figure 2.7: Cross-section of a beam shown inside a circular beampipe for two different closed orbits, denoted by x's.

3. Controlling the closed orbit of an accelerator is integral to its operation.

2.1.6 Derivation of the Closed Orbit

The ultimate goal is to derive a formalism for how machine errors alter the closed orbit, this part is dedicated to deriving the perturbation of the closed orbit caused by a *kick*. A kick, in this context, corresponds to a constant added to the canonical momentum for a first-order map. For example, the effect of an orbit corrector dipole is such a kick. The linear one-turn transfer map to be used is:

$$\mathrm{T}(s_0, s_0 + C) = \mathrm{T}_1(s_0) = \begin{bmatrix} \cos 2\pi Q + \alpha_0 \sin 2\pi Q & \beta_0 \sin 2\pi Q \\ -\gamma_0 \sin 2\pi Q & \cos 2\pi Q - \alpha_0 \sin 2\pi Q \end{bmatrix}, \tag{2.77}$$

where quantities evaluated at $s = s_0$ are indexed by zero and all indices indicating transverse plane have been left out. Now assume an orbit kick θ at $s = s_0$, and an otherwise perfect machine. The following then holds for the perturbed one-turn-map:

$$f^1_{s_0}(\mathbf{x}) = \mathrm{T}_1(s_0)\mathbf{x} + \begin{bmatrix} 0 \\ \theta \end{bmatrix}, \tag{2.78}$$

where the closed orbit naturally must satisfy $f^1_{s_0}\Big[\mathbf{x}_{co}(s_0)\Big] = \mathbf{x}_{co}(s_0)$. This can be computed directly yielding

$$\begin{aligned} \mathbf{x}_{co}(s_0) &= \Big\{\mathrm{T}_1(s_0) - I\Big\}^{-1} \begin{bmatrix} 0 \\ -\theta \end{bmatrix} \\ &= \frac{1}{2(1-\cos 2\pi Q)} \begin{bmatrix} \cos 2\pi Q - \alpha_0 \sin 2\pi Q - 1 & -\beta_0 \sin 2\pi Q \\ \gamma_0 \sin 2\pi Q & \cos 2\pi Q + \alpha_0 \sin 2\pi Q - 1 \end{bmatrix} \begin{bmatrix} 0 \\ -\theta \end{bmatrix} \\ &= \frac{\theta}{2(1-\cos 2\pi Q)} \begin{bmatrix} \beta_0 \sin 2\pi Q \\ 1 - \cos 2\pi Q - \alpha_0 \sin 2\pi Q \end{bmatrix}. \end{aligned}$$

This can be further simplified using some trigonometry:

$$\begin{aligned} x_{co}(s_0) &= \frac{\theta\beta_0 \sin 2\pi Q}{2(1-\cos 2\pi Q)} = \frac{2\theta\beta_0 \sin \pi Q \cos \pi Q}{4\sin^2 \pi Q} \\ &= \frac{\theta\beta_0 \cos \pi Q}{2 \sin \pi Q}, \\ p_{x,co}(s_0) &= \frac{\theta}{2(1-\cos 2\pi Q)}\Big(1 - \cos 2\pi Q - \alpha_0 \sin 2\pi Q\Big) \\ &= \frac{\theta}{4\sin^2 \pi Q}\Big(2\sin^2 \pi Q - 2\alpha_0 \sin \pi Q \cos \pi Q\Big) \\ &= \frac{\theta}{2 \sin \pi Q}\Big(\sin \pi Q - \alpha_0 \cos \pi Q\Big). \end{aligned}$$

At this point the closed orbit at the position of the kick has been computed. It is possible to compute the closed orbit as a function of s by applying the transfer map $\mathbf{T}(s_0, s)$:

$$\begin{aligned} \mathbf{x}_{co}(s) &= \mathbf{T}(s_0, s)\mathbf{x}_{co}(s_0), \\ x_{co}(s) &= \frac{\theta}{2\sin \pi Q}\Big\{\sqrt{\beta_0\beta_s}(\cos\mu + \alpha_0 \sin\mu)\cos \pi Q + \sqrt{\beta_0\beta_s}(\sin \pi Q - \alpha_0 \cos \pi Q)\sin\mu\Big\} \\ &= \frac{\theta\sqrt{\beta_0\beta_s}}{2\sin \pi Q}\Big(\cos\mu\cos \pi Q + \sin\mu \sin \pi Q\Big) \\ &= \frac{\theta\sqrt{\beta_0\beta_s}}{2\sin \pi Q}\cos(\pi Q - \mu), \\ p_{x,co}(s) &= \frac{\theta}{2\sin \pi Q}\sqrt{\frac{\beta_0}{\beta_s}}\Big\{\cos\mu \sin \pi Q - \cos \pi Q \sin\mu - \alpha_s(\cos\mu\cos \pi Q + \sin\mu \sin \pi Q)\Big\} \\ &= \frac{\theta}{2\sin \pi Q}\sqrt{\frac{\beta_0}{\beta_s}}\Big\{\sin(\pi Q - \mu) - \alpha_s \cos(\pi Q + \mu)\Big\} \\ &= \frac{\theta}{2\sin \pi Q}\sqrt{\frac{\beta_0}{\beta_s}}\sin(\pi Q - \mu) - \frac{\alpha_s}{\beta_s}x_{co}(s), \end{aligned}$$

where μ is the phase advance from s_0 to s. The answer is thus:

$$\begin{aligned} x_{co}(s) &= \frac{\theta\sqrt{\beta_0\beta_s}}{2\sin \pi Q}\cos(\pi Q - \mu), \\ p_{x,co}(s) &= \frac{\theta}{2\sin \pi Q}\sqrt{\frac{\beta_0}{\beta_s}}\sin(\pi Q - \mu) - \frac{\alpha_s}{\beta_s}x_{co}(s). \end{aligned} \tag{2.79}$$

The key observations are that:

1. The closed orbit caused by an orbit kick can be expressed using only the Twiss parameters, phase advance and the kick strength.
2. The larger the β-function is at the source of the kick, the greater the impact.

3. The induced closed orbit is linear in the orbit kick.

Recalling the plot of β_x in Figure 2.3, it follows from the second point that the areas most sensitive to kicks for HL–LHC are around `IP1` and `IP5`.

The last point is however especially poignant: given a distribution of kicks along the accelerator, the resulting aggregate closed orbit is the sum of the individual kicks' contribution to the closed orbit. Next it will be shown that machine errors give rise to kicks which are linearly proportional to the errors, effectively extending the previous statement to distributions over errors. This implies that the closed orbit perturbation caused by orbit correctors and machine errors can be superimposed, the underpinning of the matrix formalism derived in Chapter 3.

2.1.7 Machine Errors to Orbit Kicks

It should come as no surprise at this point that machine errors have an impact on the closed orbit. It is however not obvious that the most pronounced machine errors (from the perspective of transverse beam dynamics) can be well-approximated by a linear treatment. The underlying idea is simple: if a given machine error can be perceived as effectively inducing an orbit kick θ at a given s, then it can be treated linearly as per (2.79). From this perspective, the task is to map different machine errors to their corresponding orbit kicks. Below follows a list of the errors that will be considered:

- Transverse misalignment: Δx, Δy
- Rotational misalignment: $\Delta\psi$
- Field error: Δk_0, Δk_1

where positive rotation is defined by the right-hand rule around the longitudinal axis using the coordinate system in Figure 2.1. All these errors are to be analyzed for bending magnets and quadrupoles. In the end, there will be a mapping from these errors to their orbit kick and hence to their effect on the closed orbit, meaning that for a given machine (as defined by its errors) it is possible to compute the aggregate closed orbit perturbation.

Bending Magnet Errors

For a bending magnet, the analysis is straightforward. First, let us consider the case of transverse misalignment. The magnetic field inside a bending magnet is uniform. Performing a transverse shift will not impact the magnetic field at any position inside the bending magnet and therefore any particle passing through it will not experience any difference:

$$\begin{aligned} \theta^{d,x}_{\Delta x} &= \theta^{d,x}_{\Delta y} = 0\,, \\ \theta^{d,y}_{\Delta x} &= \theta^{d,y}_{\Delta y} = 0\,. \end{aligned} \tag{2.80}$$

For a field error Δk_0, let us recall from the transfer map of the bending magnet (2.37) that there is constant term in the mapping for p_x, $(h - k_0)L$. This term is ideally zero, but in the case that an error Δk_0 is added to k_0, there is a non-zero orbit kick as per

$$\begin{aligned} \theta^{d,x}_{\Delta k_0} &= -\Delta k_0 L\,, \\ \theta^{d,y}_{\Delta k_0} &= 0\,. \end{aligned} \tag{2.81}$$

A rotation $\Delta\psi$ around the longitudinal axis can be treated as splitting the dipole field into two components, one in x and one in y as per:

$$\begin{aligned} \Delta k_0^x &= -k_0 \sin\Delta\psi \approx -\Delta\psi k_0 \,, \\ \Delta k_0^y &= k_0(\cos\Delta\psi - 1) \approx 0 \,, \end{aligned} \tag{2.82}$$

where $\Delta\psi \ll 1$ has been assumed. This directly induces orbit kicks

$$\begin{aligned} \theta_{\Delta\psi}^{d,x} &= 0 \,, \\ \theta_{\Delta\psi}^{d,y} &= -\Delta\psi k_0 L \,. \end{aligned} \tag{2.83}$$

Quadrupole Errors

For the treatment of the quadrupole the thin-lens approximation will be used and the assumption that the transverse planes are uncoupled. A transverse misalignment of $\boldsymbol{\delta}$ is equivalent to an increment in orbit at the point by $-\boldsymbol{\delta}$. This directly gives the kick for transverse misalignment:

$$\begin{aligned} \theta_{\Delta x}^{q,x} &= k_1 L \Delta x_{co} \,, \\ \theta_{\Delta y}^{q,x} &= 0 \,, \\ \theta_{\Delta x}^{q,y} &= 0 \,, \\ \theta_{\Delta y}^{q,y} &= -k_1 L \Delta y_{co} \,. \end{aligned} \tag{2.84}$$

The orbit kick induced by a field error follows analogously from the treatment of the bending magnet:

$$\begin{aligned} \theta_{\Delta k_1}^{q,x} &= -\Delta k_1 L x_{co} \,, \\ \theta_{\Delta k_1}^{q,y} &= \Delta k_1 L y_{co} \,. \end{aligned} \tag{2.85}$$

Similarly, the rotational error $\Delta\psi$ around the longitudinal axis is analogous to the bending magnet with the added observation that a quadrupole has two axes of symmetry in the transverse plane instead of the one for the dipole:

$$\begin{aligned} \Delta k_1^x &= k_1(\cos 2\Delta\psi - 1) + k_1 \sin 2\Delta\psi \approx 2k_1\Delta\psi \,, \\ \Delta k_1^y &= -k_1(\cos 2\Delta\psi - 1) + k_1 \sin 2\Delta\psi \approx 2k_1\Delta\psi \,, \end{aligned} \tag{2.86}$$

where $\Delta\psi \ll 1$ has been assumed. This induces the following kicks:

$$\begin{aligned} \theta_{\Delta\psi}^{q,x} &= -2k_1\Delta\psi L x_{co} \,, \\ \theta_{\Delta\psi}^{q,y} &= -2k_1\Delta\psi L y_{co} \,. \end{aligned} \tag{2.87}$$

Summary

The mapping from element error to induced orbit kick is summarized below in Table 2.1. An important observation from Table 2.1 is that all the induced kicks are linear in the machine errors. Recalling that the closed orbit perturbation in turn is linear in orbit kicks leads to the conclusion that the closed orbit perturbation is linear in the machine errors.

Table 2.1: Table over errors and corresponding transverse kicks.

BENDING MAGNET		
Error type	Kick in x	Kick in y
Δx	0	0
Δy	0	0
$\Delta\psi$	0	$-k_0\Delta\psi$
Δk_0	$-\Delta k_0 L$	0
QUADRUPOLE		
Error type	Kick in x	Kick in y
Δx	$k_1 L\Delta x$	0
Δy	0	$-k_1 L\Delta y$
$\Delta\psi$	$-2k_1 L x_{co}\Delta\psi$	$-2k_1 L y_{co}\Delta\psi$
Δk_1	$-L x_{co}\Delta k_1$	$L y_{co}\Delta k_1$

2.1.8 Statistical Interpretation of Twiss Paramters

All analysis up to this point has been analogous to that of a single particle with a given emittance. In practice, particles in a circular accelerator are inserted in groups called *bunches*, where one bunch has a given longitudinal length, contains up to billions of particles and a beam can be made up of thousands of bunches. By physical necessity, the particles in a bunch will have a spread in transverse phase space, and therefore in transverse emittance. This warrants a different interpretation of the Twiss parameters.

When transverse emittance is discussed for an actual beam, it is as a descriptor for the distrubition over transverse emittances. In this thesis, it is assumed that the underlying distribution is Gaussian, and all numbers provided in the context of emittance are the 1*RMS (root-mean-square) values. In this context, phase space ellipses for a given emittance ϵ_x can be seen as enveloping the phase space area where 68% of all particles in a bunch have ϵ_x or lower emittance. From this perspective, the closed orbit can be seen as the centroid of the Gaussian distribution, that is, the mean state vector particles assume in phase space. The argument for controlling the closed orbit remains the same: smaller closed orbit translates into greater distance to the beampipe and therefore less particles lost.

Emittance can either be specified in terms of *geometric emittance* ϵ_x, which is the emittance we have considered so far, or in terms of *normalized emttance* $\epsilon_{N,x}$, defined as:

$$\epsilon_{N,x} = \beta_{rel}\gamma_{rel}\epsilon\,, \tag{2.88}$$

where β_{rel} and γ_{rel} are the relativistic factors. The benefit of the normalized emittance is that it remains constant as the beam gets accelerated, which is not the case for the geometric emittance. The underlying cause is that our dynamical variable for the transverse momentum is scaled by the longitudinal momentum, and if the beam is accelerated, then the phase space ellipse has to shrink.

Another important metric of the beam is the *beam size* σ_x:

$$\sigma_x(s) = \sqrt{\beta_x(s)\epsilon_x}\,. \tag{2.89}$$

The beam size is recognizable from Figure 2.4 as the amplitude of the oscillation at any point s. Lastly, the value of the β function at an interaction point is denoted by β^*, and unless specified otherwise is assumed to be equal for the two transverse planes.

2.1.9 Luminosity

LHC and HL–LHC are *colliders*, meaning they accelerate beams up to a collision energy and then have them collide with each other. These collisions are performed to study rare events taking place inside them, and the more rare events that take place the better. The number of events per second can be written as:

$$\frac{dR}{dt} = \mathcal{L}\sigma_p\,, \tag{2.90}$$

where $\boldsymbol{L}$ is the *luminosity* and the σ_p the cross-section for a given event. The cross-section is a constant from the perspective of an accelerator designer, wheras the luminosity is a quantity that can be optimized for. Assuming both beams collide head-on, have the same internal structure and Gaussian cross-section, the luminosity can be expressed as [7]:

$$\mathcal{L} = \frac{N^2 f N_b}{4\pi\sigma^2} \times \exp\left(-\frac{d^2}{4\pi\sigma^2}\right), \tag{2.91}$$

where N is the number of particles in a bunch, N_b is the number of bunches in a beam, f is the revolution frequency of the beams, σ is the beam size at collision, assumed equal for both planes, and d is the distance between the centers of the beams at collision, referred to as the *beam separation*.

The luminosity as defined in (2.91) does not account for all effects, such as the impact of colliding at an angle, but it includes the major contributors relevant for this thesis. It follows that increasing the luminosity can be done by either increasing the number of particles in collsion, increasing the revolution frequency (more bunches in collision per unit time) or reducing the beam size (denser beams means more collisions). The term that is most relevant for this thesis is d, the beam separation. The beam separation is equal to the Pythagorean distance between the closed orbit positions of the beams at the collision point, and as such it provides a strong incentive to minimize the closed orbit at collision points.

2.2 Linear Algebra

The primary concern of this thesis is effective modeling of closed orbit perturbation and correction. Because of the linearity of closed orbit perturbation with respect to a kick, it is possible to model this in a linear algebra formalism, which in turn warrants a review of the tools in linear algebra that will be used. As the vector spaces considered will be finite, all discussions of linear mappings will be expressed in terms of matrices.

2.2.1 Notation and Elementary Operations

Let $\mathbf{A} \in \mathbb{R}^{m\times n}$. We will make use of the '†' operator as $\mathbf{A}^\dagger$ to signify the operation of taking a subset of rows and columns of $\mathbf{A}$. Which rows and columns is given by context and specified in text where ambiguous. We define the *range* of $\mathbf{A}$ as

$$\text{range}(\mathbf{A}) = \{\mathbf{A}\mathbf{x} \in \mathbb{R}^m \,\big|\, \mathbf{x} \in \mathbb{R}^n\}\,, \tag{2.92}$$

and the *kernel* as

$$\ker(\mathrm{A}) = \{\mathbf{x} \in \mathbb{R}^n \,\big|\, \mathrm{A}\mathbf{x} = \mathbf{0}\}\,. \tag{2.93}$$

The notation $\|\mathbf{x}\|_p\,, \mathbf{x} \in \mathbb{R}^n$, is used for the L^p-norm on $\mathbb{R}^n$, and $\mathbf{abs(x)}$ for the absolute value applied to each element of the vector $\mathbf{x}$.

Inequalities of the form $\mathbf{x} < \mathbf{b}$ define one inequality per element in $\mathbf{x}$.

2.2.2 Projection Matrices

A square matrix P of size $n \times n$ is called a *projection matrix* if it satisfies

$$\mathrm{P}^2 = \mathrm{P}\,. \tag{2.94}$$

A corollary is that if P is a projection matrix onto $\mathbf{range(P)}$, then $\mathrm{I} - \mathrm{P}$ is another projection matrix onto $\mathbf{ker(P)}$. Trivially it follows that

$$\mathrm{P}(\mathrm{I} - \mathrm{P}) = 0\,, \tag{2.95}$$

where 0 is the zero matrix. Finally, a projection matrix P which also satisfies

$$\mathrm{P}^\intercal = \mathrm{P}\,, \tag{2.96}$$

is called an *orthogonal projection matrix*.

2.2.3 Singular Value Decomposition

The Singular Value Decomposition (abbrev. SVD) is a decomposition that can be performed on any complex matrix, here we limit the treatment to real matrices. The (full) SVD of $\mathrm{A} \in \mathbb{R}^{m \times n}$ is written:

$$\mathrm{A} = \mathrm{U}\mathrm{S}\mathrm{V}^\intercal\,, \tag{2.97}$$

where U and V are orthogonal matrices of dimension $m \times m$ and $n \times n$ respectively and S is a $m \times n$ matrix of the form

$$\mathrm{S} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}, \tag{2.98}$$

and it has been assumed $m > n$. The σ_i in S are referred to as the *singular values* of A. Important properties of the SVD:

- $\sigma_1 \geq \ldots \geq \sigma_i \geq \ldots \geq \sigma_n \geq 0$
- $\mathrm{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i\,, \quad i = 1, \ldots, n$
- The column vectors $\mathbf{v}_i$ of V with $\sigma_i = 0$ span $\mathbf{ker(A)}$.
- The column vectors $\mathbf{u}_i$ of U with $\sigma_i > 0$ span $\mathbf{range(A)}$.

2.2.4 Moore-Penrose Inverse

Consider the problem

$$\mathbf{A}\mathbf{x} = \mathbf{b}\,, \tag{2.99}$$

where one is to solve for $\mathbf{x}$. Focusing on $\mathbf{A}$, there are three possibilities:

- $\mathbf{A}$ is invertible, i.e. there is a unique solution.
- $\mathbf{A}$ is underdetermined, i.e. there are infinitely many solutions.
- $\mathbf{A}$ is overdetermined, i.e. there is no solution.

Now, for the invertible case, we trivially get the solution

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}\,, \tag{2.100}$$

via the inverse. In the other scenarios, however, no inverse is available, in which case one can define a *pseudoinverse*. The most common pseudoinverse is the *Moore-Penrose inverse*, here denoted by $\mathbf{pinv}(\cdot)$ and used interchangeably with the word pseudoinverse from now on. It has the following properties:

- If $\mathbf{A}$ is invertible, then $\mathbf{pinv}(\mathbf{A}) = \mathbf{A}^{-1}$.
- If $\mathbf{A}$ is underdetermined, then $\mathbf{x}' = \mathbf{pinv}(\mathbf{A})\mathbf{b}$ is a solution to $\mathbf{A}\mathbf{x} = b$ with minimal L^2-norm.
- If $\mathbf{A}$ is overdetermined, then $\mathbf{x}' = \mathbf{pinv}(\mathbf{A})\mathbf{b}$ is a solution to $\mathbf{min}_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$ with minimal L^2-norm.
- It commutes with transposition: $\{\mathbf{pinv}(\mathbf{A})\}^\intercal = \mathbf{pinv}(\mathbf{A}^\intercal)$.

Another property of the pseudoinverse is that it can be used to construct orthogonal projections $\mathbf{P}_1$, $\mathbf{P}_2$ onto $\mathbf{range}(\mathbf{A})$ and $\mathbf{range}(\mathbf{A}^\intercal)$ respectively:

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{A}\,\mathbf{pinv}(\mathbf{A})\,, \\ \mathbf{P}_2 &= \mathbf{pinv}(\mathbf{A})\mathbf{A}\,. \end{aligned} \tag{2.101}$$

Lastly, a more esoteric property of the pseudoinverse is that it is *continuous*, under some conditions. To see that its continuity is not unconditional, consider the following example:

$$\begin{aligned} \mathbf{A}_\epsilon = \begin{bmatrix} \epsilon & 0 \\ 0 & 0 \end{bmatrix} \implies \mathbf{pinv}(\mathbf{A}_\epsilon) = \begin{bmatrix} 1/\epsilon & 0 \\ 0 & 0 \end{bmatrix}, \\ \lim_{\epsilon\to 0} \mathbf{A}_\epsilon = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \text{ but } \lim_{\epsilon\to 0} \mathbf{pinv}(\mathbf{A}_\epsilon) \text{ does not converge.} \end{aligned}$$

The condition under which the pseudo-inverse is continuous is that its argument must *retain its rank in the limit* [8]. In the above example, $\mathbf{A}_\epsilon$ converges to a matrix with rank zero, whereas it started out as a rank one matrix, hence the discontinuity of the pseudo-inverse.

