



UPPSALA
UNIVERSITET



Optimising the present and designing the future: a novel SPS injection system

Thesis in partial fulfillment of the requirements for the degree of Master in
Science with a Major in Physics

Student:
Elias Waagaard

Supervisors:
Francesco Velotti,
Volker Ziemann

Subject reader:
Maja Olvegård



June 15, 2022

Abstract

The Super Proton Synchrotron (SPS) injection system plays a fundamental role to preserve the quality of injected high-brightness beams for the Large Hadron Collider (LHC) physics program and to maintain the maximum storable intensity. The present system is the result of years of upgrades and patches of a system not conceived for such intensities and beam qualities. In this study, we first investigate the effect of emittance growth due to amplitude-dependent tune shifts for erroneously injected beams. As a next step, we propose the design of a completely new injection system for the SPS using multi-level numerical optimisation, including realistic hardware assumptions. Methods and pseudo-algorithms of how this hierarchical optimisation framework can be adapted to other situations for optimal accelerator system design are shown. In addition, we explore the benefits of a numerical optimisation framework for the current SPS injection kicker timing system to minimise residual injection oscillations for maximised delivered beam intensity. We also demonstrate how a simple neural network based upon recorded data can approximate the injection system as a surrogate model, allowing for further studies of different optimisation algorithms even without beam time.

Popular science summary

The Super Proton Synchrotron (SPS) is a particle accelerator and the last link in the injector chain leading into the Large Hadron Collider (LHC), the largest particle accelerator in the world, located at the CERN accelerator complex outside Geneva. The injection system of the SPS plays a fundamental role to preserve the quality of the injected beams for the LHC physics programs, acting as a bottleneck for the number of particle collisions delivered to the different experiments. The present injection system is the result of years of upgrades and patches, where the magnets have not always been configured in the ideal way to handle beams of high intensity. In this study, we explore how software can be implemented to design a new configuration of injection magnets to deliver a stable beam at the lowest resource cost possible. We use optimisation methods that span over many levels, and mimic the process of natural selection to find the ideal injection design that minimises the magnet resources needed, gives more space to the beam and makes the magnet design more uniform.

As a second step, we study the present injection system and reap the benefits of modifying the parameters without changing the hardware, in order to reduce injection errors that cause the beam to oscillate and potentially get lost. We first perform theoretical calculations of the injection errors, and simulate how these errors would alter the final beam quality. We then implement a numerical optimisation framework that finds the ideal timing on the nanosecond-level for the injection magnets in the SPS. This optimiser finds timing configurations that reduce these oscillations by up to 70%. As a last step, we use the collected data to develop a neural network model - an algorithm mimicking the human brain - to imitate the accelerator in order to compare different optimisation methods even when beam time is not available.

Populärvetenskaplig sammanfattning

Super Proton Synchrotron (SPS) är en partikelaccelerator och den sista länken i injektorkedjan som leder in i Large Hadron Collider (LHC), den största partikelacceleratorn i världen, belägen vid CERN-komplexet utanför Genève. Injektionssystemet för SPS spelar en grundläggande roll för att bevara kvaliteten på de injicerade strålarna för LHC:s fysikprogram, och fungerar som en flaskhals för antalet partikelkollisioner som levereras till de olika experimenten. Det nuvarande injektionssystemet är resultatet av år av uppgraderingar och lagningar, vars magneter inte alltid har konfigurerats på rätt sätt för att hantera strålar med hög intensitet. I denna studie utforskar vi hur mjukvara kan implementeras för att designa en ny konfiguration av injektionsmagneter för att leverera en stabil stråle till lägsta möjliga kostnad. Vi använder optimeringsmetoder som sträcker sig över många nivåer, och imiterar det naturliga urvalet för att hitta den idealiska injektionsdesignen som minimerar magnetresurserna, ger mer utrymme åt strålen och gör magnetdesignen mer enhetlig.

I nästa steg studerar vi det nuvarande injektionssystemet och modifierar parametrarna utan att ändra hårdvara, för att minska injektionsfel som gör att strålen oscillerar och potentiellt försvinner. Vi utför först teoretiska beräkningar av injektionsfelen och simulerar hur dessa fel skulle förändra den slutliga strålkvaliteten. Vi implementerar sedan ett numeriskt optimeringssystem som hittar den idealiska timingen på nanosekundnivå för injektionsmagneterna i SPS. Detta optimeringsprogram hittar magnettimningar som minskar dessa oscillationer med upp till 70%. Som ett sista steg använder vi insamlad data för att utveckla en neural nätverksmodell - en serie algoritmer som efterliknar den mänskliga hjärnan - för att imitera acceleratoren så att vi kan jämföra olika optimeringsmetoder även när acceleratortiden inte är tillgänglig.

Contents

Abstract	i
Popular Science Summary	i
Populärvetenskaplig sammanfattning	ii
Table of contents	iii
1 Introduction	1
2 Theoretical background	2
2.1 Single-particle dynamics	2
2.2 Linear beam dynamics	5
2.3 Off-momentum effects	8
2.4 Linear imperfections from dipole errors	10
3 Beam transfer physics	12
3.1 Ring and transfer line beam dynamics	12
3.2 Injection and extraction	13
4 Emittance growth after injection	15
5 The Super Proton Synchrotron	19
5.1 SPS physics users	20
5.2 SPS injection sequence	21
5.2.1 MSI and MKP injection elements	22
5.3 The PS-SPS Transfer Line	24
5.4 SPS injection error tolerances	26
6 Injection system design with hierarchical optimisation	26
7 SPS injection kicker delay optimisation	31
7.1 Offline analysis and optimisation simulations	31
7.2 Online kicker delay optimisation	33
7.3 Neural network model of the SPS injection	35
7.4 Optimisation algorithm selection using the surrogate model	41

8	Conclusions	44
9	Acknowledgements	45
A	Appendix	49

1 Introduction

The Super Proton Synchrotron (SPS) is the last injector of the Large Hadron Collider (LHC), hence the second-largest accelerator of the CERN accelerator complex [1]. The SPS accelerates protons from 26 GeV up to 450 GeV. One of the main duties of the SPS is to provide high-brightness beams of high quality for the LHC physics program, as well as to the fixed-target experiments of the North experimental Area (NA). The SPS injection system plays a fundamental role to the SPS beam quality, not only to preserve the quality of the injected beam, but also to contribute to the maximum storable intensity in the ring. Any injection errors of the beam might imply severe consequences of the delivered beam quality. In Section 4, we investigate the effect of emittance growth of kicked and mismatched beams due to amplitude-dependent tune shift for arbitrary beams, published in [2]. We also present the error tolerances for the SPS injection system due to these non-linear effects in Section 5.

The present injection system is the result of years of upgrades and ad-hoc modifications, undergoing many modifications in its lifetime to adapt to the ongoing evolution of performance requirements. In particular, the upcoming High-Luminosity LHC upgrade further stresses the need for a new robust long-term injection solution [3]. Numerical optimisation and machine learning have gained attention in accelerator physics, for instance from efforts to tune parameters to improve beam quality at the AWAKE experiment [4, 5]. Some attempts have also been made to create genetic algorithms for multi-objective optimisation to construct transfer lines [6], but a final solution has not yet been proposed. In Section 6, based upon the results from our previous study [7]¹, we present a generic hierarchical optimisation framework that gradually constructs a new injection sequence and generates ideal candidate solutions for all SPS optics, while considering realistic hardware and targeting the best trade-off in hardware complexity and performance.

After the CERN Long Shutdown 2 (LS2), significant modifications of the injection were undertaken to allow the higher brightness required from the LHC Injectors Upgrade (LIU) upgrade [8]. In this context, a completely new injection interlock system was put in place, together with new instrumentation and layout changes. In Section 7, we investigate the injection beam stability, beam quality and kicker optimisation for the SPS along with its operational performance. In the light of the imminent LHC run 3, we explore the benefits of a numerical optimisation framework for the SPS injection kicker timing system to minimise residual injection oscillations to maximise delivered beam intensity, using an optimisation environment. In other words, these numerical optimisers constitute “low-hanging fruit” that can be rapidly implemented by accelerator operators to stabilise the injected beam quality, without any hardware modifications. In Section 7.3, we also present how a simple neural network can approximate the injection system behaviour as a surrogate model to fully test different optimisation algorithms also offline, when beam time is not available.

¹Section 2, Section 3, Section 5 and parts of Section 6 of this thesis appear in the official mid-project report [7].

2 Theoretical background

2.1 Single-particle dynamics

The force experienced by charged particles in an electromagnetic field can be described by the Lorentz force

$$\vec{F} = \frac{d\vec{p}}{dt} = q(\vec{E} + \vec{v} \times \vec{B}), \quad (1)$$

where q is the charge of the particle, $\vec{E}$ is the electric field vector, $\vec{v}$ is the velocity vector and $\vec{B}$ is the magnetic field vector. The electric and magnetic fields can be used to accelerate and guide the particles. The *beam dynamics* describes the evolution of the particle trajectory due to this Lorentz force. In order to steer the beam of charged particles, magnets are often used. Multipolar magnets constitute important building blocks for accelerator components to control the motion of charged particles. Depending on their order, magnets can be used for different purposes, for example to bend the trajectories of particles and to focus beams of charged particles. For an n^{th} -order multipole, the magnetic field in the transverse plane is expressed as