Computing the Pseudoinverse

Several methods for computing the pseudoinverse exist, but for this thesis the one used will be based on the SVD. The merit of using an SVD-based pseudoinverse is that a variable number of singular values can be used. To see the merit of this, let us consider the following system of equations:

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} 10^{-12} & 0 \\ 0 & 1 \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} 10^{-3} \\ 1 \end{bmatrix}. \tag{2.102}$$

Here the matrix is close to being rank one, but its upper-left value is non-zero so the solution is $\mathbf{x} = \begin{bmatrix} 10^9 & 1 \end{bmatrix}^\mathsf{T}$. If one includes all singular values in the pseudoinverse, it will give the same solution. However if one enforces that no singular value smaller than 10^{-10} is used for the computation of the pseudoinverse, then $\mathbf{x} = \begin{bmatrix} 0 & 1 \end{bmatrix}^\mathsf{T}$. In this case, the argument could be made that the upper-left entry in $\mathbf{A}$ is virtually zero or numerical noise, and a singular value cutoff offers one method of systematizing this value judgement by leaving out small singular values.

The nomenclature will still refer to these pseudoinverses by '$\mathrm{pinv}(\cdot)$', but it is to be kept in mind that by excluding singular values in its computation some of the previous properties break down. For example, if any singular value is excluded then an invertible matrix will not have a pseudoinverse equal to its inverse, nor does the subsequent properties for the under- or overdetermined systems apply. However, removing singular values in the computation does not impact the commutativity, nor does it change the fact that $\mathbf{P}_1$ and $\mathbf{P}_2$ are orthogonal projections (even though their range is altered).

2.2.5 Linear Transformations of Random Vectors

Let $\mathbf{x} \in \mathbb{R}^n$ be sampled from a distribution $\mathbf{D}(\mathbb{R}^n)$ with covariance matrix $\mathbf{\Sigma}_x$, and let $\mathbf{A} \in \mathbb{R}^{m \times n}$. Let $\mathbf{y} = \mathbf{A}\mathbf{x}$, and $\mathbf{\Sigma}_y$ its covariance matrix. The following holds:

$$\mathbf{\Sigma}_y = \mathbf{A}\mathbf{\Sigma}_x\mathbf{A}^\mathsf{T}. \tag{2.103}$$

Equation (2.103) allows for directly computing the statistics of a linear operation applied to random vectors, assuming that the covariance of the distribution is known.

2.3 Optimization

As part of this thesis revolves around an optimization problem, it would be prudent to comment on it briefly. For analysis of the optimization problem at hand, see Chapter 4. If no structure beyond the problem category exists, then the most general paradigm of optimization is non-linear constrained minimization. It can be stated as follows:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, k, \\ & h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, l. \end{aligned} \tag{2.104}$$

When it comes to non-linear constrained optimization, methods of solving them are problem-dependent. Any structure beyond the minimal should be utilized to construct a good solver, such as *convexity*.

2.3.1 Convex Optimization

A function $r : \mathbb{R}^n \to \mathbb{R}$ is called *convex* if it satisfies that

$$r[\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}] \leq \lambda r(\mathbf{x}) + (1 - \lambda) r(\mathbf{y}), \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \forall \lambda \in [0, 1], \tag{2.105}$$

and a function $a : \mathbb{R}^n \to \mathbb{R}$ is called *affine* if it is of the form

$$a(\mathbf{x}) = \mathbf{c}^\mathsf{T} \mathbf{x} + b, \tag{2.106}$$

where $\mathbf{c} \in \mathbb{R}^n$ and $b \in \mathbb{R}$. Now, if the functions g_i and f are convex and all h_j are affine, then the problem in (2.104) belongs to the category of *convex optimization*. Convex optimization problems have the useful property that every local minimum is guaranteed to be a global minimum. Several solvers for convex optimization problems exist, and assuming that a given problem is well-conditioned and its dimensionality is manageable, then these solvers can be robust and effective.

Chapter 3
Framework Implementation

The framework was implemented as a Python package called POCKPy (Perturbation, Orbit Correction and Knobs in Python). It relies on the fact that all sources of closed orbit perturbation are linear in their kick. In Chapter 2, it was shown that element imperfections give rise to kicks linearly proportional to them, and that elements called orbit corrector dipoles can produce controllable orbit kicks. In this chapter, we will make use of these observations to construct a matrix formalism for computing the closed orbit.

For simplicity, we first assume a single-beam circular accelerator consisting out of nothing but the following elements:

1. Drift spaces
2. Bending magenets
3. Quadrupoles
4. Orbit correctors

Secondly, we assume that we have access to a table which provides the following quantities at the middle of each element in the accelerator:

1. Twiss parameters
2. Phase advances (relative to some point)
3. Position s along the accelerator (relative to some point)
4. Element parameters (L, k_0, k_1)
5. Closed orbit in (x, p_x) and (y, p_y)

If the accelerator under consideration had m elements, then there would be m rows in this table. This allows, for example, to store the values of β_x in a vector $\boldsymbol{\beta}_x$ of length m. Analogously, orbits in transverse phase space evaluated at these positions could be described by vectors of length $4m$.

3.1 Linear Algebra Formalism

We observe that given data listed above, it is possible to compute:

1. Any error-induced orbit kick.
2. The perturbation of the closed orbit as the result from any orbit kick.

Most importantly, recall that the closed orbit perturbations caused by machine errors are linear with respect to the errors. Let there be n unique machine errors where e_i is the magnitude of the i^{th} error. The closed orbit perturbation caused by this error can then be written

$$\Delta \mathbf{u}_i = e_i \mathbf{p}_i \,, \tag{3.1}$$

where $\Delta \mathbf{u}_i$ and $\mathbf{p}_i$ are $4m$ long vectors containing the closed orbit perturbation in the (x, p_x) and (y, p_y) phase spaces. Note that $\mathbf{p}_i$ is defined by (2.79) together with scaling factor from Table 2.1. It follows that the total closed orbit perturbation caused by machine errors can be written as follows:

$$\Delta \mathbf{u} = \sum_{i=1}^{n} e_i \mathbf{p}_i = \begin{bmatrix} \mathbf{p_1} & \mathbf{p_2} & \dots & \mathbf{p_n} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = \mathrm{RM_e}\mathbf{e} \,, \tag{3.2}$$

where $\mathbf{RM_e}$ is the *error response matrix* of size $4m \times n$ and $\mathbf{e}$ the *error vector* of length n. By an anologous derivation, the impact on closed orbit from powering the orbit correctors in the machine is given by:

$$\Delta \mathbf{v} = \mathrm{RM_c}\mathbf{c} \,, \tag{3.3}$$

where $\mathbf{RM_c}$ is the *corrector response matrix* of size $4m \times k$ and $\mathbf{c}$ is the vector containing the strength of each corrector.

We have derived linear mappings from the n-dimensional error space and k-dimensional corrector space to the $4m$-dimensional orbit space. Any possible instance of the machine imperfections corresponding to a vector $\mathbf{e}$ can be analyzed from the perspective of its impact on the closed orbit, its correction and the residual orbit in the machine, using the following equation:

$$\mathbf{r} = \mathrm{RM_e}\mathbf{e} - \mathrm{RM_c}\mathbf{c} \,. \tag{3.4}$$

To correct for a given machine, defined by its errors $\mathbf{e}$, means to achieve a suitable $\mathbf{r}$ by powering correctors $\mathbf{c}$, subject to constraints on $\mathbf{c}$ and $\mathbf{r}$.

Implementing an orbit knob in this formalism is done by solving for $\mathbf{c}$ in

$$\mathrm{RM_c}^{\dagger}\mathbf{c} = \mathbf{k} \,, \tag{3.5}$$

where † denotes the operation of taking a subset of rows, corresponding to the element positions and variables of interest, and a subset of columns, corresponding to desired correctors to be used for implementing a given knob, with $\mathbf{k}$ encoding the requisite values at said rows. We recall that the purpose of an orbit knob is to achieve a certain closed orbit in the machine, often corresponding to a certain opertational scenario. An example of an orbit knob is shown in Section 3.4.

Collider Caveat

Previous analysis assumed a circular accelerator with one beam, which does not fit the description of HL–LHC, a collider. This can be addressed by extending the number of rows in the matrices. If both beams pass through the same number of elements, then the extension would lead to an $8m$-dimensional orbit space, i.e. each row in the response matrices can be identified by a 3-tuple (beam, dynamical variable, element name).

From the perspective of computing the response matrices, this introduces two changes. Firstly, the table supplied as input has to contain all information for both beams. Secondly, some elements are shared between the two beams. This happens around collision points where the beams have to be brought close together before colliding, wherefore they pass through shared optics. This has an impact in that some correctors and sources of errors are common between the beams, i.e. there are elements which can impact both beams. In practice, this comes down to more bookkeeping, and while shared elements have an impact on the analysis, they do not considerably complicate the computation of the response matrices.

3.2 Core Structure

To implement the framework the programming language was chosen to be Python, specifically Python 3. The arguments, in no particular order, for doing so are:

1. The dominion of Python as a scripting language at CERN.

2. The availability of good packages for numerical linear algebra and data structures (e.g. `numpy` [9] and `pandas` [10]).

3. Already developed Python interfaces to software at CERN being available.

4. The possibility of performing the analysis on SWAN (Service for Web based ANalysis), a cloud-based data analysis service with a Jupyter notebook interface used at CERN.

Currently all HL–LHC and LHC optics are defined in MAD-X scripts, stored as `.madx` files. Different modes of analysis of optics are available inside MAD-X, among which the most relevant one for POCKPy is the `TWISS` module. It computes the Twiss parameters and closed orbit for a given beam and outputs it in a `.tfs` file, an output format unique to MAD-X. With this background, a simplified input pipeline and overview of POCKPy is given in Figure 3.1. POCKPy supports input in the form of either `.tfs` files or a `.madx` file defining a valid instance of HL–LHC. If a `.madx` file is supplied, it is executed by MAD-X inside the framework to produce `.tfs` files. Supplying one `.tfs` file per beam provides sufficient information for analysis in POCKPy. Once valid input is provided, it is parsed and passed around in the framework, finally resulting in the response matrices. After this, the user can directly interact with the `solver` module to perform analysis on the closed orbit.

More information together with the source code of POCKPy can be found at: `https://gitlab.cern.ch/jodander/pockpy`.

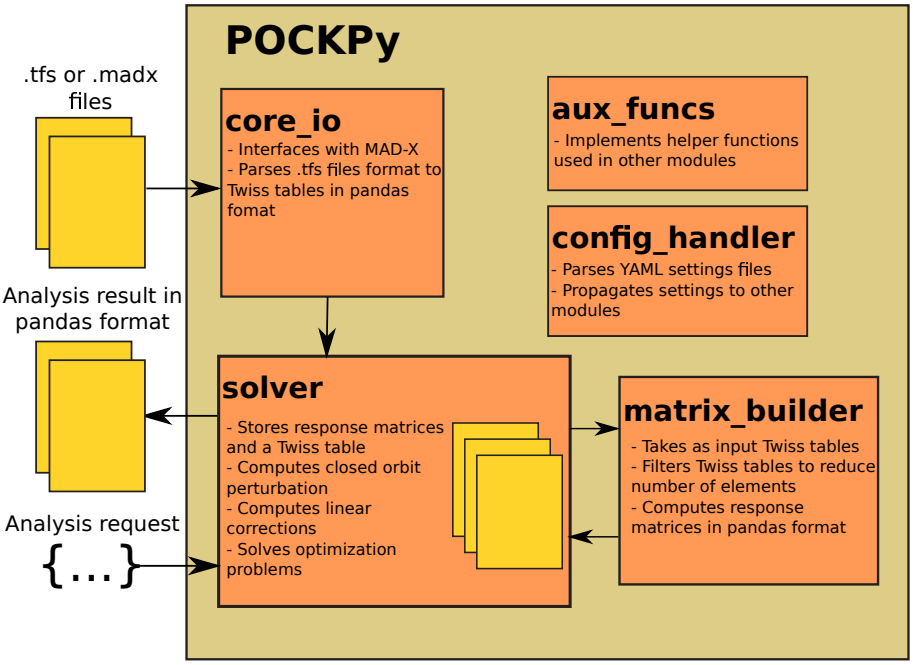


Figure 3.1: Rough schedule of the input-output pipeline and structure of POCKPy. Orange boxes correspond to Python modules with name in bold.

3.2.1 Twiss Table

The *Twiss table* is what gets parsed from the input and contains the necessary data for the computation of the response matrices. It has an ordered index where all defined elements in HL–LHC are sorted by longitudinal position and where each column corresponds to a variable evaluated at that position. A subset of the relevant columns are given in Table 3.1.

Table 3.1: Table describing a subset of the data columns in a Twiss table.

Column name	Description
`BETX`	Twiss β-function in the horizontal plane.
`ALFY`	Twiss α-function in the vertical plane.
`MUX`	Phase advance in horizontal plane.
`L`	Length of the element.
`K0L`	Integrated dipole field strength.
`K1L`	Integrated quadrupole field strength.
`X`	Closed orbit in x.
`Y`	Closed orbit in y.
`KEYWORD`	String specifying the element type.
`S`	Longitudinal position along the lattice.

From the perspective of closed orbit perturbation, the important quantities are the Twiss parameters. What the Twiss table provides in this respect is the Twiss parameters evaluated at the middle of every element in HL–LHC. For the purpose of this thesis, that is sufficient.

What is important is that the closed orbit is known at sensitive elements and that no source of error is missed, but since all sensitive elements and all sources of errors are, trivially, elements, nothing is missed by discretizing the path along the accelerator w.r.t. to the center of elements.

3.2.2 Response Matrices

One of the most important parts of the framework is the ability to compute the response matrices $\mathbf{RM_c}$ and $\mathbf{RM_e}$. Let us recall (2.79) which defines the closed orbit perturbation caused by a kick θ at $s = 0$, evaluated at an arbitrary s, repeated next for convenience.

$$x_{co}(s) = \frac{\theta\sqrt{\beta_0\beta_s}}{2\sin\pi Q}\cos(\pi Q - \mu)\,,$$
$$p_{x,co}(s) = \frac{\theta}{2\sin\pi Q}\sqrt{\frac{\beta_0}{\beta_s}}\sin(\pi Q - \mu) - \frac{\alpha_s}{\beta_s}x_{co}(s)\,.$$

We note that given a kick θ at the middle of an arbitrary element, the closed orbit perturbation at the center of every other element can be computed. Not only that, but due to the column structure of the Twiss table, the computation can be vectorized. We also observe that the conversion factor from machine imperfection to kick involves quantities which are available in the Twiss table, e.g. k_0 and L.

The data structure for the response matrices warrants a comment. The response matrices are stored as `pandas` DataFrames, meaning they can be accessed by an index and a column. The structure of the columns is direct: each column in the corrector response matrix corresponds to a corrector, and each column in the error response matrix corresponds to a type of error for a given bending magnet or quadrupole. The index is a bit more contrived, it is accessed by a 3-tuple of `(beam, dymamical_variable, element)`. The benefit of this structure is that given a source of closed orbit perturbation, its effect can be evaluated for beams, dimensions and elements in a single matrix multiplication.

Technicalities beyond what has been discussed do exist, e.g. taking the direction of beams into consideration, but the general idea has been accounted for. Computing response matrices can be done efficiently with the Twiss table.

3.3 Additional Routines

Here a few general comments are dedicated to routines in the framework not covered before.

3.3.1 Slicing the Twiss Table

HL–LHC, as its predecessor LHC, are big machines from the perspective of the number of elements in them; if the full accelerator is considered, the Twiss table for HL-LHC contains over 31 thousand rows. Combining this with the fact that a good portion of these are sources of error and the fact that each response matrix need a factor four as many rows as the Twiss table (one per transverse dynamical variable), the final response matrices will have hundreds

of millions of entries, each encoded as a float value. Even assuming 32-bit float representation, physical memory in the order of 64 GB would still be necessary for their storage.

In order to minimize the amount of memory, and therefore also speeding up the calculations, a method for filtering the Twiss table w.r.t. element types, longitudinal position and whether it is a source of closed orbit perturbation was introduced. The result is that the user can choose to focus on a certain region (e.g. around `IP5`) or, alternatively, if the user requires to model the full accelerator, it is possible to select only the most relevant elements (e.g. correctors, quadrupoles), which can shrink the number of rows to the point where the response matrices can be stored in memory without floating point precision loss, memory swapping or crashing.

3.3.2 Pseudoinverse

The pseudoinverse routine used in POCKPy is the SVD-based `pinv` method in `numpy.linalg`. It is however wrapped to allow for:

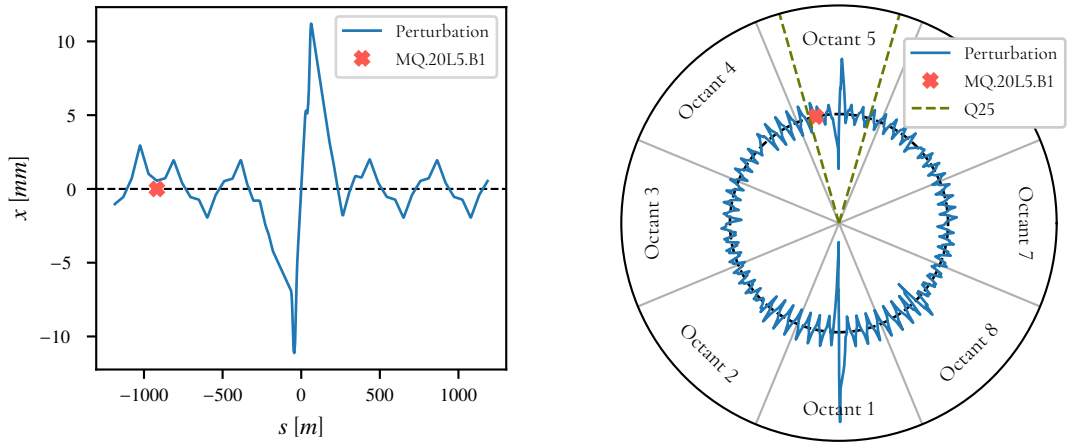
- Specifying the number of singular values to retain. The original `pinv` method only implements a cutoff based on the relative size of singular values.
- Computing separately the pseudoinverse for the horizontal and vertical plane, as the two plane are supposed to be uncoupled in this treatment. This allows to have the singular values computed independently for each plane.
- Allowing for DataFrames as input, while retaining and inverting the index and columns in the result.

3.3.3 Optimization

For later optimization of the later knob implementation and error correction, the Python package `cvxpy` [11, 12] was used. It is a convex optimization library in Python with an embedded language for defining constraints and objective functions. It is built on the theoretical framework of *disciplined convex programming* [13], which is used to break each function down into its components to try to ascertain its convexity, and then uses the information to find an appropriate solver. Both the error correction and the knob implementation optimization problems posed in Chapter 4 proved tractable when using the SOCP (second-order cone programming) solver ECOS [14], included in `cvxpy`.

3.4 Example Computations and Output

Before verifying the integrity of the framework and performing studies using it, an overview of what the framework outputs will be provided. For quantitative analysis, the most straightforward format for analyzing the closed orbit is to represent it as a function of the longitudinal position along the accelerator, often only for a section of the accelerator. Such a plot can be seen in Figure 3.2 where the horizontal closed orbit perturbation for horizontally shifting a quadrupole is shown for Beam 1. The important observation is that there is nothing intrinsically local about errors in a given region, their effect is global and often have their greatest



(a) Octant 5.

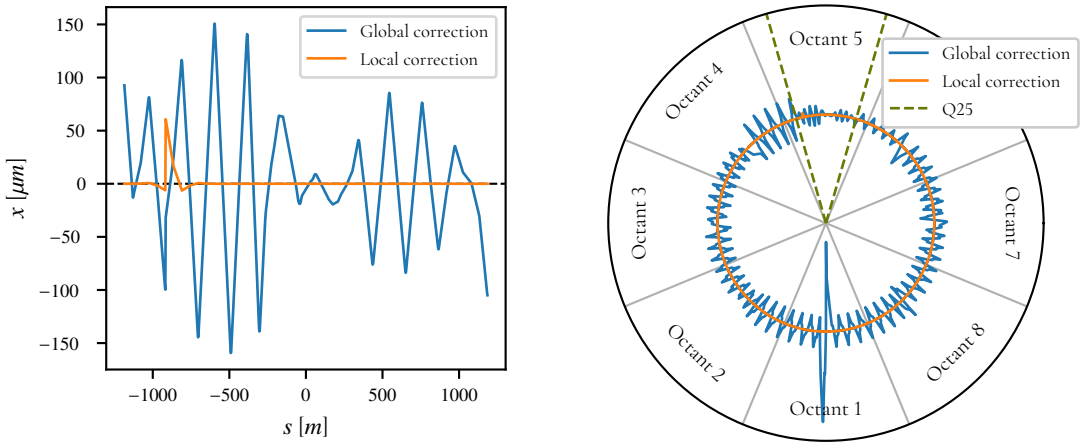
(b) Full ring.

Figure 3.2: Closed orbit perturbation from a horizontal misalignment of the quadrupole `MQ.20L5.B1` plotted for Beam 1 in the horizontal plane. Plotted as an unfolded line from Q25 to Q25 in octant 5 **(a)**, and for the full ring **(b)**.

impact far away from their immediate vicinity. Plots of the same type as Figure 3.2a will be used for studies and comparisons as they are easier for quantitative reasoning, whereas polar representation mainly serves for qualitative analysis. The work in this thesis will be focused around octant 5 enclosed by `Q25`, the same region as presented in 3.2a. Importantly, this runs contrary to the previous argument about closed orbit perturbation being a global phenomenon, but by intelligently correcting the orbit, it can be canceled such that it does not propagate outside of a given region, granted there are sufficiently many and effectively positioned orbit correctors. This can be seen in Figure 3.3a, which shows the residual after two different corrections of the horizontal misalignment of `MQ.20L5.B1`. The global correction corresponds to a correction that reduces the orbit in the region with limited corrector usage, whereas the local correction corresponds to a correction where the error is corrected close to its origin at the expense of additional orbit corrector strength. This is for octant 5, the corresponding residuals for the full machine can be seen in Figure 3.3b.