$$B_y(x, y) + iB_x(x, y) = [B_n(s) + iA_n(s)](x + iy)^{n-1}, \quad (2)$$

where $B_n(s)$ and $A_n(s)$ are the normal and skew relative coefficients

$$\begin{aligned} B_n(s) &= \frac{1}{(n-1)!} \left. \frac{\partial^{n-1} B_y}{\partial x^{n-1}} \right|_{(0,0;s)} \\ A_n(s) &= \frac{1}{(n-1)!} \left. \frac{\partial^{n-1} B_x}{\partial x^{n-1}} \right|_{(0,0;s)} \end{aligned} \quad (3)$$

that depend on the longitudinal coordinate s [9]. A dipole corresponds to $n = 1$, a quadrupole to $n = 2$ and a sextupole to $n = 3$, and so on. Magnets of order $n = 1$ and $n = 2$ can be described with matrices, whereas magnets of higher order need more complex maps to describe their action on the beam of particles. A skew n^{th} -order multipole is simply a normal n^{th} -order multipole rotated by $90^\circ/n$ in the transverse plane. A schematic representation of a dipole, a quadrupole and a sextupole is displayed in Figure 1.

A reference orbit is often used to describe the ideal motion of charged particles in circular accelerators. The coordinate system employed for a particle in a circular accelerator is a moving coordinate system where s is the curvilinear coordinate, x, y are horizontal/vertical positions, and ρ is the local radius of curvature. An example of such a coordinate system is displayed in Figure 2. For charged particles of high energy, the speed is close to that of the speed of light, with path length $s = ct$. We denote the transverse phase space as (x, x', y, y') , where prime indicates a derivative over path length. On the other hand, the trajectory in which a particle will return to its initial location with the same accelerator conditions is denoted *closed orbit* [9].

To describe the dynamics of a system such as a circular collider, transfer maps provide a useful tool as a set of functions to give a final set of phase space coordinates from an initial set of phase space coordinates. Linear transfer maps can be described by matrices, where the most simple linear element is a drift space of length L

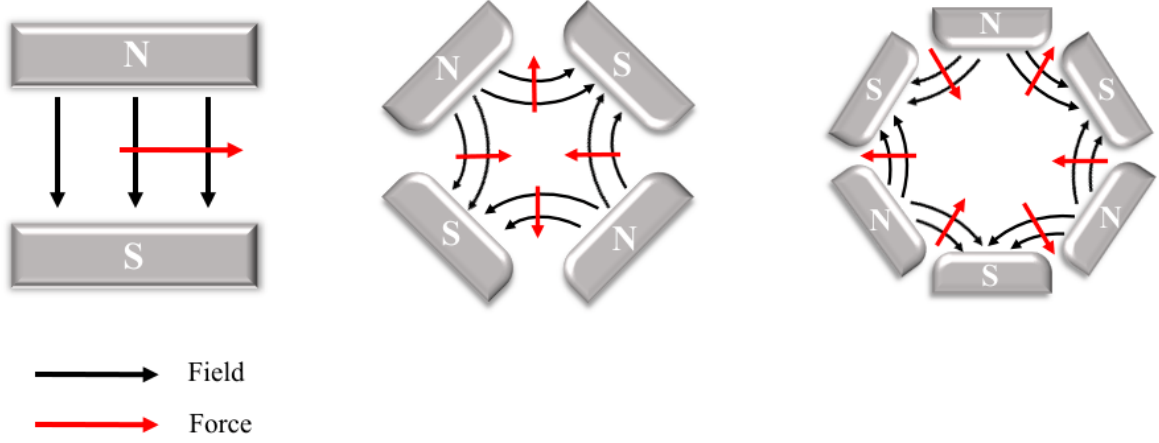


Figure 1: Schematic drawing of field lines and force on a positively charged particle moving out of the paper for a dipole (left), quadrupole (middle) and sextupole (left). Dipoles and quadrupoles are both linear elements, whereas the sextupole is the non-linear element of lowest order.

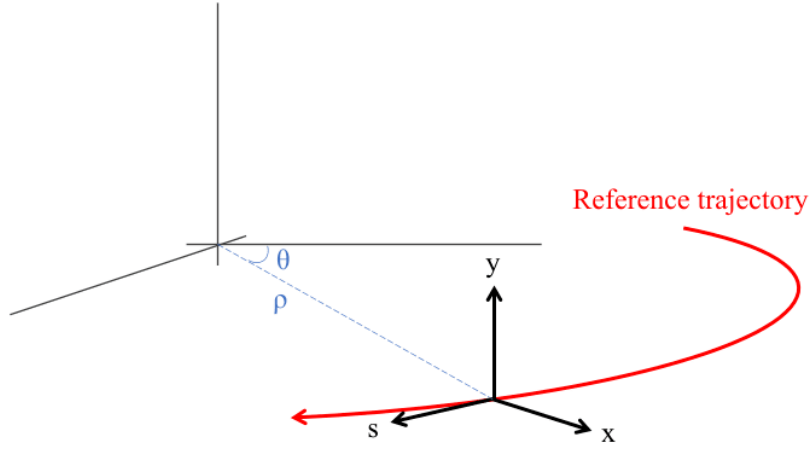


Figure 2: Reference system for the particle propagating in the accelerator.

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{s+L} = \begin{pmatrix} 1 & L \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ x' \end{pmatrix}_s. \quad (4)$$

A sector dipole, as seen in Figure 1, of length L can be described by

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{s+L} = \begin{pmatrix} \cos(L/\rho) & \rho \sin(L/\rho) \\ (1/\rho) \sin(L/\rho) & \cos(L/\rho) \end{pmatrix} \begin{pmatrix} x \\ x' \end{pmatrix}_s, \quad (5)$$

where ρ is the local curvature of the reference orbit. The linear transfer maps described by matrices are taken from [10]. For the quadrupoles, or magnet of order $n = 2$, the gradient of the magnetic field is needed. The normalised gradient k of quadrupole that generates a linear

field that vanishes at the reference orbit is defined as

$$k = \frac{1}{B_0 \rho} \frac{\partial B_y}{\partial x}, \quad (6)$$

where the magnetic rigidity $B_0 \rho$ is defined as

$$B_0 \rho = \frac{p_0}{q}, \quad (7)$$

with q being the charge of the particles and p_0 the design momentum. A focusing quadrupole ($k > 0$) of length L is described by

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{s+L} = \begin{pmatrix} \cos(L\sqrt{|k|}) & (1/\sqrt{k}) \sin(L\sqrt{|k|}) \\ -\sqrt{k} \sin(L\sqrt{|k|}) & \cos(L\sqrt{|k|}) \end{pmatrix} \begin{pmatrix} x \\ x' \end{pmatrix}_s, \quad (8)$$

whereas a defocusing quadrupole ($k < 0$) is described by

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{s+L} = \begin{pmatrix} \cosh(L\sqrt{|k|}) & (1/\sqrt{k}) \sinh(L\sqrt{|k|}) \\ \sqrt{k} \sinh(L\sqrt{|k|}) & \cosh(L\sqrt{|k|}) \end{pmatrix} \begin{pmatrix} x \\ x' \end{pmatrix}_s. \quad (9)$$

In addition, some linear elements can be simplified in the case of $L \rightarrow 0$ and $kL \rightarrow \text{constant}$, also called the *thin-lens approximation* - used when the focal length is much longer than the magnet length. The resulting transfer map for a focusing quadrupole in this limit becomes

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{s+L} = \begin{pmatrix} 1 & 0 \\ -kL & 1 \end{pmatrix} \begin{pmatrix} x \\ x' \end{pmatrix}_s. \quad (10)$$

A combination of focusing and defocusing sections is often referred to as a FODO lattice. A FODO cell contains one focusing quadrupole (F), a drift space of length L (O), a defocusing quadrupole (D) and another drift space of length L . A lattice built of FODO cells will make the particle motion behave as a harmonic oscillator with a constant restoring effect, since any diverging particle from the reference orbit will be focused back. An example illustration of horizontal particle motion in a FODO lattice is shown in Figure 3.

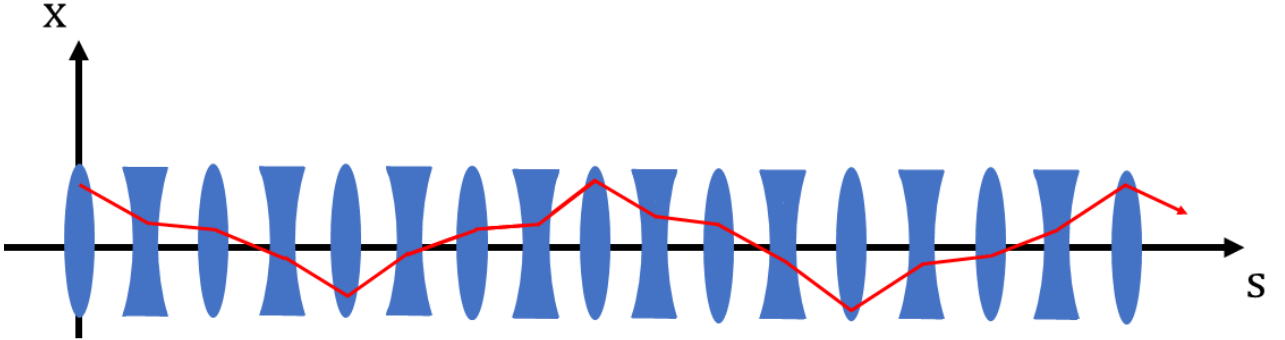


Figure 3: Example of particle motion in a FODO lattice.