Note that the residual for the local correction does not impact the closed orbit outside of octant 5 as the closed orbit perturbation in Figure 3.3a was canceled at the end points. Thanks to the linearity of the closed orbit perturbation and correction problem, if we only deal with the errors in a given region, and if the closed orbit at the boundaries of such a region are set to be the zero vector, then one can consider only that portion of the machine, without looking elsewhere.

Lastly, an example of an orbit knob is shown in Figure 3.4. This knob has been implemented by finding a solution to (3.5), where the equation system enforced $p_x = \pm 295\,\mu\text{rad}$ at the interaction point and zero orbit and angle at elements `Q12` on each side of `IP5`. The dashed lines shows the same knob scaled to achieve a crossing angle of $\pm 100\,\mu\text{rad}$. This is a general attribute of knobs: they are scalable. From an operational perspective, this allows for smooth transitions when activating knobs, and the ability to fine-steer closed orbit at select locations by using predefined knobs.



(a) Octant 5.

(b) Full ring.

Figure 3.3: Residual orbit from two different corrections of the orbit perturbation from a 1 mm horizontal misalignment of the quadrupole `MQ.20L5.B1`, shown for Beam 1 in the horizontal plane. Plotted as an unfolded line from `Q25` to `Q25` in octant 5 **(a)**, and for the full ring **(b)**.

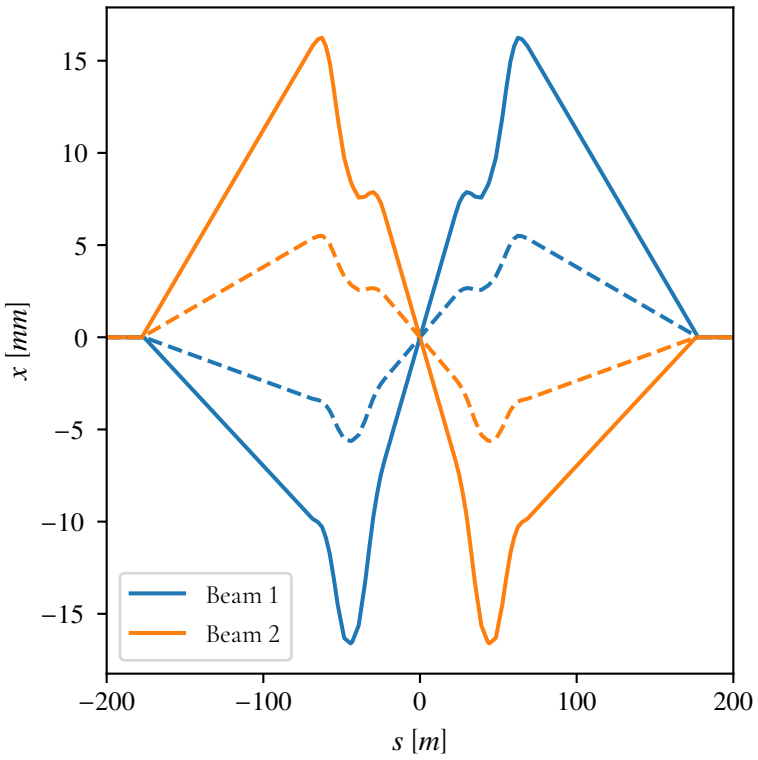


Figure 3.4: An implementation of a knob enforcing a crossing angle of ±295 μrad in the horizontal plane at `IP5`, zoomed in around `IP5`. Dashed lines correspond to multiplying the corrector configuration by 100/295.

3.5 Verification of Framework

As the end goal of POCKPy is to provide continuous verification of HL–LHC optics, it is necessary to verify that the framework is consistent with tools already in place. The verification of HL–LHC optics is today largely done in MAD-X scripts, wherefore the most direct means of cross-checking the output of POCKPy is comparison with MAD-X. The comparison considered here used the following test cases:

1. Verify computation of closed orbit for a distribution of errors.
2. Verify computation of closed orbit for a distribution of corrector strengths.
3. Verify that a given error and its correction produces the same closed orbit residual.

The distributions used for the comparisons were uniform with boundaries for the machine errors given as

- ± 0.5 mm transverse quadrupole misalignments
- ± 0.002 relative strength error for quadrupoles and dipoles
- ± 1 mrad roll for quadrupoles and ± 0.5 mrad for dipoles

and boundaries for the corrector strengths given as ± 1 mTm, where the error distribution is the one currently used to model HL–LHC. All comparisons were performed on the HL–LHC collision optics version 1.5, release candidate 0 [15]. The section of the accelerator investigated was octant 5, from `Q25` to `Q25` around `IP5`, for both beams and planes.

3.5.1 Computing Perturbations in MAD-X

The procedure of computing perturbations in MAD-X is summarized as follows:

1. Compute the closed orbit for no machine errors.
2. Add machine errors.
3. Compute the new closed orbit in presence of machine errors.
4. Detract the first orbit from the second to get the closed orbit perturbation.

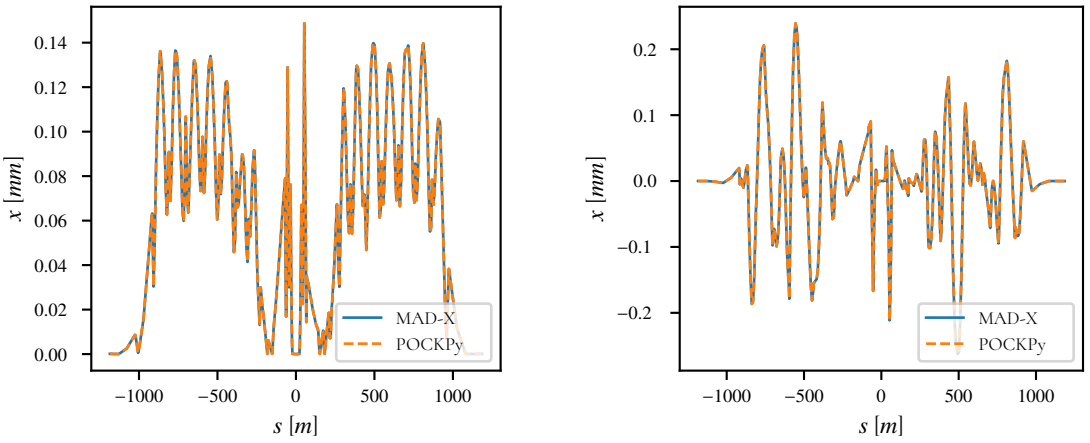
As the computation has to be performed for the full machine in order for it to qualify as a closed orbit perturbation, the process is not instantaneous, taking in the order of a minute on a Dell XPS 13 with Intel Core i7-8550U CPU @ 1.80GHz when run as a single process. This is one of the reasons for using an analytical estimation of orbit perturbation and correction as presented in Chapter 2 and 3.

3.5.2 Verifying Linearity of MAD-X

For this verification, the response matrices inside POCKPy were initialized by direct interaction with MAD-X. Unit errors were added to elements, unit kicks to correctors, and the closed orbit perturbation reported by MAD-X for each of these alterations was inserted into the columns of the response matrices. Subsequently these matrices were used for direct comparison with MAD-X. Each of the test cases was run for 100 samples, where the comparison of closed orbit post-correction was performed as follows:

1. An erroneous machine was sampled from the distribution.
2. The machine was corrected by applying a pseudoinverted corrector response matrix inside POCKPy.
3. The errors and their corrections were entered into MAD-X and the resulting closed orbit residual was compared to the one in POCKPy.

The orbits induced by machine errors as well as those induced by correctors showed no notable discrepancy. The scenario most likely to deviate is the third case where the residual after correction is analyzed, as it is formed from the difference between two orbits of comparable size, i.e. the error-induced perturbation and its approximate negation in the corrector-induced correction. This case proved consistent as well with plots shown in Figure 3.5 for 100 samples. The RMS orbit in Figure 3.5a shows no noticeable discrepancy, nor does the single



(a) RMS.

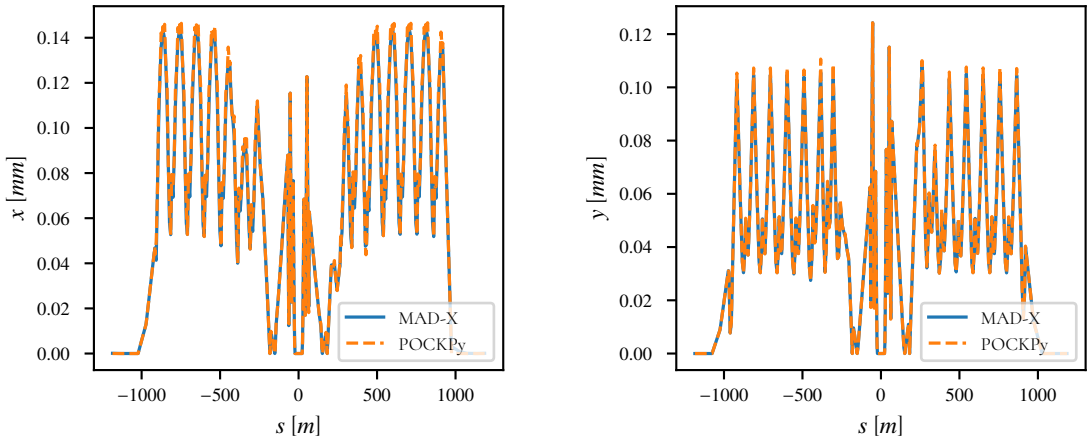
(b) Worst sample.

Figure 3.5: Comparison between MAD-X and POCKPy (using response matrices from MAD-X) for the residual post-correction, plotted for Beam 1 in the horizontal plane. Shown for **(a)**, the RMS over all 100 samples, and **(b)**, the sample with biggest deviation as measured in L^2 norm.

sample with the greatest measured deviation in Figure 3.5b. The fact that the worst sample from the worst test scenario has good correspondence provides a strong argument for believing that the closed orbit computations performed in MAD-X can be well-approximated by response matrices.

3.5.3 Verifying Analytical Response Matrices

In the second verification step, the response matrices initalized by MAD-X was compared to the response matrices computed analytically inside POCKPy, using the same test scenarios and distributions as before. By the same reasoning, the scenario most likely to show discrepancies is the error correction, which is showcased in Figure 3.6. For the horizontal residual in



(a) Horizontal plane.

(b) Vertical plane.

Figure 3.6: Comparison between MAD-X and POCKPy matrices for the RMS residual post-correction for a pseudo-inverse-based correction, here shown for Beam 1 and both planes.

Figure 3.6a, there are minor discrepancies visible at some of the peaks of the residual, albeit not that large. Their root cause is field errors in bending magnets: they are treated differently between MAD-X and POCKPy, and arguably more consistently in POCKPy. Since the bending magnet field error only affects the horizontal plane, Figure 3.6b, showing the vertical closed orbit residual, lacks this discrepancy. There are still minor differences visible at some peaks, which is dominated by a different assumption: MAD-X treats quadrupoles as thick lenses, whereas POCKPy assumes they can be treated as thin lenses, and this explains the remaining discrepancy.

Based on this inquiry, it has been established that there is a discrepancy between the computation of closed orbit in MAD-X and POCKPy. This discrepancy can be traced back to a difference in assumptions made in POCKPy contrary to MAD-X, and it is ultimately small enough not to significantly impact the results presented in this thesis. Summarizing, POCKPy is deemed sufficiently consistent with MAD-X to be used in its place for closed orbit perturbation computations.

Chapter 4
Orbit Corrector Budget

In this chapter the formalism developed in Chapter 3 will be made use of to state the error correction and knob implementation problems. In words, the objective is to implement a set of knobs and to correct for closed orbit perturbation caused by machine errors, both of which require the use of orbit correctors. Each orbit corrector has an upper limit on how great a kick it can produce, hence it makes sense to talk of an *orbit corrector budget*, which enforces a constraint to be respected across both problems. This orbit corrector budget problem will be stated, analyzed and solved for HL–LHC.

Unavoidably, this study makes use of the element naming scheme used in LHC and HL–LHC, and while it is possible to still deduce the meaning of the names employed, the naming conventions are described in Appendix A for greater clarity, if necessary. For ease of notation, elements will also be referred to using *regular expressions* [16] from now on. A regular expression defines a pattern, and when a regular expression is used in this thesis, we are referring to every element matching this pattern. A good reference for regular expression can be found at [17], but for the purpose of thesis it suffices to know that an expression of the form `B[12]`, matches a `B` followed by either a `1` or a `2` and that any subpattern followed by `?` makes that pattern optional to match for. As an example, `[AB]?2` matches `A2`, `B2` or `2`.

4.1 Problem Formulation

The problem will be approached by splitting it into two separate parts: the error correction and the knob implementation. The reasons for doing so are

- Implementing knobs is done independently of the error correction in practice.
- Error correction is contingent on which error configuration, drawn from a distribution, is being corrected. Treating the error correction separately allows for statistical analysis of corrections over a given distribution.

- There are applications where only one of the two problems are of interest.

We begin with the error correction problem.

4.1.1 Error Correction

When correcting for machine errors in an accelerator one measures the closed orbit perturbation in the machine and then powers orbit correctors to negate the measured perturbation. It is however not the case that the closed orbit can be measured at every element; it can only be measured at dedicated elements called *beam position monitors* (BPMs for short), and moreover only the transverse closed orbit positions, not angles, are observable. With this in mind, we state the error correction problem as follows:

$$\begin{aligned}
&\text{find} && \mathbf{c} \in \mathbb{R}^k\,, && \text{(4.1a)}\\
&\text{given} && \mathbf{e} \in \mathbb{R}^n\,, && \text{(4.1b)}\\
&\text{subject to} && \text{abs}(\mathbf{c}) \leq \mathbf{b}_{err}\,, && \text{(4.1c)}\\
& && (\mathrm{RM_e}\mathbf{e} + \mathrm{RM_c}\mathbf{c}_{q+1})^{\dagger_{\text{eq}}} = \mathbf{0}\,, && \text{(4.1d)}\\
& && \text{abs}(\mathrm{RM_e}\mathbf{e} + \mathrm{RM_c}\mathbf{c})^{\dagger_{\text{ineq}}} \leq \mathbf{t}\,. && \text{(4.1e)}
\end{aligned}$$

The problem as posed in (4.1) involves finding a feasible corrector strength configuration $\mathbf{c}$, contingent on a given machine error configuration $\mathbf{e}$,under constraints on the closed orbit residual and corrector strength. The constraint on the corrector strength in (4.1c) is given in absolute value as the direction of orbit corrector kicks is irrelevant; it is the magnitude of the kick that is constrained. Equation (4.1d) takes a subset of transverse positions measured at BPMs (i.e. a subset of rows) by applying the operator $\dagger_{\text{eq}}$ and enforces that the closed orbit must be zero at these locations. This constraint is used to enforce small closed orbit around chosen BPMs, e.g. BPMs surrounding an interaction point. Finally, the inequality in (4.1e) uses the operator $\dagger_{\text{ineq}}$ to take all rows with BPMs and transverse position that was not included in $\dagger_{\text{eq}}$ to put constraints on absolute closed orbit residual.

In terms of structure, note that the constraints are trivially convex, even linear. The absolute value inequalities can be written as two linear vector inequalities since $|a| < b \iff -a < b\,, a < b$, hence if there is a feasible region, it is a convex polytope. To give the problem some more structure, it can be changed from a feasibility problem to a minimization one by adding an objective function f, as per

$$\begin{aligned}
&\min_{\mathbf{c}\in\mathbb{R}^K} && f(\mathbf{c})\,,\\
&\text{given} && \mathbf{e} \in \mathbb{R}^n\,,\\
&\text{subject to} && \text{abs}(\mathbf{c}) \leq \mathbf{b}_{err}, \qquad (4.2)\\
& && (\mathrm{RM_e}\mathbf{e} + \mathrm{RM_c}\mathbf{c}_{q+1})^{\dagger_{\text{eq}}} = \mathbf{0}\,,\\
& && \text{abs}(\mathrm{RM_e}\mathbf{e} + \mathrm{RM_c}\mathbf{c})^{\dagger_{\text{ineq}}} \leq \mathbf{t}\,.
\end{aligned}$$

Many possible choices for f exist, but a decent one for the problem is

$$f(\mathbf{c}) = \|(\mathrm{RM_c}\mathbf{c} + \mathrm{RM_e}\mathbf{e})^{\dagger_{\text{ineq}}}\|_2 + \lambda\|\mathbf{c}\|_2\,, \quad \lambda \geq 0\,. \tag{4.3}$$

which is a convex function. The convexity follows from the norm being a convex function, the composition of a convex function with an affine function being convex, and the sum of

two convex functions being convex. This makes the optimization problem of (4.2) convex. Most importantly, it allows for the overall minimization of the orbit with a regularizing term penalizing inefficient use of the allotted corrector strength.

Linear Error Correction

While the convex formulation of the error correction is sufficient for correcting errors on a case-by-case basis, there is an argument for expressing the correction as a linear mapping, i.e. a matrix mapping from orbit measured at BPMs to the corrector strength for its correction. The reasons for wanting to do this are multiple:

1. Such an error correction can be computed once and then re-used indefinitely for different machine errors.

2. By virtue of it being a linear mapping, the covariance matrix of the residual and corrector strength used can be computed directly, instead of estimated by sampling error distributions with known covariance matrices

3. The *orbit correction feedback*, a low-level control process for continuous correction of beams in LHC, is based on a linear correction [18, 19]. More on this in Chapter 5.

A naive solution would be to directly apply the pseudoinverse to the corrector response matrix with an adequate number of singular values, which guarantees a solution and minimizes the L^2-norm (for the given number of singular vectors) of the residual $\mathbf{r}$, as defined by

$$\mathbf{r} = \mathrm{RM_c}^{\dagger_{\mathrm{BPM}}}\mathbf{c} + \mathrm{RM_e}^{\dagger_{\mathrm{BPM}}}\mathbf{e} = \mathrm{RM_c}^{\dagger_{\mathrm{BPM}}}\mathbf{c} + \mathbf{o}\,. \tag{4.4}$$

where $\dagger_{\mathrm{BPM}}$ is the operation of taking all rows corresponding to BPMs and transverse closed orbit positions. An issue with directly applying the pseudo-inverse and computing $\mathbf{c}$ as

$$\mathbf{c} = -\,\mathrm{pinv}(\mathrm{RM_c}^{\dagger_{\mathrm{BPM}}})\mathbf{o} \tag{4.5}$$

is that minimizing the L^2-norm of $\mathbf{r}$ is not necessarily conducive to the objective. When minimizing the residual, each component of the residual is considered equally important. E.g. considering a full accelerator with an arbitrary density of BPMs along it, if all BPMs are treated equally then sections of the accelerator with a higher density of BPMs will be more aggressively corrected, and vice versa. A slight modification to the naive approach that can allay this concern and add flexibility is

$$\mathbf{c} = -\,\mathrm{pinv}(\Lambda^L\mathrm{RM_c}^{\dagger_{\mathrm{BPM}}})\Lambda^L\mathbf{o}\,, \tag{4.6}$$

where Λ^L is a diagonal matrix where each entry functions as a weight for a certain beam, BPM, and dynamical variable. By weighting important BPMs more, the underlying least-squares problem of the pseudoinverse will penalize orbit at those BPMs more, leading to more correction at those locations. Extrapolating the same argument, if a few BPMs are weighted considerably more than all others, they will be corrected to zero orbit. This allows for a linear correction strategy to enforce hard constraints on orbit. Inversely, a weight of zero implies that the correction will not utilize the information at the corresponding BPM.

There could also be scenarios where a subset of correctors should be used less or more than their complement. This can be achieved in the same way be by adding yet another diagonal matrix Λ^R:

$$\mathbf{c} = -\Lambda^R \,\mathrm{pinv}(\Lambda^L \mathrm{RM_c}^{\dagger_{\mathrm{BPM}}} \Lambda^R)\Lambda^L \mathbf{o} = A\mathbf{o}\,, \tag{4.7}$$

where each diagonal entry in Λ^R corresponds to a certain corrector. In this case, greater weight corresponds to it being used more and a weight of zero to it not being used at all. In general, we refer to $\mathbf{A}$ of (4.7) as a *linear correction strategy*.

The alert reader will have noticed that what is provided by (4.7) is not an algorithm for finding 'the best error correction strategy' given $\mathbf{RM_c}$, $\mathbf{RM_e}$ and a distribution over errors $\mathbf{e} \in \mathbb{R}^n$, but rather a way of encoding error corrections. Beyond heuristics such as 'give greater weight to BPMs in important sections', there is no optimization routine involved in finding a good $\mathbf{A}$; it is currently done ad hoc.

Statistical Analysis Based on Linear Correction

Given a linear correction strategy $\mathbf{A}$ that maps from orbit observed at BPMs to corrector strength, it is possible to analyze some statistical quantities if the distribution of machine error configurations is known. If so, the resulting residual after correcting and the corrector strength used for it also become random variables as per:

$$\begin{aligned} \mathbf{c} &= \mathrm{ARM_e}^{\dagger_{\mathrm{BPM}}}\mathbf{e} = \mathrm{B_c}\mathbf{e}\,, \\ \mathbf{r} &= \mathrm{RM_c ARM_e}^{\dagger_{\mathrm{BPM}}}\mathbf{e} + \mathrm{RM_e}\mathbf{e} \\ &= (\mathrm{RM_c ARM_e}^{\dagger_{\mathrm{BPM}}} + \mathrm{RM_e})\mathbf{e} = \mathrm{B_r}\mathbf{e}\,. \end{aligned} \tag{4.8}$$

Now, assuming that $\mathbf{e}$ has a covariance matrix Σ_e, the corresponding covariance matrices for $\mathbf{c}$ and $\mathbf{r}$ can be derived from the standard transform of covariance matrices:

$$\begin{aligned} \Sigma_c &= \mathrm{B_c}\Sigma_e \mathrm{B_c}^\intercal\,, \\ \Sigma_r &= \mathrm{B_r}\Sigma_e \mathrm{B_r}^\intercal\,. \end{aligned} \tag{4.9}$$

Another interesting metric that can be derived in this manner is the *worst-case* corrector strength and residual. If it is assumed that the errors have an absolute bound, $\mathbf{e}_{max}$, then the worst-case quantities can be computed as:

$$\begin{aligned} \mathbf{c}_{wc} &= \mathrm{abs}(\mathrm{B_c})\mathbf{e}_{max}\,, \\ \mathbf{r}_{wc} &= \mathrm{abs}(\mathrm{B_r})\mathbf{e}_{max}\,. \end{aligned} \tag{4.10}$$

It is worth stating that the worst-case quantities are evaluated on a per entry basis for the two vectors. For example, this means that the worst-case for a given corrector is not necessarily the same worst-case as for any other corrector.