These transfer matrices can be combined to a single transfer matrix of all elements along the beamline (*one-turn map* for a circular collider and rings), propagating the particle through all the elements from its initial location s . Using the corresponding matrix M_i of each linear element:

$$M(s_1) = M_N \dots M_2 M_1 \quad (11)$$

If we instead want to relate the one-turn matrices M at different locations s_1 and s_2 , we resort to the similarity transform

$$M(s_2) = M_1 M(s_1) M_1^{-1}. \quad (12)$$

As we desire the motion of the particles to be stable over a large number of turns, $M(s)$ has to be related to a pure rotation matrix via a similarity transform [11].

2.2 Linear beam dynamics

For the equations of motion in the case of linear fields, we consider s -dependent fields from dipoles and quadrupoles. We can write the total momentum as $p = p_0(1 + \delta)$, where $\delta = (p - p_0)/p_0$ represents off-momentum particles. The linearised equations of the *betatron motion* (assuming only horizontally bending dipoles) are

$$\frac{d^2 x}{ds^2} + K_x(s)x = \frac{1}{\rho(s)}\delta \quad (13)$$

$$\frac{d^2 y}{ds^2} + K_y(s)y = 0 \quad (14)$$

with restoring coefficient $K = K(s)$ being a function of the longitudinal position s [10]. Evaluated at the closed orbit, we have

$$\begin{aligned} K_x(s) &= 1/\rho^2 - \frac{\partial B_y}{\partial x} \frac{1}{B\rho} \\ K_y(s) &= \frac{\partial B_y}{\partial x} \frac{1}{B\rho} \end{aligned} \quad (15)$$

where ρ is the local bending radius of the accelerator. Thus, we note that Equations (13) and (14) are inhomogeneous linear differential equations [12]. In the linear case, we consider particles with design momentum $\delta = 0$. These transverse oscillations are called betatron oscillations, described by *Hill's equations of linear transverse particle motion*

$$\begin{aligned} \frac{d^2 x}{ds^2} + K_x(s)x &= 0 \\ \frac{d^2 y}{ds^2} + K_y(s)y &= 0 \end{aligned} \quad (16)$$

The restoring functions are periodic with $K_{x,y}(s) = K_{x,y}(s + L)$, where L is the length of the periodic structure in the accelerator, such as a FODO cell illustrated in Figure 3. The solution to Hill's equations in the x -plane is finally given by, for $K > 0$

$$x = \sqrt{\varepsilon\beta} \cos \phi \quad (17)$$

$$x' = -\alpha\sqrt{\varepsilon/\beta} \cos \phi - \sqrt{\varepsilon/\beta} \sin \phi, \quad (18)$$

where $\beta = \beta(s)$ is the amplitude modulation due to the changing focusing strength, and $\phi = \phi(s) - \phi_0$, which also depends on the focusing strength. Equation (17) and (18) follow from the so-called Floquet's theorem for periodic differential equations [10]. ε and ϕ_0 are constants that depend on initial conditions. We also define $\alpha = -\beta'/2$ and $\gamma = \frac{1+\alpha^2}{\beta}$. If we plot these stable oscillations in phase space through a FODO lattice over many turns at a given location in the accelerator, an ellipse emerges. This ellipse, which is a conic section, is described by the equation

$$2J_x = \gamma x^2 + 2\alpha x x' + \beta x'^2, \quad (19)$$

where $2J_x$ describes its magnitude. In the horizontal plane, J_x defines the shape of the ellipse for single-particle dynamics. If we consider multiple particles over many turns, we are also interested in the *emittance* ε , which is the beam property illustrating how much area the beam occupies in phase space. The area of the ellipse, $A = \pi\varepsilon$, remains constant in a FODO lattice if only conservative forces act on the beam. The ellipse described by Equation (19) is illustrated in Figure 4. The transverse emittance ε , with unit $[\text{m} \cdot \text{rad}]$, is determined from initial conditions. The transformation that transforms the ellipse into a circle is given by

$$\begin{pmatrix} x \\ x' \end{pmatrix} = \begin{pmatrix} \sqrt{\beta} & 0 \\ -\alpha/\sqrt{\beta} & 1/\sqrt{\beta} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad (20)$$