4.1.2 Knob Implementation

To implement a knob is to find a corrector strength configuration such that the closed orbit perturbation at prescribed element positions attain requisite values, as defined in (3.5). An

example of a knob could be to move both beams in opposite directions at a collision point. Such a knob could then be powered before entering collision, separating the two beams, and then be gradually turned off to move the two beams into collision. Another feature of knobs is that when utilized, they are used up to a sign; if $\mathbf{c}$ defines a knob, then $-\mathbf{c}$ could also be run in practice, and would be referred to as defining the same knob. Said differently, knobs are implemented up to a sign.

Unlike error correction, the implementation of a knob is done offline, i.e. independent of any error-induced orbit in the machine, which allows for knob definitions to include elements other than BPMs. We state the knob implementation problem as follows:

$$\begin{aligned} &\text{find} && \mathrm{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_q] \in \mathbb{R}^{k\times q} && \text{(4.11a)}\\ &\text{subject to} && \sum_{j=1}^{q} \mathrm{abs}(\mathbf{c}_j) \leq \mathbf{b}_{knob}\,, && \text{(4.11b)}\\ & && \mathrm{RM_c}^{\dagger_j}\mathbf{c}_j = \mathbf{k}_j\,, \quad j = 1, \ldots, q\,. && \text{(4.11c)} \end{aligned}$$

The problem as formulated in (4.11) concerns finding q valid knob implementations, where the i^{th} knob is defined to implement a closed orbit for variables and positions defined by $\dagger_i$ with values $\mathbf{k}_i$, under constraints on the total corrector strength employed. The inequality in (4.11b) needs to take the absolute value of each term since there could otherwise be cancellation between knobs that in practice would add up, and hence a feasible point for the problem might be an unfeasible point in practice, which would be an unattractive feature of the formulation. In practice, there will always be some cancellation between corrector strength configurations for any combination of signs, as such, the formulation, while not giving false positives, will be more pessimistic than is the real-world case.

An important point to mention before delving deeper into the problem is that the equation systems posed in (4.11c) are underdetermined. This is not conveyed by how they are presented, but makes sense from an operational perspective: if tasked with implementing a set of offsets and angles at different positions along the accelerator, it has to be assumed that the task is feasible to begin with, and more likely than not, there is probably some degree of freedom in the implementation. With this in mind, the equality constraints in 4.11c form hyperplanes in $\mathbb{R}^{k\times q}$. The inequality constraints in (4.11b) based on the absolute value of vectors can be rewritten as 2^q linear vector inequalities, which form half-space in $\mathbb{R}^{k\times q}$. Supposing that the problem is feasible, then the feasible region is formed in the intersection of a set of q hyperplanes and 2^q half-space in $\mathbb{R}^{k\times q}$, which makes it another convex polytope.

Once more, picking a convex objective functions renders the optimization problem convex:

$$\begin{aligned} &\min_{\mathrm{C}\in\mathbb{R}^{k\times q}} && f(\mathrm{C}) && \\ &\text{subject to} && \sum_{j=1}^{q} \mathrm{abs}(\mathbf{c}_j) \leq \mathbf{b}_{knob}\,, && \text{(4.12)}\\ & && \mathrm{RM_c}^{\dagger_j}\mathbf{c}_j = \mathbf{k}_j\,, \quad j = 1, \ldots, q\,. && \end{aligned}$$

The form of the objective function was chosen as

$$f(\mathrm{C}) = \|\mathrm{W}_0 \sum_{j=1}^{q} \mathrm{abs}(\mathbf{c}_j)\|_2 + \sum_{k=1}^{l} \|\mathrm{W_k RM_c}\mathbf{c}_{i_k}\|_{p_k}\,. \tag{4.13}$$

where p_k is either 2 or ∞. Firstly note that the objective function achieves what has been requested of the optimizer. The first term corresponds to minimizing the absolute corrector strength of some subset of all correctors used, encoded in $\mathbf{W}_0$ with only positive entries. The second term(s) attempts to minimize the orbit in an L^{p_k} sense at some locations, encoded in $\mathbf{W}_\mathrm{k}$ for knob i_k. In essence, the optimization function captures the ability to minimize the usage of a set of correctors, and to minimize the orbit at a custom set of locations in the accelerator for different knobs.

Secondly, observe that the objective function is convex. The second term(s) in the function are convex by the same argument as for the error correction objective function: convex functions composed with affine functions are convex. The first term is convex because the absolute value is a convex function, a sum of convex functions is convex, a positive weighted sum of convex functions is also convex and, lastly, a compound function where the outer one (here the norm) is convex and non-decreasing in each of its arguments, is convex, if each of the arguments is a convex function. Since each term in the objective function is convex, it follows that the objective function is convex too.

4.1.3 Satisfying the Orbit Corrector Budget

Both (4.2) and (4.12) have constraints on the corrector strength employed, given by $\mathbf{b}_{err}$ and $\mathbf{b}_{knob}$ respectively. Assuming the orbit corrector budget introduces a total constraint of $\mathbf{b}$, and we have $\mathbf{b} = \mathbf{b}_{err} + \mathbf{b}_{knob}$, then solving the error correction and knob implementation problems implies that the total corrector strength used will be within the budget. By this reasoning, the larger problem of satisfying the orbit corrector budget can seen as iterative in the following sense:

1. Reserve corrector strength for the error corrections as defined by $\mathbf{b}_{err}$.

2. Verify that $\mathbf{b}_{err}$ confers a sufficient amount of corrector strength to correct the closed orbit adequately *sufficiently often*.

3. Find *one* feasible knob implementation given the corrector strength constraint $\mathbf{b}_{knob} = \mathbf{b} - \mathbf{b}_{err}$.

This approach has the downside of making $\mathbf{b}_{err}$ a hyperparameter, but it is at least an intuitive hyperparameter: greater entries in $\mathbf{b}_{err}$ means more corrector strength for error correction and less for knob implementation, and vice versa. As for the error correction, what is deemed as *sufficiently often* is context dependent, but will always involve deriving statistics from the error correction based on an assumed distribution of machine errors.

4.2 Results

After POCKPy was implemented, studies were conducted using it. This is a study of the orbit corrector budget performed inside POCKPy.

4.2.1 Boundary Conditions and Specifications

For the investigation of the orbit corrector budget the HL–LHC optics version 1.5, release candidate 0 was considered. As proven in Section 3.4 it is possible to focus on a small section of the ring of interest. Here octant 5 was chosen, from `MQ.25L5.B[12]` to `MQ.25R5.B[12]`. Because the interaction regions in octant 1 and 5 are the main regions affected by the HL–LHC upgrade, and since they are symmetric, it suffices to treat one of them. In treating the error correction, the following assumptions were put on the errors:

- ± 0.5 mm transverse quadrupole misalignments
- ± 0.002 relative strength error for quadrupoles and dipoles
- ± 1 mrad roll for quadrupoles and ± 0.5 mrad for dipoles
- No machine errors past `Q20`.

where the values specify bounds on the uniform distribution. Moreover, the correction had to implement:

- Zero residual orbit at `IP5`.
- Zero residual orbit at the crab cavities, `ACFCA.[AB][LR]5.B[12]`.
- No orbit leakage at the end of the accelerator segment (i.e. zero residual orbit and angle at `MQ.25[LR]5.B[12]`).
- Less than 1 mm residual orbit at every other position.

The zero orbit at `IP5` is due to it being a point of collision, whereas the zero orbit at the crab cavities follows from how these elements affect the luminosity (see Appendix B for more details on crab cavities). Enforcing no orbit leakage at the extremities of the segment under consideration is in line with the assumption that each octant can be treated independently. Finally, enforcing a sub-millimeter residual at every other point is done to reduce the distance between beam and pipe, i.e. to maximize the aperture available for the beam. To enforce the zero orbit and angle, BPMs close to the specified element positions were corrected to zero. If two BPMs are corrected to zero, and no source of closed orbit perturbation is situated between them, it then follows that the perturbation between them is zero.

The knobs to be implemented were:

- `IP_CROSSING`: Implement a crossing angle (up to ± 295 μrad) in the horizontal plane for both beams in `IP5`. Allowed correctors: every corrector from `IP5` up to and including the ones in `Q4` on each side.
- `IP_SEPARATION`: Implement a separation of the beams (up to ± 0.75 mm) at `IP5` in the vertical plane. Allowed correctors: every corrector up to `Q4`, excluding all correctors of type `MCBY`.
- `CC_SEPARATE_B1/CC_SEPARATE_B2`: Implement an orbit at the crab cavities (up to ± 1 mm) in the horizontal and vertical plane, independently for each beam. Allowed correctors: all correctors up to `Q6` on each side.

- `OFFSET_REMOTE_ALIGNMENT`: Implement an offset of both beams (up to ±2 mm) in the horizontal and vertical plane at `IP5` when offsetting every quadrupole from `Q4` on the left to `Q4` on the right of `IP5` by ±2 mm, and the quadrupole at `Q5` by ±1 mm. Allowed correctors: every corrector from `Q4` to `Q8` on each side, excluding all correctors of type `MCBRD`.
- `LUMISCAN_B1/LUMISCAN_B2`: Implement an orbit at `IP5` (up to ±100 μm) in the horizontal and vertical plane for each beam, independently. Allowed correctors: every corrector from `Q4` to `Q5`, on each side of `IP5`.

For all knobs an additional optimization goal for the implementation was to minimize the use of the `MCBY` correctors and the residual orbit at the crab cavities.

The corrector budget, i.e. how much corrector strength in Tm is allotted to each corrector, is given in Table 4.1. The corrector strength allotted to the knob implementation was

Table 4.1: The orbit corrector budget in HL–LHC. Corrector names are provided as regular expressions and corrector strength is given in units of Tm.

Corrector name	Corrector strength limit [Tm]
`MCB[HV]`	1.895
`MCBC[HV].9`	1.895
`MCBC[HV].[78]`	2.8
`MCBC[HV].[56]`	2.1
`MCBY[HV].[AB]?4`	2.25
`MCBRD[HV]`	5.0
`MCBXFA[HV].3`	4.5
`MCBXFB[HV].[AB][12]`	1.87
`MCBW[HV]`	1.87

chosen to be the orbit corrector budget, minus the 2*RMS corrector strength used for the error correction. Other measures than the 2*RMS value could have been taken as the reference, e.g. the worst-case over a number of samples or another multiple of the root-mean-square. An argument for choosing the two-time root-mean-square value is experience; it was used for similar modeling of LHC where it was a good predictor for the corrector strength used [20].

4.2.2 Error Correction Procedure

For the error correction, two error correction methods were implemented and used:

1. Linear correction based on a weighted pseudoinverse of the corrector response matrix.
2. Constrained convex optimization in `cvxpy` based on (4.2).

The first method constitutes a direct approach and has been used before [3]. It however has the drawbacks of not being adaptable (the same mapping will be applied to the most pathological as well as the most easily corrected machines) and lacking functionality for mapping weights and singular value cutoffs to given constraints. Having said this, it can serve as a benchmark for the performance of the convex optimization. The methods are specified next.

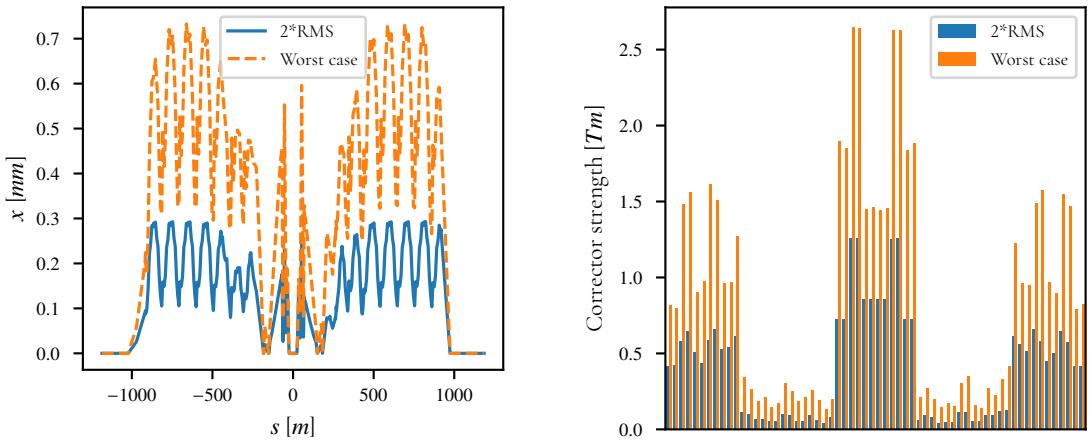
Linear Correction

For the linear correction, 50 out of 57 singular values was used for each plane, and BPMs were weighted as follows:

- BPMs that were to be corrected to zero were weighted as 1.0.
- Remaining BPMs in the IR were weighted as $1e{-}5$.
- Remaining BPMs in the arc were weighted as $1e{-}6$.

where the 1.0 weights serve to enfore the equality constraints, and the remaining weights are used to achieve a feasible residual orbit less than $1\ \text{mm}$ at the remaining BPMs.

Applying this linear correction to the given error distribution results in Figure 4.1.



(a) Residual orbit.

(b) Corrector use.

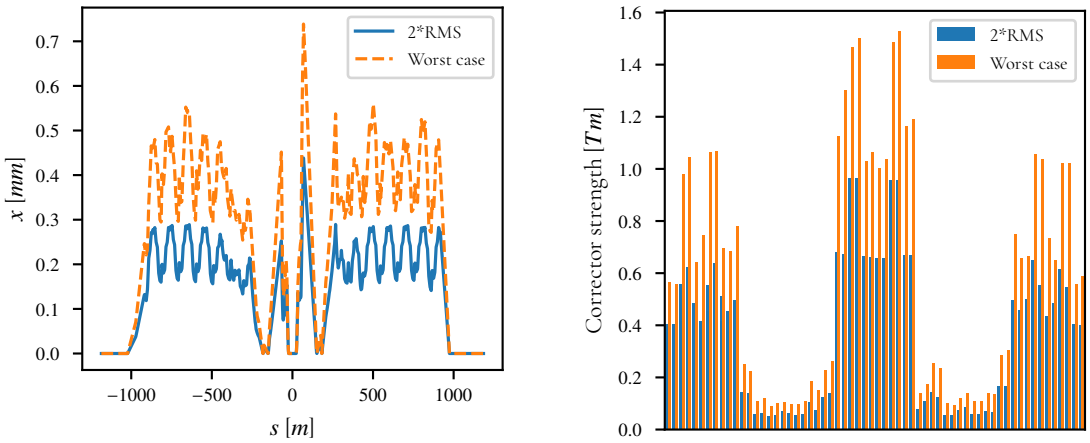
Figure 4.1: The residual orbit **(a)** and corrector strength **(b)** for the linear correction, given as 2*RMS and worst-case. Residual is given for Beam 1 in the horizontal plane, and corrector strength is plotted for all correctors up to `Q10` on each side where each pair of bars corresponds to a corrector.

Convex Optimizer Correction

For the optimization of the error correction, the convex optimizer was used where

1. The orbit was forced to zero at the BPMs around critical points.
2. The orbit constraint was set to $1\ \text{mm}$ at every other BPM in the segment.
3. Each corrector was constrained to use at most $2\ \text{Tm}$.
4. The weight for the regularizing term (corresponding to $\|\mathbf{c}\|_2$) was set to $2e{-}3$.

A point worth making is that even if the constraints could be made stricter, the optimizer will try to minimize both the corrector strength usage and the residual orbit. As such, even if the constraints are lax, the orbit will ideally be minimized to the point where none of the constraints are active. Unlike for a linear mapping where the worst-case and RMS result can be computed analytically, the convex optimizer has to estimate the RMS and worst-case values. To do this, 20000 samples were drawn from the machine error distribution and were attempted to be corrected by the optimizer within the given constraints. Out of the 20000 samples, none failed to be corrected within the set limits. The estimated RMS and worst-case orbits along with corrector strength can be seen below in Figure 4.2.



(a) Residual orbit.

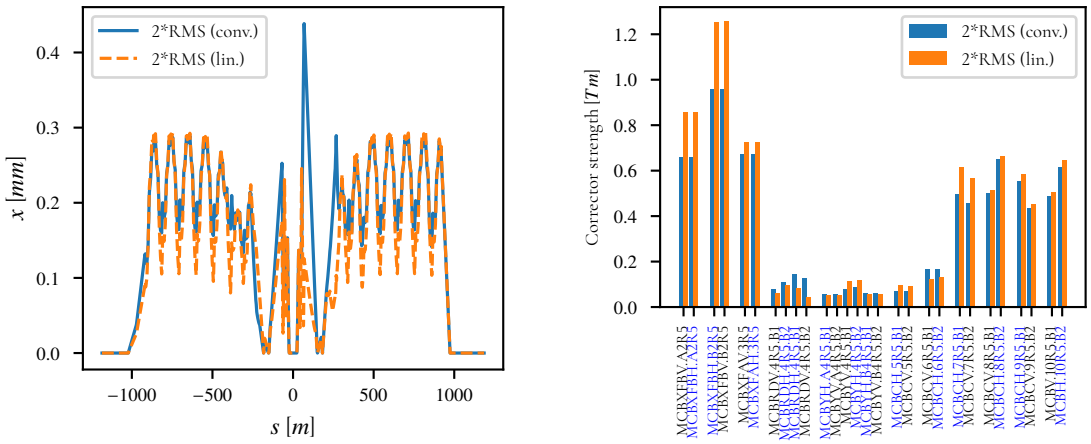
(b) Corrector use.

Figure 4.2: The residual orbit **(a)** and corrector strength **(b)** for the convex correction, given as 2*RMS and worst-case. Residual is given for Beam 1 in the horizontal plane, and corrector strength is plotted for all correctors up to `Q10` on each side where each pair of bars corresponds to a corrector.

4.2.3 Comparison of Corrections

For the given settings, the most important comparison between the convex and linear correction is that of the RMS orbit and corrector strength used. These comparisons can be seen in Figure 4.3, where the correctors are plotted on the right-hand side of `IP5` up to Q10. The linear correction corrects more aggressively, producing a smaller residual while also using more corrector strength. This result is however predicated on the parameterization of the linear correction and the convex optimizer. Both methods here provide corrections that are adequate under the given constraints. If one were interested in correcting more aggressively with the convex optimizer, then it would be necessary to decrease the weight of the regularizing term in the objective function, or to force it by setting harder constraints on the orbit. Note however that enforcing harder constraints could render the correction of some machines infeasible.

The worst case scenarios depicted in Figures 4.2 and 4.1 have to be interpreted as the 'worst scenario per point', where a scenario corresponds to a configuration of machine er-



(a) Residual orbit.

(b) Corrector use.

Figure 4.3: Comparison of 2*RMS residual orbit **(a)** and corrector strength **(b)** between the linear and the convex correction. Residual is given for Beam 1 in the horizontal plane, whereas the corrector strengths shown are those of the right-hand side of `IP5` up to `Q10`.

rors. Unlike RMS values which are easily estimated, extreme values of a distribution are in general not. The linear correction is a special case where the worst case can be analytically computed as per (4.10), and it gives an idea why it is hard to estimate directly from the distribution. The worst case error configuration for any given quantity will correspond to each error attaining either maximum or minimum value, and is therefore virtually impossible to attain by repeated sampling from the error configuration distribution. Nevertheless, the worst case outcome over a large number of samples can still be a meaningful indicator of the worst case scenarios one is likely to encounter in practice.

4.2.4 Knob Implementation

Given an error correction, the 2*RMS corrector strength for the correction was detracted from the total available corrector strength, which was then defined to be the corrector budget available for the knob implementation. The optimizer for the knob implementation was run with optimization goal of minimizing the use of the `MCBY` correctors and with added regularizing term for minimizing the orbit at the crab cavities. As the error correction study gave two valid error correction strategies with similar corrector strength usage, both are valid for the knob implementation. Using either one for the knob implementation gave a feasible corrector configuration, and the results from using the convex correction can be seen in Figures 4.4 and 4.5. The result demonstrates that the `MCBY` correctors are used sparingly and several constraints are active (bars touching the limits), in line with what is to be expected when an optimum has been found.