fully derived in [13]. The parameters (x_1, x_2) describe the normalised phase space. The parameters α , β and γ are known as *Twiss parameters*, and J_x is the action, also known as the *Courant-Snyder invariant*. The fact that we observe an underlying oscillation leads us to decompose the full-turn transfer matrix $\hat{R}$ into a matrix $\mathcal{A}$ that twists and stretches the phase space coordinate axes together with a rotation matrix $\mathcal{O}$ [13]

$$\begin{aligned} \hat{R} &= \mathcal{A}^{-1} \mathcal{O} \mathcal{A} = \begin{pmatrix} \sqrt{\beta} & 0 \\ -\alpha/\sqrt{\beta} & 1/\sqrt{\beta} \end{pmatrix} \begin{pmatrix} \cos \mu & \sin \mu \\ -\sin \mu & \cos \mu \end{pmatrix} \begin{pmatrix} 1/\sqrt{\beta} & 0 \\ \alpha/\sqrt{\beta} & \sqrt{\beta} \end{pmatrix} \\ &= \begin{pmatrix} \cos \mu + \alpha \sin \mu & \beta \sin \mu \\ -\frac{1+\alpha^2}{\beta} \sin \mu & \cos \mu - \alpha \sin \mu \end{pmatrix} \end{aligned} \quad (21)$$

where

$$\mathcal{A} = \begin{pmatrix} 1/\sqrt{\beta} & 0 \\ \alpha/\sqrt{\beta} & \sqrt{\beta} \end{pmatrix} \quad \text{and} \quad \mathcal{O} = \begin{pmatrix} \cos \mu & \sin \mu \\ -\sin \mu & \cos \mu \end{pmatrix}. \quad (22)$$

As $\mathcal{O}$ and $\mathcal{A}$ have unit determinants, so does $\hat{R}$. The angle μ is commonly referred to as the *phase advance*. The tune

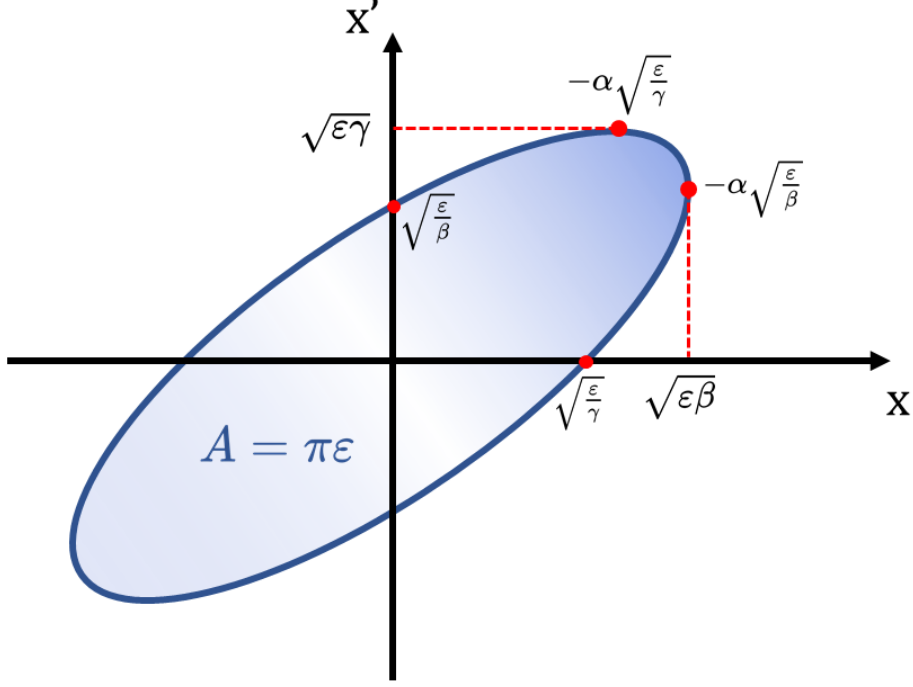


Figure 4: The phase space ellipse of x and x' of a particle trajectory.

$$Q = \frac{\mu}{2\pi} \quad (23)$$

describes the number of betatron oscillations per turn, or the phase space rotation angle. The tune is represented by Q_x along the x -axis, and by Q_y along the y -axis. Using the trace of $M(s)$ from Equation (11), we can compute its fractional part [11]

$$2 \cos(2\pi Q_x) = \text{Tr}[M(s)]. \quad (24)$$

In order to facilitate theoretical calculations, we use variables x_1 and x_2 in normalised phase space, related to the position x and angle x' by

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \mathcal{A} \begin{pmatrix} x \\ x' \end{pmatrix}. \quad (25)$$

Collectively we often denote the phase-space variables by a vector $\vec{x} = (x_1, x_2)^\top$.

From this point onward, we do not refer only to single particles but to ensembles of many particles. We always assume that the initial beam distribution is a multi-variate Gaussian, given in the two-dimensional case by the distribution

$$\psi(\vec{x}; \vec{X}, \sigma) = \frac{1}{2\pi\sqrt{\det \sigma}} \exp \left[-\frac{1}{2} \sum_{j,k=1}^2 \sigma_{jk}^{-1} (x_j - X_j)(x_k - X_k) \right], \quad (26)$$

where $\vec{X}$ is the centroid position and the 2×2 matrix σ is the beam matrix that describes the width and orientation of the Gaussian. Note that in coordinates of normalised phase space, the beam matrix σ of a matched beam is proportional to the unit matrix. For a matched beam, injected on axis, the proportionality constant is the emittance of the injected beam ε_0 .

2.3 Off-momentum effects

Until now, we assumed that all particles had the same momentum, without any momentum offset such that $\delta = 0$. However, some effects on the beam dynamics arise for particles with momentum $p = p_0(1 + \delta)$. These effects due to differences in the particle momentum are called *chromatic effects*, as the effect is equivalent to that of optical lenses on light of different wavelengths. The first type of chromatic effect, *chromaticity*, is the effect of momentum-dependent focusing of quadrupoles. In a thin quadrupole, a particle with momentum off-set $\delta = (p - p_0)/p_0$ with respect to the reference particle will have a tune shift

$$\Delta Q = -\frac{\hat{\beta}}{4\pi f_0}, \quad (27)$$

where $\hat{\beta}$ is the betatronic function evaluated at that point and f_0 is the focal length [13]. For instance in the x -plane, the chromaticity $Q'_x = \Delta Q/\delta$ for the entire ring is obtained by summing over the contributions of all quadrupoles

$$Q'_x = \Delta Q/\delta = -\sum_i \frac{\beta_i}{4\pi f_i}. \quad (28)$$

Considering the chromaticity of long quadrupoles with normalised gradient $k_1 = (\partial B_y/\partial x)/B\rho$, we have

$$Q'_x = -\oint \beta(s)k_1(s)ds, \quad (29)$$

where the integral extends over the whole ring and only picks up contributions from the quadrupoles where $k_1 \neq 0$. The second type of chromatic effect is *dispersion* D , which is momentum-dependent bending of dipoles. A dipole magnet that bends in the horizontal plane has the property of giving small horizontal position off-sets x_i , proportional to the momentum off-set $\delta = \Delta p/p$ and the dispersion D , defined as

$$D(s) = \frac{x_i(s)}{\delta}, \quad (30)$$

describing how much the trajectories are separated due to their momentum. We can simply add the dispersion to the transfer matrix

$$M_{1 \rightarrow 2} = \begin{bmatrix} C & S \\ C' & S' \end{bmatrix}, \quad (31)$$

with $C = \cos \phi$ and $S = \sin \phi$ (primes indicating derivatives with respect to s), to include the dispersion

$$\begin{bmatrix} \bar{x} \\ \bar{x}' \end{bmatrix} = \begin{bmatrix} C & S \\ C' & S' \end{bmatrix} \begin{bmatrix} x \\ x' \end{bmatrix} + \frac{\Delta p}{p_0} \begin{bmatrix} D \\ D' \end{bmatrix}. \quad (32)$$

To clearly see what happens with off-momentum particles, we look at their perturbed magnetic rigidity

$$B(\rho + \Delta\rho) = \frac{p_0(1 + \delta)}{q} \rightarrow \frac{\Delta\rho}{\rho} = \frac{\Delta p}{p_0} = \delta. \quad (33)$$

If we consider that the effective length of the dipole has not changed, we observe that

$$\begin{aligned} \theta\rho = l_{\text{effective}} = \text{constant} &\rightarrow \rho\Delta\theta + \theta\Delta\rho = 0 \\ \rightarrow \frac{\Delta\theta}{\theta} = -\frac{\Delta\rho}{\rho} = -\frac{\Delta p}{p_0} = -\delta, \end{aligned} \quad (34)$$

which means that off-momentum particles get a different deflection and thus a different orbit, described by Equation (35) and illustrated in Figure 5