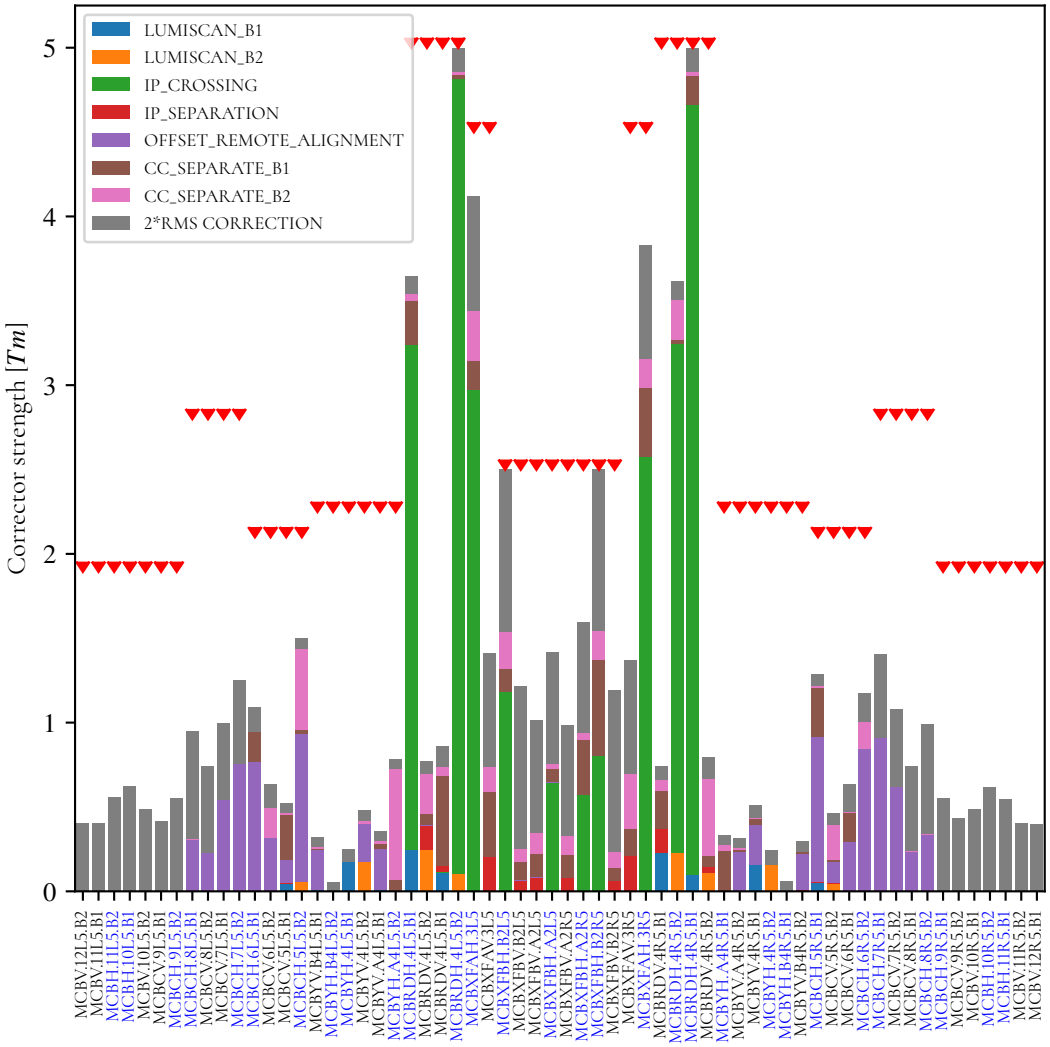


Figure 4.4: Bar plot of the corrector strength used for implementing all knobs and a 2*RMS error correction. Red triangles represent the corrector strength limit for each corrector. Shown for all correctors up to `Q10` on both sides of `IP5`.

4.3 Conclusion

Under the given assumptions the orbit corrector budget holds, in line with preliminary verifications [15]. There is a sufficient amount of orbit corrector strength allotted for error correction and implementing the listed knobs for the HL–LHC optics version 1.5, release candidate 0. For this iteration of the orbit corrector budget problem, both linear and convex corrections provided feasible solutions, but this need not be the general case. If constraints are hard enough then it is possible that the flexibility of the convex optimizer is necessary to produce a feasible solution. Moreover, as the constraints get harder the parameterization of the linear correction gets more constrained. Since there is no reliable algorithm currently available for finding feasible linear correction strategies subject to constraints, it follows that correcting based on linear correction strategies would not scale well with harder constraints.

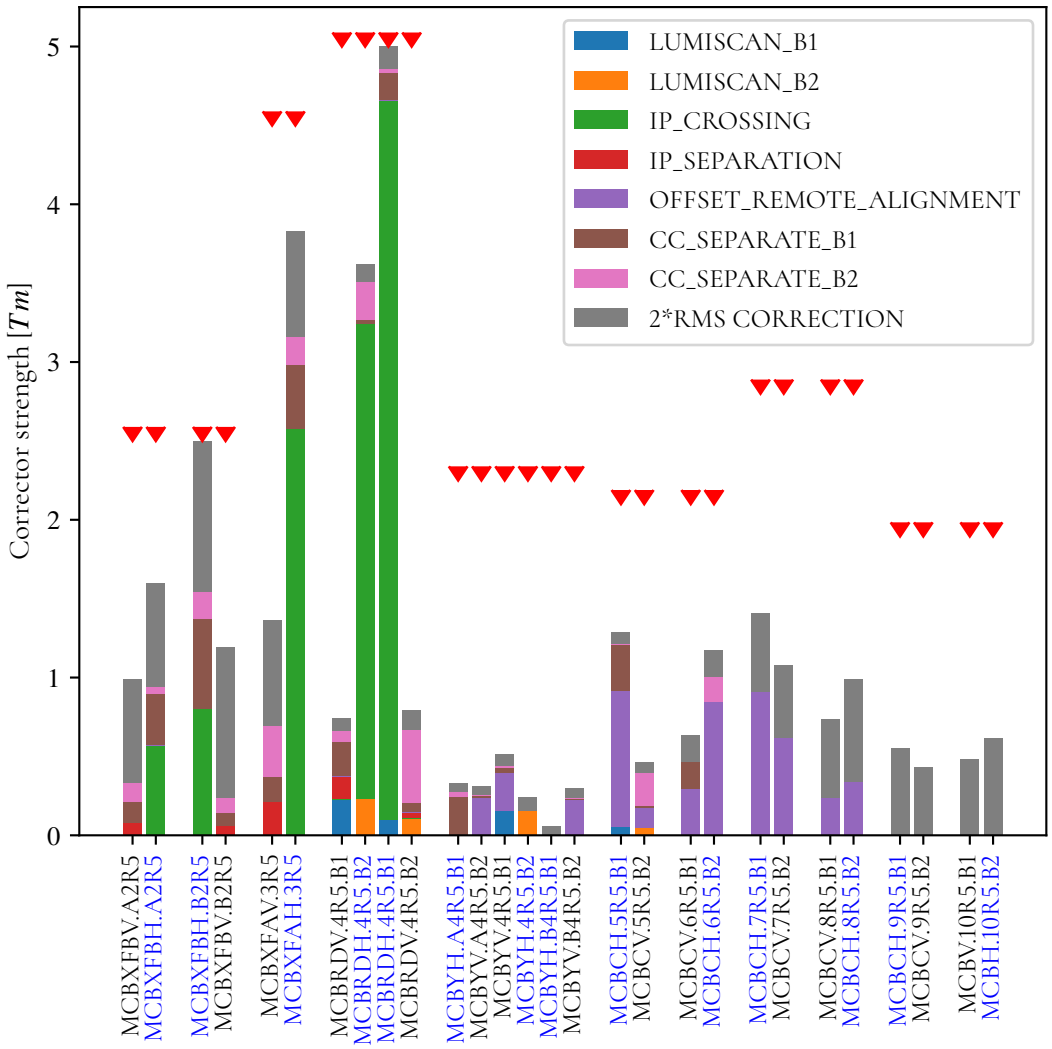


Figure 4.5: Bar plot of the corrector strength used for implementing all knobs and a 2*RMS error correction. Red triangles represent the corrector strength limit for each corrector. Shown for all correctors up to `Q10` on the right-hand side of `IP5`.

Chapter 5

Estimating Impact of Orbit Feedback System for Maintaining Collision

Currently in LHC there is a feedback system employed for stabilizing the closed orbit during operation, referred to as the *Orbit Feedback System* (OFB) [18, 19]. It is run throughout the different phases of the machine cycle, from beam injection to dump at the end of collision. Here the focus will be on the phase 'stable beams', when beams are in collision.

During stable beams the global orbit is not a concern; the order of the usual closed orbit perturbation accrued over stable beams is too small to cause any beam loss. Exceptions to this are abrupt events (e.g. great ground motion, or equipment faults), but they are outliers and not the topic of this analysis. The issue is instead that the typical perturbation causes the beams to drift away from each other at the collision points, which means less luminosity. The object of analysis is therefore not the usual global orbit but rather the beam separation at the collision points, here with a focus on `IP5`.

HL–LHC will also have a dedicated feedback system for closed orbit stabilization, but its final design is yet to be decided. As a first approximation of the feedback system in HL–LHC, we will transfer the OFB as defined in LHC to HL–LHC and compare it to the current implementation. This allows for some insight to be gained into how the OFB impacts luminosity and how this can be expected to scale for HL–LHC.

5.1 Problem Formulation

New concepts have to be introduced in order to adequately treat the subject. The notion of imperfect BPM readings will be accounted for together with a model for the effect of an orbit feedback system during stable beam conditions, followed by an overview of the OFB performance in LHC.

5.1.1 Imperfect BPMs

In terms of nomenclature, we define the signal $\mathbf{y}$ measured at BPMs as

$$\mathbf{y} = \mathrm{RM_e}^{\dagger}\mathbf{e} + \mathbf{w}\,, \tag{5.1}$$

where $\mathbf{e}$ represents machine errors inducing closed orbit perturbation and $\mathbf{w}$ is noise added to the signal, i.e. BPM reading errors. Assuming a linear correction strategy $\boldsymbol{A}$, the corrector strength used to correct for a given signal can be written as

$$\mathbf{c} = \mathrm{A}\mathbf{y} = \mathrm{A}(\mathrm{RM_e}^{\dagger})\mathbf{e} + \mathrm{A}\mathbf{w} = \mathrm{B_c^e}\mathbf{e} + \mathrm{B_c^w}\mathbf{w}\,, \tag{5.2}$$

where the analogous equation for the residual turns out as

$$\begin{aligned}\mathbf{r} = \mathrm{RM_e}\mathbf{e} + \mathrm{RM_c}\mathbf{c} &= (\mathrm{RM_e} + \mathrm{RM_c}\mathrm{ARM_e}^{\dagger})\mathbf{e} + \mathrm{RM_c}\mathrm{A}\mathbf{w} \\ &= \mathrm{B_r^e}\mathbf{e} + \mathrm{B_r^w}\mathbf{w}\,.\end{aligned} \tag{5.3}$$

Looking at (5.3), one can obtain some insight on what constitutes a good linear correction strategy. In particular, it should satisfy:

1. $||(\mathrm{RM_e} + \mathrm{RM_c}\mathrm{ARM_e}^{\dagger})\mathbf{e}||_2$ is minimal, i.e. A is effective at inverting perturbations caused by machine errors and sampled at BPMs.

2. $||\mathrm{RM_c}\mathrm{A}\mathbf{w}||_2$ is minimal, i.e. A is ineffective at inverting the noise reported by BPMs.

The relative importance of these points depends on the distributions of $\mathbf{e}$ and $\mathbf{w}$. If the machine errors are considerably smaller than the reading errors then the best correction strategy is to correct cautiously, and for the inverse the best approach is to correct aggressively. To make this rigorous, we let the linear correction strategy be the standard pseudoinverted corrector matrix as per

$$\mathrm{A} = -\,\mathrm{pinv}(\mathrm{RM_c}^{\dagger})\,. \tag{5.4}$$

As the number of singular values is variable for the pseudoinverse, there is a relation between the number of singular values and the characteristics of the correction. Large singular values in the corrector response matrix correspond to unit-length corrector strength vectors (left-singular vectors) which induce large closed orbit (right-singular vectors multiplied by their singular values), and vice versa for small singular values. This is showcased in Figure 5.1. Smaller singular values correspond to basis vectors in orbit space which are localized and exhibit noisy behaviour, conversely, greater singular values impact larger sections of the ring with more physical characteristics. In constructing A, if only a few singular values are used, we call the correction *global* and if many are used it is instead referred to as *local*.

Returning to the notion of imperfect BPMs, since smaller singular values correspond to noisier basis vectors in orbit space, it follows that local corrections will propagate more noise from BPMs to the closed orbit. Viewed differently, the more basis vectors in orbit space one uses, the better reconstruction of an arbitrary orbit vector, including unphysical orbits.

Recalling Chapter 4 it was shown that a linear correction strategy provides simple equations for computing the covariance of the correction assuming the underlying error has a known distribution. Analogously, a linear correction strategy for a noisy signal leads to linear mappings in $\mathbf{e}$ and $\mathbf{w}$ for the corrector strength employed and residual after correction.

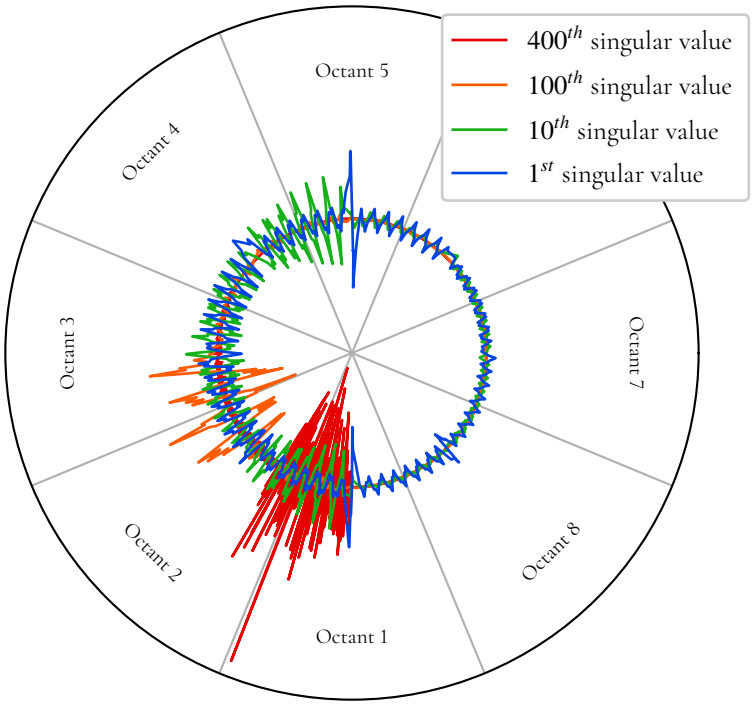


Figure 5.1: Right-singular vectors of $\mathbf{RM_c}$ for Beam 1 in the horizontal plane, where the singular values are sorted by their size in decreasing order. Each singular vector shown has L^2norm equal to 1.

Assuming that $\mathbf{e}$, $\mathbf{w}$ are random uncorrelated vectors with corresponding covariance matrices Σ_e, Σ_w, respectively, the covariance matrices for $\mathbf{c}$ and $\mathbf{r}$ can be directly derived as

$$\begin{aligned} \Sigma_c &= \mathrm{B_c^e}\Sigma_e(\mathrm{B_c^e})^\intercal + \mathrm{B_c^w}\Sigma_w(\mathrm{B_c^w})^\intercal\,, \\ \Sigma_r &= \mathrm{B_r^e}\Sigma_e(\mathrm{B_r^e})^\intercal + \mathrm{B_r^w}\Sigma_w(\mathrm{B_r^w})^\intercal\,. \end{aligned} \tag{5.5}$$

In the simplest model of the correction problem, one can assume *i.i.d* (independently and identically distributed) zero-mean distributions for the BPM reading errors and machine errors, respectively. Assuming that ground motion is the dominating source of orbit perturbation, the transverse misalignment of quadrupoles can be perceived as i.i.d., in which case (5.5) can be written as:

$$\begin{aligned} \Sigma_c &= \sigma_{quad}^2(\mathrm{B_e^c})^{\dagger_q}\{(\mathrm{B_e^c})^{\dagger_q}\}^\intercal + \sigma_{BPM}^2\mathrm{B_w^c}(\mathrm{B_w^c})^\intercal\,, \\ \Sigma_r &= \sigma_{quad}^2(\mathrm{B_e^r})^{\dagger_q}\{(\mathrm{B_e^r})^{\dagger_q}\}^\intercal + \sigma_{BPM}^2\mathrm{B_w^r}(\mathrm{B_w^r})^\intercal\,, \end{aligned} \tag{5.6}$$

where $\dagger_q$ corresponds to taking the subset of columns corresponding to transverse misalignment of quadrupoles, and where σ_{quad}, σ_{BPM} are the standard deviations for quadrupole transverse misalignment, assumed equal for both transverse planes, and BPM reading errors respectively. Using (5.6) it becomes possible to evaluate linear correction strategies based only on the relation between σ_{quad} and σ_{BPM}.

5.1.2 Orbit Correction over Time

Consider orbit correction on the time interval $t \in [0, T]$ for a machine error $\mathbf{e}(t)$ that evolves continuously in time and where BPM readings are i.i.d. Now, assume that on this interval n corrections are performed at times $t_1 < t_2 < \ldots < t_{n-1} < t_n$. Let $\mathbf{w}_i$ be the BPM reading error at time t_i and define the corrector strength and residual directly after performing the i^{th} correction as $\mathbf{c}_i$ and $\mathbf{r}_i$, respectively. Let $\mathrm{A} = -\mathbf{pinv}(\mathrm{RM_c}^\dagger)$ be the linear correction strategy employed for a given number of singular values in the pseudoinverse, and define $\mathrm{P} = -\mathrm{RM_c}^\dagger \mathrm{A} = \mathrm{RM_c}^\dagger \mathbf{pinv}(\mathrm{RM_c}^\dagger)$. Recalling Chapter 2, P is an orthogonal projection, whereby it follows that $\mathrm{P}^2 = \mathrm{P}$ and $\mathrm{A}(\mathrm{I} - \mathrm{P}) = 0$. Granted this, the following holds:

$$
\begin{aligned}
\mathbf{c}_1 &= \mathrm{A}(\mathrm{RM_e}^\dagger \mathbf{e}_1 + \mathbf{w}_1)\,, \\
\mathbf{r}_1 &= \mathrm{RM_e}^\dagger \mathbf{e}_1 + \mathrm{RM_c}^\dagger \mathbf{c}_1 \\
&= (\mathrm{I} + \mathrm{RM_c}^\dagger \mathrm{A})\mathrm{RM_e}^\dagger \mathbf{e}_1 + \mathrm{RM_c}^\dagger \mathrm{A}\mathbf{w}_1 \\
&= (\mathrm{I} - \mathrm{P})\mathrm{RM_e}^\dagger \mathbf{e}_1 - \mathrm{P}\mathbf{w}_1\,, \\
\mathbf{c}_2 &= \mathbf{c}_1 + \mathrm{A}\Big(\mathrm{RM_e}^\dagger \mathbf{e}_2 + \mathrm{RM_c}^\dagger \mathbf{c}_1 + \mathbf{w}_2\Big) \\
&= \mathrm{A}\Big[(\mathrm{I} - \mathrm{P})\mathrm{RM_e}^\dagger \mathbf{e}_1 + \mathrm{RM_e}^\dagger \mathbf{e}_2 + (\mathrm{I} - \mathrm{P})\mathbf{w}_1 + \mathbf{w}_2\Big] \\
&= \mathrm{A}\Big(\mathrm{RM_e}^\dagger \mathbf{e}_2 + \mathbf{w}_2\Big)\,, \\
\mathbf{r}_2 &= \mathrm{RM_e}^\dagger \mathbf{e}_2 + \mathrm{RM_c}^\dagger \mathbf{c}_2 \\
&= (\mathrm{I} - \mathrm{P})\mathrm{RM_e}^\dagger \mathbf{e}_2 - \mathrm{P}\mathbf{w}_2\,, \\
&\vdots \\
\mathbf{c}_n &= \mathrm{A}(\mathrm{RM_e}^\dagger \mathbf{e}_n + \mathbf{w}_n)\,, \\
\mathbf{r}_n &= (\mathrm{I} - \mathrm{P})\mathrm{RM_e}^\dagger \mathbf{e}_n - \mathrm{P}\mathbf{w}_n \\
&= (\mathrm{I} + \mathrm{RM_c}^\dagger \mathrm{A})\mathrm{RM_e}^\dagger \mathbf{e}_n + \mathrm{RM_c}^\dagger \mathrm{A}\mathbf{w}_n\,,
\end{aligned}
\tag{5.7}
$$

The important point to note in (5.7) is that the residual and corrector strength used directly after the i^{th} correction depends only on the machine error and BPM reading at that point in time; there is no memory with respect to previous corrections. If one studies the continuous evolution of the residual, then naturally each correction is visible on the interval, but for any given point in time $t \in [0, T]$ it is only the latest correction that impacts the residual at that point. To underline this point, we consider an example.

Assume that a quadrupole, initially at rest, moves horizontally as described by a *Brownian motion*, that is, it is continuous, its movement during any time-interval can be described by a zero-mean Gaussian distribution with variance linearly proportional to the timespan, and disjoint intervals are independent. Figure 5.2 shows a numerical example where a quadrupole moves like a Brownian motion and the closed orbit is tracked over time at an arbitrary position. The plot demonstrates the previous statement: when considering the evolution of closed orbit in the machine over time, it is only the latest correction that has an impact. The statement remains valid when including BPM errors: it is only the latest BPM reading error that enters the residual.

Using (5.7) with the residual evaluated for the whole machine, it is possible to compute

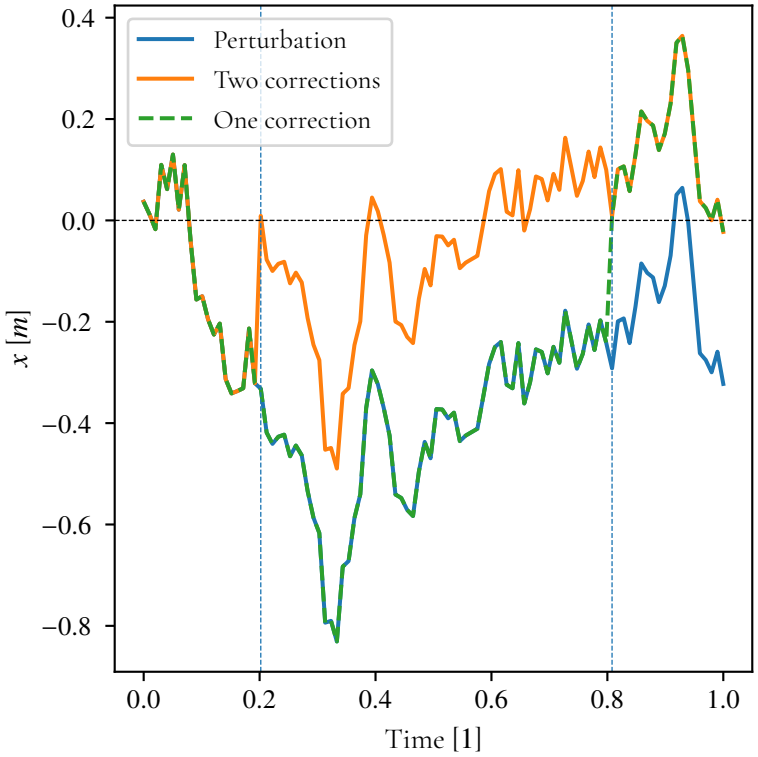


Figure 5.2: Horizontal closed orbit induced by a quadrupole following a Brownian motion, measured at an arbitrary element. The raw perturbation is shown alongside two correction scenarios: one where the orbit is corrected once at $t = 0.8$ and another where the orbit is corrected at $t = 0.2$ and $t = 0.8$, all with the same correction strategy and perfect BPMs.

the covariance matrices directly after the i^{th} correction:

$$\begin{aligned} \Sigma_c(t_i) &= \mathrm{B}_\mathrm{c}^\mathrm{e}\Sigma_e(t_i)(\mathrm{B}_\mathrm{c}^\mathrm{e})^\intercal + \mathrm{B}_\mathrm{c}^\mathrm{w}\Sigma_w(\mathrm{B}_\mathrm{c}^\mathrm{w})^\intercal\,, \\ \Sigma_r(t_i) &= \mathrm{B}_\mathrm{r}^\mathrm{e}\Sigma_e(t_i)(\mathrm{B}_\mathrm{r}^\mathrm{e})^\intercal + \mathrm{B}_\mathrm{r}^\mathrm{w}\Sigma_w(\mathrm{B}_\mathrm{r}^\mathrm{w})^\intercal\,. \end{aligned} \tag{5.8}$$

which is the same as (5.5) except for a time-dependence in the covariance matrix of the machine errors. Summarizing, to investigate the corrector strength usage and closed orbit residual after having performed several corrections over time, it is sufficient to consider only the very last correction. If the distributions of the machine and BPM reading errors are known at the time of a correction, then this information suffices for computing the residual at that point in time.

5.1.3 Orbit Feedback System in LHC

A *fill* in LHC denotes the full sequence of injecting beams into LHC, ramping them up to collision energies, colliding and then dumping them. A simplified overview of a fill can be seen below in Figure 5.3.

The OFB itself is not a trivial construct; it is a full-fledged feedback system. Despite this, in the first-order, it can be described as doing the following, iteratively:

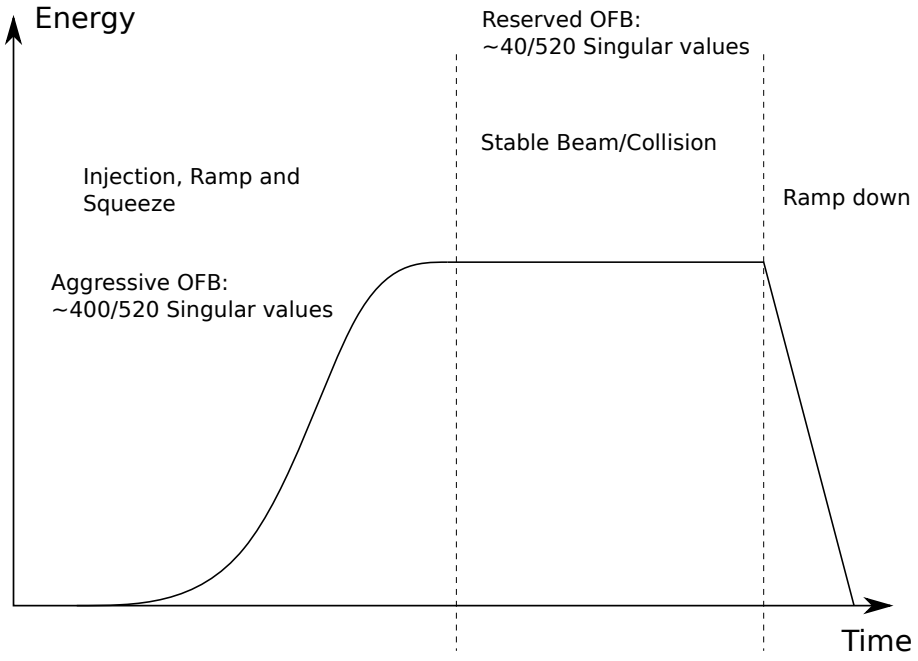


Figure 5.3: An overview of the different stages of a fill in LHC along with approximate information of the number of singular values employed currently by the orbit feedback in LHC.

1. Retrieve reading of current closed orbit $\mathbf{y}$ from BPM system.
2. Compute the difference with respect to a given reference closed orbit, Δy.
3. Compute necessary corrector strength update via a linear correction strategy as per $\Delta\mathbf{c} = \mathrm{A}\Delta\mathbf{y}$.
4. Apply the correction multiplied by a gain (typically $\ll 1$) over a period of time to optimize for overshoot, response time and robustness.

Based on the current performance of LHC, the OFB is currently run aggressively (many singular values, i.e. local correction) during the ramp and leniently (few singular values, i.e. global correction) during collision [21]. A heuristic as to why the OFB is run aggressively for the ramp is that the beam optics and energy are changed significantly during that stage. It is a complex phase and so more sources of errors are introduced, e.g. transient currents and hysterisis-like effects in magnets when changing beam energy and machine optics. In contrast to this, during stable beams the beam optics and energy remain the same, wherefore there is less orbit perturbation and so less need for corrections.

5.2 Results

To model the effect of the OFB for stable beams, the following data and assumptions were used:

1. The OFB does not use any correctors that are shared between the beams [21].

2. The OFB uses 40 out of 520 singular values (per transverse plane) for the pseudoinverse of the corrector response matrix [21].

3. All closed orbit perturbation taking place during collision is, or can be treated as being, caused by transverse movement of quadrupoles.

4. All transverse quadrupoles movement is i.i.d. in each dimension.

5. All BPM reading errors are i.i.d. in each dimension.

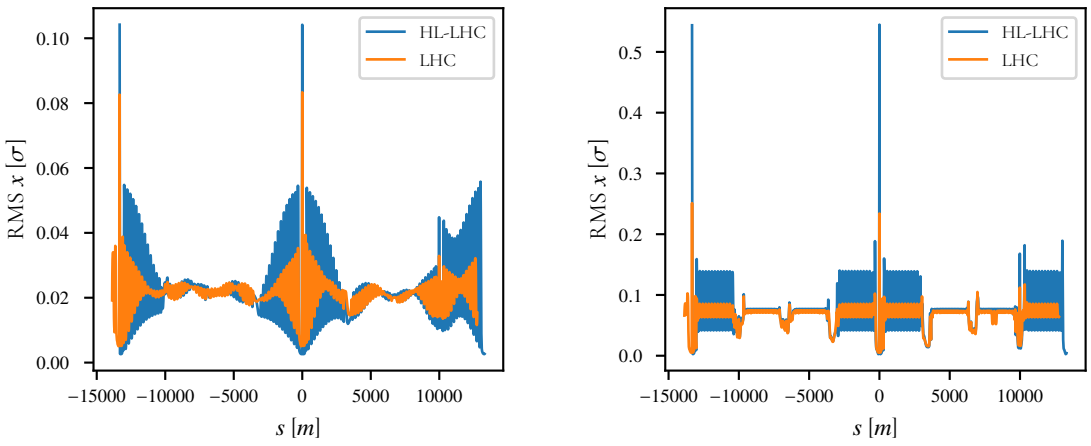
6. As reference values, $\sigma_{quad} = 0.3\,\mu\text{m}$ is taken as the RMS quadrupole movement during a typical collision time of 12 hours and $\sigma_{BPM} = 20\,\mu\text{m}$ as the RMS BPM error for one reading [22].

7. Equation 5.6 is used to compute the residual at the end of collision, where $\mathbf{A} = -\text{pinv}(\mathbf{RM_c}^{\dagger})$ with the proper number of singular values.

8. LHC optics used: $\beta^* = 30\,\text{cm}$, $\epsilon_N = 2.5\,\mu\text{m}$ and $E = 6.5\,\text{TeV}$.

9. HL–LHC optics used: $\beta^* = 15\,\text{cm}$, $\epsilon_N = 2.5\,\mu\text{m}$ and $E = 7\,\text{TeV}$.

Given the previous assumptions, an example result for the residual orbit in the full machine can be seen in Figure 5.4 where LHC and HL–LHC are compared when using 40 and 400 singular values for the linear correction strategy $\mathbf{A}$ in the OFB. The plots shown in Figure 5.4 are normalized in units of beam size. Recalling the definition of beam size in (2.89), it is proportional to the square-root of the β-function, which provides the explanation for the peaks at `IP1` and `IP5` where the β-function assumes its smallest value. The relative size



(a) 40 singular values.

(b) 400 singular values.

Figure 5.4: The RMS horizontal horizontal residual for Beam 1 at the end of collision using the OFB, measured in beam size and compared between LHC and HL–LHC for $\sigma_{quad} = 0.3\,\mu\text{m}$, $\sigma_{bpm} = 20\,\mu\text{m}$ for 40 **(a)**, and 400 **(b)** singular values used in $\mathbf{A}$. $S = 0$ corresponds to `IP5` and the leftmost peak corresponds to `IP1`.

of these peaks, however, cannot be fully explained by the relative values of the β-function at these points, as the β-function over the full machine plays a role. For HL–LHC, the β-function in the interaction regions is greater, which has the impact that quadrupole misalignments in these regions have a greater impact on the closed orbit at every other position, including the interaction points. In addition to this, the normalized emittance is the same for the two machines, but as the collision energy is greater in HL–LHC, it has lesser geometric emittance, which reduces the beam size and therefore further constributes to the discrepancy between the machines.

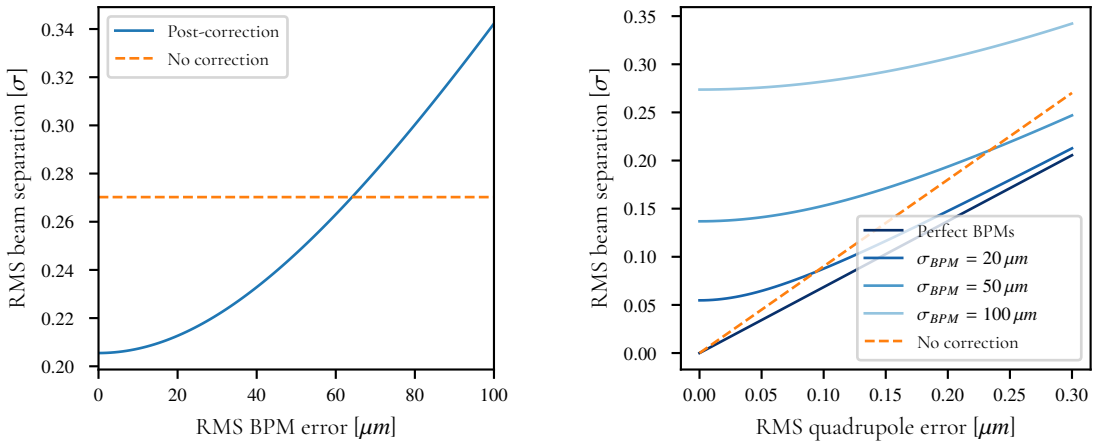
Overall, the results in 5.4a are expected: there is a worsening of the orbit at `IP1` and `IP5` where the beam size has been shrunk in HL–LHC. To quantify why the beams are not corrected more aggressively, see Figure 5.4b where the same scenario is plotted but for 400 singular values in $\mathbf{A}$. In essence, the errors in the BPM readings propagate to the orbit, worsening it substantially, unless the number of singular values is constrained for stable beams. Considering the relatively small orbit perturbations taking place during stable beams, this was to be expected.

Using (5.6) with minor modifications, it is possible to compute the RMS beam separation d (measued in beam size σ) at `IP5` for the different correction strategies in LHC and HL–LHC:

$$\begin{aligned}
d_{40}^{LHC} &= \sqrt{4.69\text{e}11\,\frac{\sigma^2}{m^2}\times\sigma_{quad}^2 + 7.49\text{e}6\,\frac{\sigma^2}{m^2}\times\sigma_{BPM}^2}\,,\\
d_{400}^{LHC} &= \sqrt{5.18\text{e}11\,\frac{\sigma^2}{m^2}\times\sigma_{quad}^2 + 4.87\text{e}8\,\frac{\sigma^2}{m^2}\times\sigma_{BPM}^2}\,,\\
d_{40}^{HLLHC} &= \sqrt{7.47\text{e}11\,\frac{\sigma^2}{m^2}\times\sigma_{quad}^2 + 1.56\text{e}7\,\frac{\sigma^2}{m^2}\times\sigma_{BPM}^2}\,,\\
d_{400}^{HLLHC} &= \sqrt{6.38\text{e}11\,\frac{\sigma^2}{m^2}\times\sigma_{quad}^2 + 2.87\text{e}9\,\frac{\sigma^2}{m^2}\times\sigma_{BPM}^2}\,.
\end{aligned} \tag{5.9}$$

Note that the scaling factors are functions of the response matrices, defined by the optics, and the number of singular values used in the correction strategy. As expected, a greater number of singular values amplifies the impact of BPM noise. Somewhat unexpectedly, 400 singular values performs worse than 40 singular values when correcting for beam separation caused by quadrupole errors in LHC. A possible explanation for this is the correction strategy being an unweighted corrector response matrix, meaning that the extra singular vectors included for the 400 singular values correction might prioritize other regions and not the two BPMs surrounding `IP5`. This effect need not be symmtetrical across the machines as small variations in the optics may shift around the singular vectors considerably, and so the difference between the machines could come down to structural differences arising from the high-luminosity upgrade.

In Figure 5.5 we show the beam separation at `IP5` in LHC for 40 singular values, and different values of the RMS quadrupole movement and BPM noise using (5.9). A few easy observations can be made for Figure 5.5. Firstly, the RMS BPM error has to be kept below roughly $60\,\mu\text{m}$ if the OFB is to have any ameliorating effect on the RMS beam separation at `IP5`. Secondly, even for perfect BPMs the OFB is only able to reduce the beam separation by roughly $20\,\%$ for the given magnitude of the quadrupole error. This is not unexpected, and is a direct result from the correction strategy being global and unweighted. If the OFB



(a) Versus σ_{BPM}.

(b) Versus σ_{quad}.

Figure 5.5: The RMS beam separation measured in beam size at `IP5` for the end of collision using the OFB with 40 singular values, plotted versus σ_{BPM} for $\sigma_{quad} = 0.3\,\mu$m **(a)**, and versus σ_{quad} for several values of σ_{BPM} **(b)**.

is to have the ability to considerably correct for beam separation, then it needs to weight the corrector response matrix and possibly change the number of singular values. In practice however, this added capability to correct for beam separation would necessarily propagate more BPM errors, and so the real bottleneck is the level of noise in the BPMs and not so much the choice of correction strategy.

It should be emphasized that the x-axes in Figure 5.5 should not be directly associated with time. Even if the assumption is made that the RMS quadrupole error increases linearly with time (contrary to e.g. a Brownian motion where it is proportional to the square-root), it would be a stretch to claim any continuity. Rather, the graphs should be seen as scans over possible quadrupole errors by the end of collision and what the respective impact on beam separation would be.

Having investigated the impact of the OFB in LHC on beam separation, Figure 5.6 shows a comparison between LHC and HL-LHC for the same settings. The important point here is that from the perspective of beam separation at `IP5`, HL–LHC will perform worse but comparably. In terms of just maintaining collision, even if no OFB is run for HL–LHC it would still be in collision as $0.4\,\sigma$ is still small enough for collisions to take place, even if luminosity is lost. This loss in luminosity is given by the exponential factor in (2.91):

$$\exp\left(-\frac{d^2}{4\sigma^2}\right) = \exp\left(-\frac{0.4^2\sigma^2}{4\sigma^2}\right) = 0.96\,. \tag{5.10}$$

In other words, a beam separation of $0.4\,\sigma$ would give a $4\,\%$ instantaneous luminosity loss by the end of collision for HL-LHC, implying a lower loss of integrated luminosity loss over a fill, but still not acceptable.

The beam operator in LHC possesses another tool for reducing the beam separation, the *lumiscan*. It is an orbit knob, recognizable as one of the knobs implemented in Chapter 4, and an implementation of it can be seen in Figure 5.7. This knob can be used to bring the beams into head-on collision, and does so by using the luminosity of the beam as a signal.

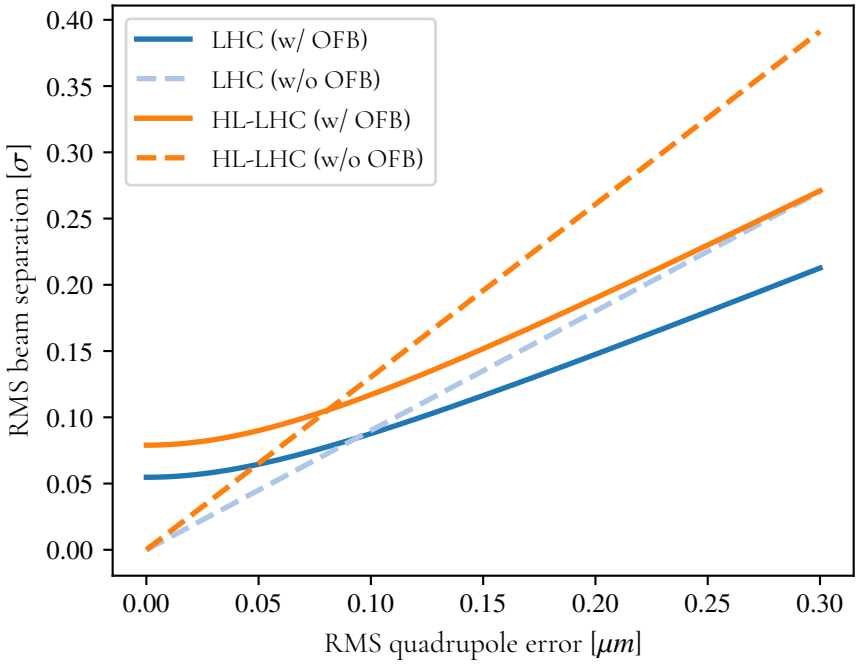


Figure 5.6: The RMS beam separation at `IP5` at the end of collision using the OFB with 40 singular values as a function of the RMS quadrupole error, measured in beam σ's for $\sigma_{bpm} = 20\ \mu$m where the RMS quadrupole error is scanned. Here plotted against the benchmark of using no correction.

In practice, this means steering one of the beams while the other one is kept stable, and measuring the luminosity as the first beam is moved until an optimum has been found, i.e. head-on collision. Importantly, the lumiscan is independent of BPM readings, as only the luminosity is used as signal.

During collision, lumiscans are called manually by an operator whenever the luminosity drops below what is expected. A critical property of the lumiscan is that it can be applied on top of the OFB, in other words, if a lumiscan is performed with the OFB active, the OFB does not cancel the orbit knob. This is a result of the OFB using a global correction, and the lumiscan modifying the orbit localized around interaction points. Due to the availability of the lumiscan, operating virtually independently of the OFB, beam separation can and is made smaller than what is predicted here for LHC. The results shown here for LHC should instead be viewed as the beam separation for an alternative operational scenario where lumiscans are disabled.

Lastly, it is worth mentioning that for Run 1 (2010-2013), LHC was run with the OFB inactive during collision and still proved stable [21]. This was however for other parameters; the collision energy was lower at 3.5 TeV and the beam sizes at collision were greater, and so that iteration of LHC was overall more resillient to closed orbit perturbation. Nevertheless, the lattice has not changed considerably since then, and so there are grounds for believing that LHC could remain operational in Run 3 without the OFB during collision.

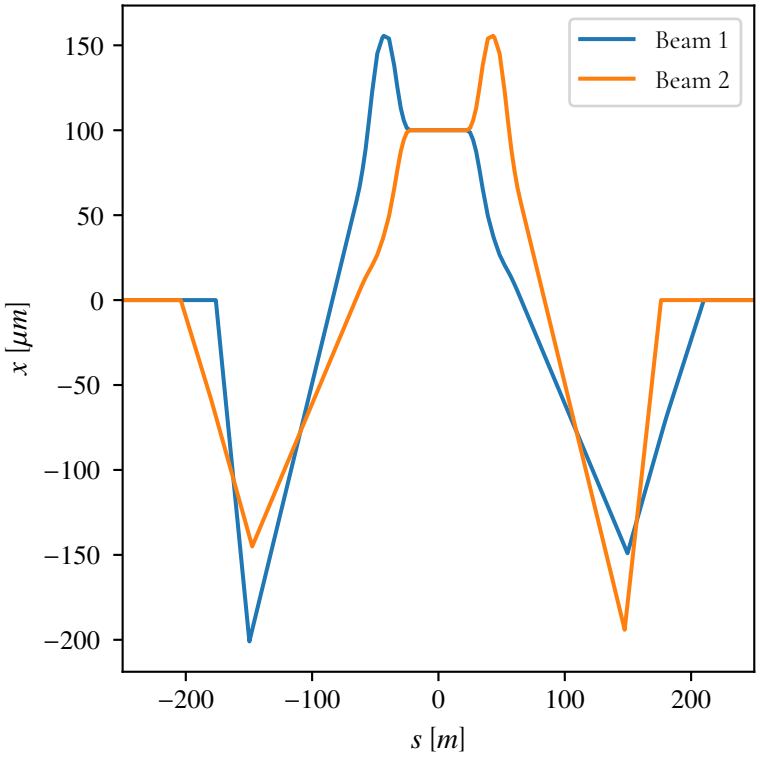


Figure 5.7: Implementation of the lumiscan orbit knob, shown for the horizontal plane around octant 5 where $s = 0$ is the position of `IP5`.

5.3 Conclusion

The results derived here indicate that the OFB used in LHC does not significantly improve the beam separation stability at `IP5` during collision. Transferring the OFB as currently implemented for the LHC to HL–LHC leads to similar results, with a worsening of beam separation driven in part by a reduced beam size at the collision points.

Chapter 6

BPM Specification for Local IP Correction

The integrated luminosity over a fill can be viewed as a benchmark for the performance of a collider, but the instantaneous luminosity also plays an important role for the experiments conducted. The *event pile-up* is a parameter describing the luminosity from a single bunch crossing, i.e. the number of events per bunch crossing. If too many events take place per bunch crossing, then the data collected by detectors may be saturated and harder to analyze. Granted this, if the integrated luminosity is treated as a constant, it is preferable to have a constant event pile-up. This can be achieved by β^**-levelling*.

To β^*-level is to change the β-function at an interaction point, i.e. β^*, during collision as a means of keeping the instantaneous luminosity constant, as per (2.91). In practice this is done in steps where the optics is changed to achieve the new β^*. At each step non-negligible closed orbit perturbation might be induced. Because of this, using a lumiscan might be necessary to bring the beams back into head-on collision. During LHC Run 2, a few β^*-levelling steps were performed, and at times, no lumiscans were necessary to recover head-on collision condition. In HL–LHC, the luminosity will be increased nominally by a factor five while the number of bunches is kept constant [23]. This will increase the event pile-up and so the β^*-levelling used in HL–LHC will be more involved with many more levelling steps, thus potentially increasing the number of lumiscans performed during collision.

Performing a lumiscan takes in the order of a minute [24], and in this process some luminosity is inevitably lost as the luminosity itself is used as the signal to optimize over. Given the new β^*-levelling scheme in HL–LHC, the integrated luminosity lost to lumiscans could increase considerably. Because of this luminosity loss, there have been calls for limiting the use of lumiscans, which warrants an alternative method for bringing beams into head-on collision. One such method could be to make use of the BPMs closest to the interaction point and correcting them to the design orbit using orbit correctors. Such a correction would require highly accurate BPMs close to the interaction points. The subject of this study is to estimate how accurate these BPMs would have to be in order to replace lumiscans.

6.1 Results

As lumiscans are performed to bring beams into head-on collision, any local IP correction employed as a substitute has to reduce the beam separation sufficiently. As a benchmark of this, we use $0.1\,\sigma$ beam separation, corresponding to a $0.25\,\%$ instantaneous luminosity loss. In other words, if a local IP corrrection can achieve a $0.1\,\sigma$ RMS beam separation, it is considered a valid substitute for lumiscans.

To study these corrections, the following setup was used and assumptions were made:

1. Only octant 5 is considered from `Q25` to `Q25`, with quadrupole errors from `Q20` to `Q20`.
2. All closed orbit perturbation is, or can be treated as being, caused by transverse movement of quadrupoles.
3. All quadrupoles' movement transversally is i.i.d. in each dimension.
4. All BPM reading errors are i.i.d. in each dimension.
5. As a reference value, $\sigma_{quad} = 0.3\,\mu\text{m}$ is taken as the RMS quadrupole movement to correct for during collision [22].
6. Equation 5.6 is used to compute the residual where $\mathbf{A} = -\,\text{pinv}(\mathbf{RM_c}^{\dagger})$, with the proper number of singular values.
7. HL–LHC optics used: $\beta^* = 15\text{ cm}$, $\epsilon_N = 2.5\,\mu\text{m}$ and $E = 7\text{ TeV}$.

The underlying assumption for studying one octant is that errors local to the octant are assumed to be the dominating source of errors at the corresponding interaction point. For the extent of this study, pairs of BPMs will be used where one pair is defined as per Figure 6.1. In other words, the i^{th} BPM pair is the pair of BPMs formed by taking the i^{th} BPM away

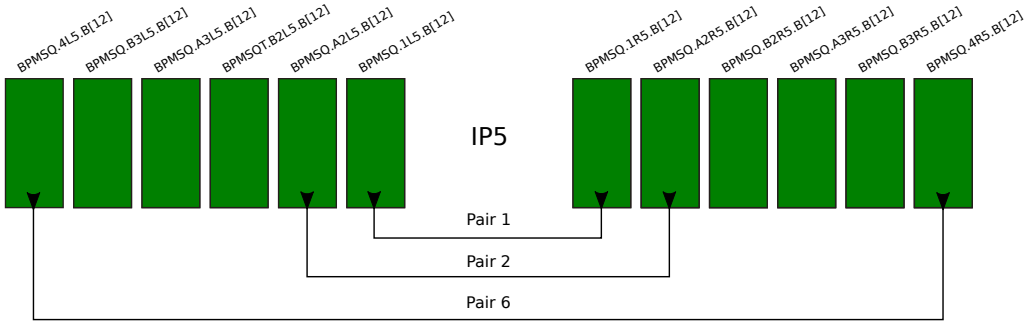


Figure 6.1: Schedule of the BPM pairs in the interaction region around `IP5`.

from the IP, on each side of the IP and for both beams. This means that what is referred to as a BPM pair in this chapter is in fact four different BPMs, one per side of IP and beam.

Under these assumptions, Figure 6.2 shows the RMS beam separation at `IP5` when using different pairs of BPMs and where each point shown corresponds to the correction for the optimal number of singular values. If one uses only one BPM pair at a time, the trend is clear: the closer the BPM pair is to the `IP5`, the more effective it is at correcting the beam

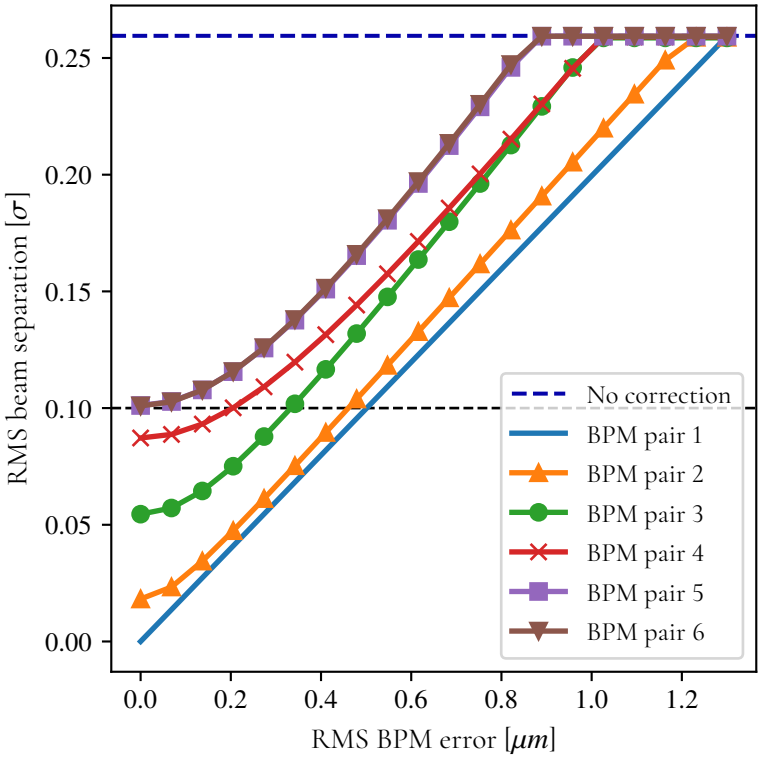


Figure 6.2: The RMS beam separation at `IP5` after local correction as a function of RMS BPM error for $\sigma_{quad} = 0.3\ \mu\text{m}$, plotted for different pairs of BPMs used in the correction.

separation. The underlying reason has to do with the number of error sources between each BPM pair. The closest BPM pair performs the best as there is no error source between the BPMs, and the further away a BPM pair is the more error sources. Because of this relation, the correction of the closest pair is in fact independent of the level of closed orbit pertubation; if the closed orbit is corrected to zero at these BPMs, it directly follows that the closed orbit at `IP5` is zero too. In the same manner, correcting to zero at a BPM pair when there is an error source between them will invariably lead to a residual at the interaction point. This is most clearly seen in Figure 6.2 at the leftmost points where the BPMs are perfect: pair 1 corrects to zero beam separation, every other pair does not with a performance worsening with distance from the interaction point.

Based on the previous reasoning, assuming all BPMs are equal in performance, the closer the pair of BPMs is to the interaction point, the better. There is however an argument for making use of groups of pairs for the local correction. If one has access to several imperfect BPMs close to each other, then it is possible to reduce the level of noise in the readings based on the collective reading being unphysical. A naive schematic of this is shown in Figure 6.3. The figure assumes that any physical orbit must be a straight line for the region. In such a system, the orbit can be estimated based on the line with the least-squares distance to every measurement. Under reasonable assumptions, more measurements imply a better estimate of the line.

This principle is used in Figure 6.4 where groups of BPM pairs are used in the correction. Here it becomes evident that one can improve the result in Figure 6.2 by combining BPMs in

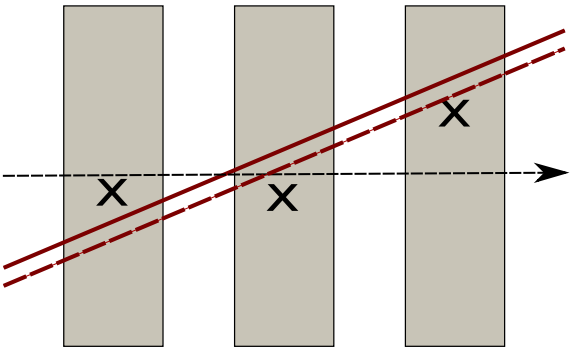


Figure 6.3: Schematic of beam measurement by imperfect BPMs. The gray blocks correspond to BPMs, the full red line corresponds to the actual closed orbit, the black crosses correspond to the BPM reading of the beam and the red dashed line corresponds to the reconstruction of the closed orbit based on the readings.

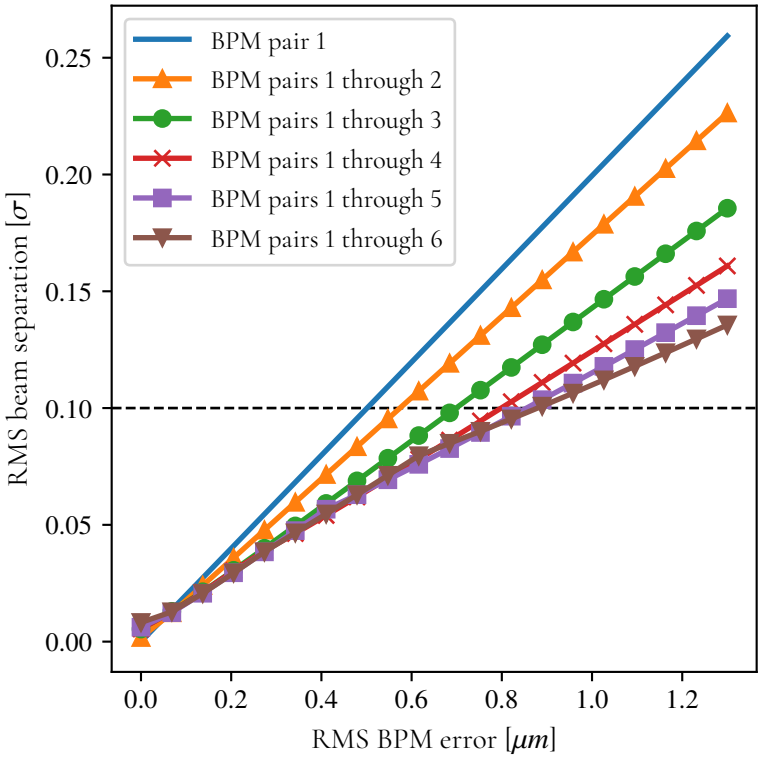


Figure 6.4: The RMS beam separation at `IP5` after local correction as a function of RMS BPM error for $\sigma_{quad} = 0.3\ \mu\text{m}$, plotted for different sets of BPM pairs used in the correction.

the IR for the given quadrupole errors, but not considerably. To get an idea why combining many BPMs does not improve the beam separation by much, consider the following argument. Assume we have n identical BPMs in a sequence of an accelerator and each of them measures the horizontal closed orbit of a beam. In the best-case scenario, we can assume that there are no sources of closed orbit perturbation between each of the BPMs, and so if any of the n BPMs would have provided a perfect reading, then the real closed orbit at every

other BPM would have been computable, i.e. the beam has one degree of freedom, the size of the orbit. The most beneficial scenario for estimating the orbit is if each BPM reading is as important. If the distribution of noise in the BPMs is equal across them, then this would correspond to a beam with constant closed orbit in the region, meaning that the ideal predictor of the closed orbit would be the arithmetic mean. It follows that in the best possible scenario, the best noise reduction we can achieve is by a factor $1/\sqrt{n}$ for a section of n BPMs.

Applied to the setup we are investigating, in the best case scenario, we can assume that groups of BPM pairs function as the closest BPM pair, scaled by the noise reduction of $1/\sqrt{n}$ where n is the number of BPM pairs used. This provides an upper-bound on the allowed BPM noise for satisfactorily correcting the beam separation at `IP5` as per:

$$\text{Required BPM stability } \sigma_{BPM} \text{ when using } n \text{ BPM pairs} \leq \sqrt{n} \times 0.50\,\mu\text{m}\,. \tag{6.1}$$

where the $0.50\,\mu$m is the maximum allowed RMS BPM error for the closest BPM pair. Note that (6.1) is independent of the error in the machine, and optimistic. As there are sources of closed orbit perturbation between the BPM pairs, the reduction factor of the noise will not be $\sqrt{n}$ but smaller, and when combinining BPM pairs it follows that some of the correction will be applied to BPMs other than the closest ones, hence their correction will be worse for perfect BPMs. Nevertheless, (6.1) gives a useful bound on the minimum BPM stability for a given choice of BPM pairs; if the inequality is not satisfied, then there is no possibility that a correction strategy will achieve the requisite beam separation using only information from BPMs.

The results from Figure 6.2 and 6.4 are summarized in Table 6.1. The table provides the

Table 6.1: Table over required BPM stability to achieve a beam separation at `IP5` corresponding to a 0.25% instantaneous luminosity loss for $\sigma_{quad} = 0.3\,\mu$m for different BPMs employed in the correction.

BPM pairs used	Required RMS BPM stability [μm]
1	0.50
2	0.46
3	0.33
4	0.20
5	N/A
6	N/A
1 through 2	0.57
1 through 3	0.70
1 through 4	0.80
1 through 5	0.85
1 through 6	0.88

necessary BPM stability to replace the lumiscans for the assumed error of $\sigma_{quad} = 0.30\,\mu m$, when performed during collision to re-enter head-on collision. This stability needs to be kept over the the full collision, which is in the order of ten hours. If this would be achieved, then local IP correction could be employed after each β^*-levelling step and to correct for orbit drift at the interaction points, i.e. it could replace lumiscans during collision.

A separate use of local IP correction that has not been mentioned is as a means of finding a luminosity signal. If HL–LHC is not operated for an extended period of time, or some equipment fault takes place between two fills, then the closed orbit in the machine might be large enough that collision cannot be found after the ramp. Lumiscans are also performed in these scenarios, but finding collision can take considerably longer than a minute since there is no luminosity signal. Once a luminosity signal is found, head-on collision can be achieved in around a minute. If the stability of the closest BPM pair around interaction points is good enough, it would be possible to correct down to the design orbit at these locations and a luminosity signal would be attained instantly.

A beam separation of $4\,\sigma$ still gives a sufficient luminosity signal to be able to quickly find head-on collision [24]. As correction using the closest BPM pair is independent of the closed orbit in the machine, the result from Table 6.1 can be scaled directly to compute the necessary necessary BPM stability to always find collision:

$$\text{Required BPM stability } \sigma_{BPM} \text{ to find collision} = 40 \times 0.50\,\mu m = 20\,\mu m\,. \tag{6.2}$$

where the factor 40 is the quotient between $4\,\sigma$ and $0.1\,\sigma$. The timescale of this stability dictates how useful the local IP correction can be to find collision. If it is less than a day, then it is not very useful, whereas if if it is in the order of months then one can possibly save time using this approach.

6.2 Conclusion

Based on the results presented, the BPMs closest to the points of collisions would need a stability in the order of a micrometer on a timescale of around ten hours to replace lumiscans as a means of steering beams into head-on collision. The BPM pairs most effective at correcting orbit at the points of collision are the closest ones, with their effectiveness degrading with distance. If one is interested in using local IP correction as a means of finding collision, then the minimum BPM stability is around twenty micrometers for whatever timescale is considered.

As to the larger question of whether it is worthwhile to remove the lumiscans in HL–LHC, the following things ought to be performed:

1. Quantify the actual gain from removing the lumiscans in terms of integrated luminosity.

2. Investigate the feasibility of micrometer stability for the BPMs near the collision points.

3. If it proves feasible to upgrade the BPMs to this precision, quantify the cost of doing so, which can be favorably compared to integrated luminosity.

Finally, the results presented here are compatible with those of a previous study on older HL–LHC optics and under different assumptions [25]. For that study, the necessary RMS BPM stability to replace lumiscans during collision was estimated to be $1.20\,\mu m$.

Chapter 7

Discussion

Having accounted for the results of the thesis, this chapter will be dedicated to discussing choices, assumptions and the outcome of these.

7.1 POCKPy

From the perspective of the task originally set out, the project is a success. Based on the comparisons between the framework and MAD-X, the framework is capable of computing response matrices that match established results. The optimization procedures provided allows for optimizing the error correction and knob implementation, and is able to optimize for both in a few seconds on a Dell XPS 13 with Intel Core i7-8550U CPU @ 1.80GHz. Assuming the full pipeline of running MAD-X to produce output for POCKPy, the full computation should be on the timescale of minutes, which is adequate from the perspective of continuous validation of the HL–LHC optics. This will depend on what is considered sufficient validation of an optics. If analyzing the statistics of the residual and corrector strength employed based on a linear correction is the relevant procedure, then this can be performed in seconds. If instead one wants to make use of the convex optimizer, then one has to sample from many machine errors to estimate these quantities. This takes more time, but doing it naively in a single process still allows hundreds of machines to be corrected in the order of a minute.

7.2 Corrector Budget

The current design of the HL–LHC v.1.5, release candidate 0, does satisfy the corrector budget, in line with previous preliminary verifications [15].

7.2.1 Error Correction

For previous studies of the corrector budget employed in HL–LHC, linear correction strategies based on weighted pseudoinverses have been used. The new routine based on convex optimization allows for a different approach where constraints can be put on individual corrections. Evidently, for appropriately chosen weights (and number of singular values), the two approaches can give similar results. The results of the error correction make a good argument for that the residual closed orbit can be corrected down to less than $1\ \mathrm{mm}$ in HL–LHC, and that this would be afforded by the given corrector budget.

7.2.2 Knob Implementation

The convex optimization used for the knob implementation here managed to find knob implementation within the budget. Finding knob implementation was previously done by manually weighting pseudoinverses which in theory should be able to find the same solution, but is not a robust approach. The new approach introduced in this thesis is faster, can handle constraints and arguably more user friendly.

7.3 Estimating Impact of Orbit Feedback System for Maintaining Collision

The results on LHC from this study are compaible with experience; LHC is a stable machine which in practice could be run during collisions without any OFB, at the expense of a growth in global orbit. The result that the OFB does not have a big impact on the beam separation is a result which is reflected in the fact that lumiscans nevertheless are occasionally performed, which hints at noticeable beam separation during fills. The reason as to why the OFB does so little to control the beam separation is a direct result from using an unweighted pseudoinverse with few singular values. However, as Figure 5.4 demonstrates, opting for using more singular values propagates the error from the BPMs to the orbit. This leads to the conclusion that if the OFB were to be of significant use in minimizing beam separation, then the BPMs employed in the OFB would have to be more stable.

Transferring the OFB as implemented in LHC to HL–LHC leads to analogous results, but scaled differently, as seen in Figure 5.6. The driving factors behind this scaling is the beam size at collision, difference in collision energy and the overall difference in β-function between the two machines. Based on the result, it would appear as if HL–LHC could be run using the same OFB but with more frequent lumiscans (or lumiscan equivalents).

7.3.1 Underlying Assumptions

The more debatable assumptions made in the previous analys should be discussed. First out, that all perturbation are caused by transverse movements of quadrupoles. This is based on the view that the major contributor to closed orbit perturbation is caused by ground motion, i.e. the ground moving somewhat over time, and not uniformly across the $27\ \mathrm{km}$ long accelerator. In the context of closed orbit perturbation during collision, tenths of micrometers across all

quadrupoles, especially the ones in the interaction regions, is enough to induce a global orbit in the order of a hundred micrometers, and beam separation of tenths of σ's at collision points. Other sources of errors are present, e.g. transient currents in magnets and power supply stability, but they are assumed to be of a smaller order. Moreover, as the quadrupole errors give rise to kicks at all quadrupoles, it is assumed that errors not accounted for could be handled by an increase in the magnitude of the assumed quadrupole errors.

Second, that all quadrupoles' transverse movement is i.i.d. in each dimension. This is not an obvious assumption as ground motion can be correlated in space. The major reason for assuming i.i.d. is that alternative formulations would lead to more complex expressions. Moreover, encoding the correlation between different quadrupoles would itself rely on further assumptions which can be questioned, nor is it a given that correlation would worsen the residual. Depending on how the correlation is structured, it is possible that the phase advance between alternatingly focusing and defocusing quadrupoles causes the effect of the orbit kicks at the quadrupoles to cancel each other out more than in the case where they are assumed independent. In short, it is an assumption that makes the problem more tractable and which does not necessarily underestimate the closed orbit perturbation.

Third, that all BPM reading errors are i.i.d. in each dimension. This is an assumption that is motivated by the assumption that BPM reading errors are dominated by electronic errors, and since each BPM has its own (but structurally similar or equal) electronics the errors themselves are independent across different BPMs but distributed similarly. There are exceptions, there was for example an issue with heating in acquisition electronics. Each BPM is connected to a rack of electronics which processes the readings, and one rack can be connected to multiple BPMs. If these racks are not properly cooled and maintained at a stable temperature, then the processed signal gets noisier, and so the noise for some BPM readings was correleated because they shared racks. This was a problem during LHC Run 1, but since then more effective cooling is employed wherefore this effect has less of an impact.

Fourthly, that 5.6 is used to compute the residual at the end of collision. The mathematical motivation as to why is given in Section 5.1.2, which relies on the OFB working based on occasionally sampling the orbit measured at the BPMs and applying a correction directly after. This is a significant simplification as the actual OFB is not blind to the time domain. Nevertheless, the primary implication of the OFB working in the time domain for this study is that each correction gets applied gradually over time. Since the study concerns itself with the beam separation at the end of collision primarily, the time span is in the order of hours, and so the transient effects from previous corrections can be neglected.

Lastly, an implicit assumption, that the ground motion will behave the same in HL–LHC as LHC. HL–LHC will make use of the LHC tunnel and will share much of the structure. Because of this there are good arguments for believing that the ground motion experienced in HL–LHC will be similar to the one experienced in LHC. Still, it is not impossible that the new triplet quadrupole assemblies in HL–LHC might worsen or improve the transfer function from ground motion to magnetic field motion.

7.4 BPM Specification for Local IP Correction

For the study of local IP correction in HL–LHC, the result is comparable to a prior study operating under slightly different assumptions [25]. The point to drive home is that both studies reach the conclusion of needing BPM stability in the order of one micrometer, which is a considerable improvement over what is currently in place (stability in the order of tens of micrometers).

As such, it is important that the feasibility, potential gain in integrated luminosity and the cost of upgrading the BPMs are quantified. Not having to use lumiscans as a means of keeping the beams in head-on collision for stable beams would be possible if the BPMs are sufficiently reliable. Analogously, having the same stability when finding collision between fills or after longer machine stops would save time, and therefore increase the integrated luminosity. This would require stability of the BPMs on the timescale of days, but would also contribute to an effective gain in integrated luminosity if achieved.

7.4.1 Underlying Assumptions

Most of the debatable underlying assumptions for this study were already dealt with in the OFB study. Still, there are two that are worth commenting on. First, the choice of setting the RMS quadrupole transverse misalignment to $0.3\,\mu\text{m}$, the same as that for the OFB study. One argument for choosing this value is that it offers a reasonable base case. It provides a good estimate of the ground motion experienced between two fills, and so the numbers provided for the BPM stability can be interpreted as requirements for finding head-on collision at the end of the ramp. Note however that the performance of the closest BPM pair is independent of the magnitude of the closed orbit perturbation, and so the related numbers, including the theoretical bound on the performance of combining BPMs, are too.

A second assumption made use of in Figures 6.2 and 6.4 was that the optimal number of singular values was available to the beam operator at the point of correction. If the BPMs are relatively unstable then fewer singular values should be used to propagate less BPM error and relatively stable BPMs should be put to use by including more singular values. In practice, the operator cannot know the optimal number of singular values at any given point in time, meaning that a 'fair' number of singular values would instead be chosen based on experience. This does not change the results provided in the study, instead what is here being emphasized is that the local correction strategies employed will at best achieve the beam separation in Figures 6.2 and 6.4 for the given assumptions.

Chapter 8
Conclusion

The thesis work has resulted in a Python framework (POCKPy) for the orbit corrector budget in HL–LHC, and first-order closed orbit correction more generally. Prior to the introduction of POCKPy, verifying the orbit corrector budget had to be performed by direct interaction with MAD-X for the computation of response matrices and subsequent ad-hoc weighted pseudoinverses for error correction and knob implementation. Any notion of optimization was based on manually altering the weights for pseudoinverses in iterations until the optimization goal was achieved. This was time-consuming as well as not robust; no guarantees could be offered for the optimality of a given result nor could the failure to find a solution be provided as proof for the nonexistence of one. In POCKPy, it is sufficient to provide input in the form of `.tfs` files, from which the corrector response matrices are computed. Convex optimization for knob implementation and error correction can be performed with constraints on orbit and corrector strength, and for different goal functions. The final result being that optimal solutions for feasible problems as well as infeasible problems can be identified robustly and efficiently.

When investigating the current corrector budget, POCKPy was able to verify that the orbit corrector budget for HL–LHC version 1.5, release candidate 0, will hold. In addition to verifying the feasibility of the problem, POCKPy was able to optimize the knob implementation such that the orbit at the crab cavities was minimal and the `MCBY` corrector usage was optimized.

Using the response matrices computed in POCKPy, two additional studies were conducted. The first study treated the effect of the orbit feedback system for collision in LHC, and studied an extension of it on HL–LHC. The results indicate that the orbit feedback system as currently used in LHC does not considerably correct for beam separation at collision points, providing an argument for the robustness of the LHC design in terms of the stability at interaction points during collision. Investigating the same orbit feedback applied to HL–LHC led to characteristically similar but differently scaled results as that of LHC, driven in part by the difference in beam sizes for the two machines. The results imply that transferring the orbit feedback system from LHC to HL–LHC, in first approximation, will still provide

a reasonably stable machine, granted that the ground motion sensitivity stays the same, but with a greater need for occasionally bringing beams back into head-on collision via lumiscans.

In the second study, local IP correction as an alternative to lumiscans was investigated for HL–LHC. Based on the results, BPM stability in the order of a micrometer would be necessary for the BPMs next to collision points. The constraints on the BPM stability can be relaxed if multiple BPM pairs around collision points are used, but not by more than a factor two.

In terms of future work for POCKPy, there are still functionalities to implement. From a usability and maintainability perspective, POCKPy is in need of improved error handling and unit testing. These are features planned to be implement as part of the author's remaining internship at CERN. There is also a good argument for including more specialized use cases of POCKPy as dedicated routines.

Lastly, a potentially worthwhile endeavour would be to investigate the possibility of automatizing the search for weights and singular value cutoffs for linear correction strategies. To naively approach this with gradient-based optimization is ruled out as the gradient of the pseudoinverse as a function of the weights and number of singular values is not numerically stable nor continuous. A possibility that could be explored is to use some form of evolutionary algorithm, possibly within the subdomain of differential evolution, with reasonable heuristics hard-coded. It would not be fast as it has to compute many pseudoinverses per generation, but if done correctly it ought to be possible to find linear corrections without having the user manually manipulate weights for individual BPMs and correctors.

Appendices

Appendix A
Element Naming Conventions

As element names occur in this thesis, it is worth spending a little time on the element naming convention used in HL–LHC. Figure A.1 is essentially a zoomed-in and stretched out variant of Figure 1.1 with additional terminology. The key take-aways from Figure A.1 for the present

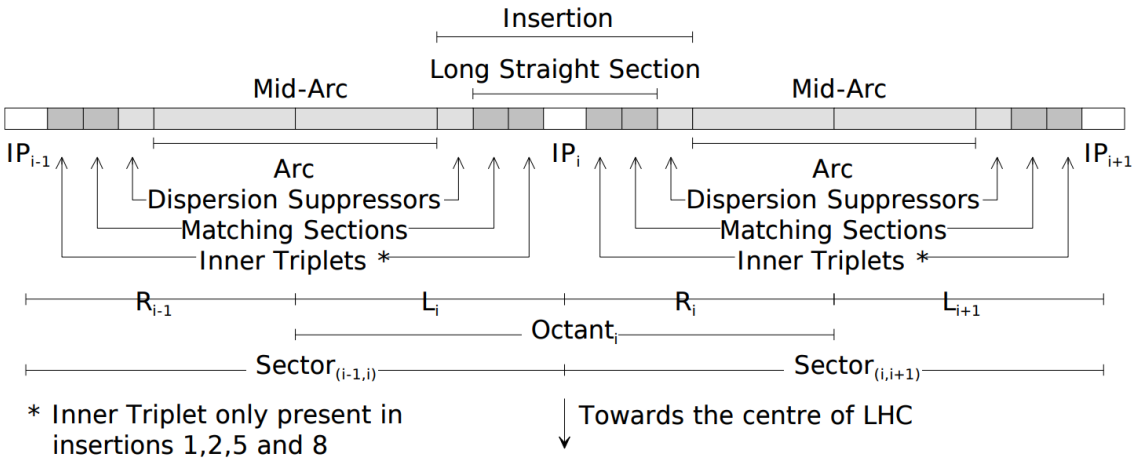


Figure A.1: In-depth view of a segment in LHC [26].

purpose are:

1. Each octant is divided into two *half-arcs* surrounding an *insertion*.
2. Each octant is also divided into a left side and a right side.
3. The center-point of some octants is the *interaction point*, `IP` for short, with their surrounding region referred to as *interaction region*, abbreviated `IR`.

From the perspective of lattice definitions, there are eight `IP`s in HL–LHC, but this is only for notational ease. An interaction point in the strict sense is a point where the two

beams collide, and is only a feature of octant 1, 2, 5 and 8 where experiments are run. When an IP or IR is referred to in this thesis, it is taken for granted that it applies to one of these octants. What all octants nevertheless have in common is that they all have a long straight section in the middle as part of the insertion. The arc can be perceived to be roughly uniform across LHC whereas the long sections differ from octant to octant.

As the base pattern is a FODO lattice (alternatingly focusing and defocusing quadrupoles), the machine can be broken up into half-cells containing one quadrupole each. In doing so, each half-cell is given a number, where the i^{th} quadrupole away from the center of its octant is associated with the i^{th} half-cell. With this in mind, the general naming convention can be summarized as follows:

```
<TYPE><SPECIAL>.<EXTRA><HALF_CELL><LEFT_OR_RIGHT><OCTANT>.B<1_OR_2>
```

- TYPE: Entry specifying the type of element. See Table A.1 for example entries.
- SPECIAL: Optional entry which can be used to specify sub-type of element, e.g. H or V to signify if a corrector is acting on the horizontal or vertical plane.
- EXTRA: Optional entry used to separate between otherwise identically named elements. E.g. A, B, C to separate between three bending magnets in the same half-cell.
- LEFT_OR_RIGHT: Entry specifying which side of the closest IP the element is on. Assumes either L or R.
- OCTANT: Entry specifying the octant the element is a part of. Valid entries are 1 to 8.
- 1_OR_2: Entry specifying which beam the element is part of. Either 1 or 2, unless the element is shared between the two beams in which case the element name ends with the OCTANT entry.

Table A.1: Table over prefixes for different element types.

Element type	Prefix
Bending magnet	MB
Quadrupole	MQ
Orbit corrector	MCB
BPM	BPM
Crab cavity	ACFCA
Drift	DRIFT

For example, the element MQ.25L5.B1 is a quadrupole on the left side of IP5, in the 25^{th} half-cell and for Beam 1. The special identifier can be used in multiple ways, for example MQML.10R1.B1 is a different type of quadrupole in half-cell 10, on the right side of IP1 for beam 1. Here the special identifier describes the type of quadrupole. For MCBH.21R5.B1, the special identifier H signifies that it is a horizontal orbit corrector. Two other horizontal orbit correctors are MCBYH.A4R5.B1 and MCBYH.B4R5.B1. In this case the two horizontal

orbit correctors share type MCBY, octant, side of IP, half-cell and beam, which is why they make use of the extra specifiers A and B to tell them apart.

Note that there are elements which skip the appendage of .B<1_OR_2>. These correspond to elements which are common to both beams, which can only happen in the IR. This is due to the fact that when two beams are brought to collision they pass through the same equipment close to the point of collision. Examples of this are MCBXFAV.3L5, a shared vertical corrector, and MBXF.4L5, a shared bending magnet. This has an important impact on the analysis, since powering a shared corrector impacts both beams, same thing goes for imperfections in shared bending magnet.

Appendix B
Crab Cavities

To maximize the *luminosity* at interaction points, HL–LHC will make use of elements called *crab cavities*. In Figure B.1 the effect of crab cavities is displayed where bunches of particle are colliding at an interaction point. Due to longitudinal dynamics (not covered in this

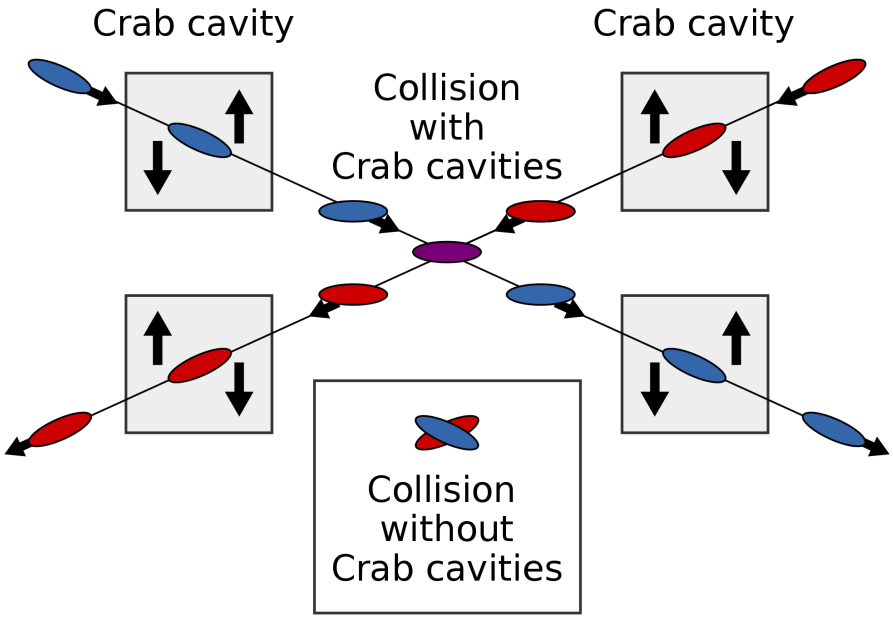


Figure B.1: Illustrative figure on the effect of Crab Cavities [27].

thesis), beams are made up of bunches with a proper longitudinal length. Moreover, by geometry, two beams colliding at a fixed point has to do so at an angle unless they are to have a continuum of collision points. If the crossing angle is too small, then the two beams may still electromagnetically interact with each other considerably before colliding, which is undesirable and enforces a noticeable crossing angle. Combining these two facts leads to the

situation depicted: the bunches making up the two beams would not overlap perfectly in the colllision, unless the bunches were somehow deformed before entering collision.

Crab cavities are used to deform bunches such that their overlap at collision is maximized. The crab cavities are one of the major contributors toward increasing the luminosity in HL–LHC, and so maintaining good conditions for their operations is important. Critically, any transverse orbit at the crab cavities will generate undesired beam-induced RF power in the crab cavity itself, which needs to be evacuated. Beam orbit at crab cavities therefore has to be minimized.

Nomenclature

α, β, γ	Twiss parameters.
β^*	Beta function at collision point.
†	Matrix and vector operator for taking subsets of rows and columns for matrices, or subsets of rows for vectors.
$\Delta\psi$	Rotational misalignments w.r.t. longitudinal axis.
Δk_0	Dipole field error.
Δk_1	Quadrupole field error.
Δx, Δy	Transverse misalignments.
ϵ	Emittance.
ϵ_N	Normalized emittance.
Λ	Diagonal matrix.
$\mathcal{L}$	Luminosity.
$\mathbf{A}$	Linear correction strategy mapping from closed orbit to corrector strengths.
$\mathbf{RM}_\mathrm{c}$	Orbit response matrix for correctors.
$\mathbf{RM}_\mathrm{e}$	Orbit response matrix for machine errors.
$\mathbf{abs}(\cdot)$	Absolute value operator. When applied to a vector quantity the output is the result from applying the absolute value operator to each entry.
$\mathbf{pinv}(\cdot)$	Operation for the Moore-Penrose inverse.
Σ	Covariance matrix.
θ	Orbit kick.

b Vector of inequality constraints for the absolute corrector strength used.

c Vector of corrector strengths.

e Vector of machine errors.

F The force vector.

k Vector of a subset of machine positions and dynamical variables where values indicate equality constraints for a knob.

o Vector representing a closed orbit of arbitrary origin.

w Vector of zero-mean noise added to BPM reading.

y Vector of BPM reading measuring positions of both beams, i.e. positions at BPMs with added noise.

c Speed of light in vacuum.

d Beam separation.

k_0 Normalized field strength of a dipole.

k_1 Normalized field strength of a quadruple.

L Length of an element.

Glossary

beam separation Difference in closed orbit between two colliding beams at a given point.

bending magnet Magnetic dipoles bending the beam reference trajectory.

BPM Beam Position Monitor. Sensors employed in accelerators to measure the transversal position of beams at a given position along the accelerator path.

bunch Group of particles in a beam with a given longitudinal length. A beam typically consists of thousands of bunches.

closed orbit Unique orbit which closes in on itself after one turn in a circular accelerator.

collision point Point where the two beams collide. For LHC and HL–LHC they are located in the middle of octants 1, 2, 5, and 8.

corrector strength Magnitude of the orbit kick for a given orbit corrector. In this thesis measured in Tm, but otherwise often expressed in rad.

crossing angle Angle under which two beams collide.

drift space Straight section in the accelerator with (virtually) no electromagnetic fields.

error correction Powering orbit correctors to cancel the closed orbit perturbation caused by machine errors.

event pile-up Luminosity per bunch crossing.

fill Process in which bunches of protons are injected into LHC, ramped up to collision energy and then collided at the collision points, and finally dumped.

FODO cell A cell in a FODO lattice consisting of a focusing and defocusing quadrupole separated by driftspaces.

FODO lattice A periodic lattice where the smallest periodic cell is a FODO cell.

half-cell Half of one FODO cell containing one of the two quadrupoles.

HL–LHC High Luminosity Large Hadron Collider. An upgrade of LHC to increase the instantaneous luminosity by a factor of five, extending the operation of the collider beyond 2025.

IP Interaction Point. Synonym to collision point..

knob A routine used to implement a specific closed orbit in the machine by powering orbit correctors, often tied to a specific operation scenario.

lattice Schematic of accelerator where all element positions are given.

LHC Large Hadron Collider. World's biggest particle collider, operational at CERN since 2008.

LHC Run Period of time during which LHC is operational. In-between LHC Runs, LHC can be upgraded and maintained.

LHCB1/LHCB2 LHC Beam 1/2. Internal name in MAD-X for Beam 1 (clockwise) and Beam 2 (counter-clockwise).

lumiscan Knob used in LHC for bringing beams into head-on collision. Currently used to enter collision and to occassionally reduce the beam separation during collision.

machine Used interchangeably with accelerator with emphasis on a machine being an accelerator with a fix set of machine errors..

machine error Error in alignment or field of bending magnet or quadrupole.

MAD-X Accelerator design software employed at CERN for simulations.

OFB Orbit Feedback System. Feedback system currently employed in LHC for keeping the beam stable and minimizing orbit throughout the different stages of a fill.

optics A lattice together with specification of magnetic fields. Also used to indicate the β-functions in a machine.

orbit corrector Dipole magnet with adjustable strength such to induces an orbit kick at its position, allowing the operator to change the closed orbit in.

orbit corrector budget Specification of the maximum corrector strength allowed in the orbit corrector of an accelerator.

perfect machine Machine with no machine errors.

Q*i* Quadrupole inside the i^{th} half-cell.

RMS Root Mean Square. For zero-mean distributions, RMS is equal to the standard-deviation.

singular value cutoff Specification of the number of singular values to be used when performing an SVD-based pseudoinverse.

SVD Singular Value Decomposition.

turn Synonym for a full lap around a circular accelerator.

Bibliography

[1] CERN Home: LHC. http://lhc-machine-outreach.web.cern.ch/lhc-machine-outreach/. [Online, accessed 5-August-2019].

[2] CERN. CERN Yellow Reports: Monographs, Vol 4 (2017): High-Luminosity Large Hadron Collider (HL-LHC) Technical Design Report V. 0.1. *CERN*, Sep 2017.

[3] D. Gamba and R. De Maria. IP Orbit Correction Update for HL-LHC. In *Proc. 9th International Particle Accelerator Conference (IPAC'18), Vancouver, BC, Canada, 29 April-04 May 2018*, number 9 in International Particle Accelerator Conference, pages 3048–3051, Geneva, Switzerland, Jun. 2018. JACoW Publishing. https://doi.org/10.18429/JACoW-IPAC2018-THPAF039.

[4] G. Roy L. Deniau, H. Grote and F. Schmidt. MAD-X: User's Reference Manual Version 5.05.02. http://madx.web.cern.ch/madx/webguide/manual.html, 2019. [Online, accessed 5-August-2019].

[5] Andrzej Wolski. *Beam Dynamics in High Energy Particle Accelerators*. IMPERIAL COLLEGE PRESS, April 2013.

[6] E.D. Courant and H.S. Snyder. Theory of the alternating-gradient synchrotron. *Annals of Physics*, 281(1-2):360–408, April 2000.

[7] Werner Herr and B Muratori. Concept of luminosity. 2006.

[8] Vladimir Rakočević. On continuity of the moore-penrose and drazin inverses. *Matematichki Vesnik*, 49(3-4):163–172, 1997.

[9] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

[10] Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

[11] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[12] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[13] Michael Grant, Stephen Boyd, and Yinyu Ye. *Global Optimization: From Theory to Implementation*, pages 155–210. January 2006.

[14] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference (ECC)*, pages 3071–3076, 2013.

[15] Riccardo De Maria et al. Optics v1.5 Update. `https://indico.cern.ch/event/836669/contributions/3507590/`, July 2019. [Online, accessed 14-October-2019].

[16] S C Kleene. Representation of events in nerve nets and finite automata. In *In Automata studies, Annals of mathematics studies*, pages 3–41. Princeton University Press, 1956.

[17] Jan Goyvaerts. Regular Expression Tutorial – Learn How to Use Regular Expressions. `https://regular-expressions.info`. [Online, accessed 17-October-2019].

[18] Ralph Steinhagen, J. Andersson, L.K. Jensen, R. Jones, and J. Wenninger. LHC Orbit Stabilisation Tests at the SPS. pages 886 – 888, June 2005.

[19] A Boccardi, A Butterworth, Eva Calvo, R Denz, Marek Gasior, Jose Gonzalez, Stephen Jackson, LK Jensen, OR Jones, Quentin King, G Kruk, Mario Lamont, S Page, Ralph Steinhagen, and J. Wenninger. Commissioning and Initial Performance of the LHC Beam-Based Feedback Systems. May 2010.

[20] Riccardo De Maria et al. Can we simplify HL-LHC circuits? `https://indico.cern.ch/event/580313/contributions/2359640/attachments/1401962/2140218/Chamonix2017_7.pdf`, January 2017. [Online, accessed 10-October-2019].

[21] Jörg Wenninger. Operation and Configuration of the LHC in Run 2. pages 24–25, Mar 2019.

[22] Arkadiusz Gorzawski and Jörg Wenninger. Orbit Drifts at the LHC Interaction Points During the Squeeze. May 2015.

[23] B J Holzer, R De Maria, S Fartoukh, A Chancé, B Dalena, J Payet, A Bogomyagkov, R B Appleby, S Kelly, M B Thomas, L Thompson, M Korostelev, K M Hock, A Wolski, C Milardi, A Faus-Golfe, and J Resta Lopez. Optics Design and Lattice Optimisation for the HL-LHC. (CERN-ACC-2013-0134), May 2013.

[24] Jörg Wenninger. private communication, July 2019.

[25] Miriam Fitterer and R De Maria. BPM Tolerances for HL-LHC Orbit Correction in the Inner Triplet Area. (CERN-ACC-2015-0176), May 2015.

[26] R. Saban. Equipment Naming Conventions in LHC. `http://lhc-proj-qawg.web.cern.ch/lhc-proj-qawg/CD-ROM/Quality/QA204.pdf`, 1999. [Online, accessed 5-August-2019].

[27] Crab Cavity Figure. https://commons.wikimedia.org/wiki/File:Crabcavity_n2.svg. [Online, accessed 5-August-2019].

Master's Theses in Mathematical Sciences 2019:E61
ISSN 1404-6342

LUTFMA-3393-2019

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/