$$\Delta\theta = -\theta\frac{\Delta p}{p_0} = -\theta\delta. \quad (35)$$

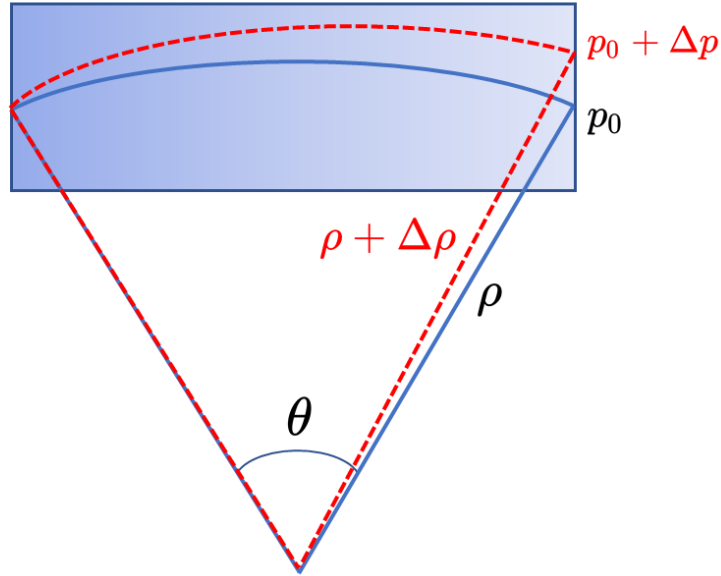


Figure 5: Distorted bending angle in a rectangular dipole due to a momentum offset.

We look back at the equation of motion in Equation (14) and conclude that for off-momentum particles, the solution is a sum of the homogeneous equation (on-momentum) and the inhomogeneous (off-momentum) equation

$$x(s) = x_H(s) + x_I(s), \quad (36)$$

which let us split the equations of motion in two parts

$$\frac{d^2 x_H}{ds^2} + K_x(s)x_H = 0 \quad (37)$$

$$\frac{d^2 x_I}{ds^2} + K_x(s)x_I = \frac{1}{\rho(s)}\delta. \quad (38)$$

Combining Equation (30) and (38), we can define the dispersion equation

$$D''(s) + K_x(s)D(s) = \frac{1}{\rho(s)}, \quad (39)$$

whose general solution can be obtained with perturbation theory and Green functions [12]. For a general matrix such as the one in Equation (32), the solution is

$$D(s) = S(s) \int_{s_0}^s \frac{C(\bar{s})}{\rho(\bar{s})} d\bar{s} - C(s) \int_{s_0}^s \frac{S(\bar{s})}{\rho(\bar{s})} d\bar{s}. \quad (40)$$

For a sector dipole, the bending radius ρ is constant and with its transfer matrix

$$M = \begin{pmatrix} \cos \frac{l}{\rho} & \rho \sin \frac{l}{\rho} \\ -\frac{1}{\rho} \sin \frac{l}{\rho} & \cos \frac{l}{\rho} \end{pmatrix}, \quad (41)$$

we find the dispersion $D(s)$ after a sector dipole to be

$$\begin{aligned} D_{\text{dipole}}(s) \Big|_{s=l} &= \sin \frac{l}{\rho} \int_0^l \cos \frac{\bar{s}}{\rho} d\bar{s} - \cos \frac{l}{\rho} \int_0^l \sin \frac{\bar{s}}{\rho} d\bar{s} \\ &= \sin \frac{l}{\rho} \left[\rho \sin \frac{\bar{s}}{\rho} \right]_0^l - \cos \frac{l}{\rho} \left[-\rho \cos \frac{\bar{s}}{\rho} \right]_0^l \\ &= \rho \sin^2 \frac{l}{\rho} + \rho \cos \frac{l}{\rho} \left(\cos \frac{l}{\rho} - 1 \right) \\ &= \rho \left(1 - \cos \frac{l}{\rho} \right), \end{aligned} \quad (42)$$

and its derivative with respect to s simply becomes

$$D'_{\text{dipole}}(s) \Big|_{s=l} = \sin \frac{l}{\rho}. \quad (43)$$

2.4 Linear imperfections from dipole errors

We consider a simple one-dimensional ring with horizontal tune $Q_x = \frac{\mu_x}{2\pi}$ and add a single thin quadrupole at a location with Twiss parameters α_0 , β_0 and γ_0 . In an otherwise ideal accelerator, we consider a single thin dipole field error at location $s = s_0$ with kick-angle $\theta = \Delta B dt / B\rho$, where $B\rho = p_0/e$ is the magnetic rigidity. The phase space coordinates just before (−) and after (+) the kick element at s_0 will be

$$x_- = \begin{pmatrix} x_0 \\ x'_0 - \theta \end{pmatrix}, \quad x_+ = \begin{pmatrix} x_0 \\ x'_0 \end{pmatrix}, \quad (44)$$

and the closed-orbit condition becomes

$$x_0 = \frac{\beta_0 \theta}{2 \sin(\pi Q_x)} \cos(\pi Q_x) \quad (45)$$

$$x'_0 = \frac{\theta}{2 \sin(\pi Q_x)} \left[\sin(Q_x \pi) - \alpha_0 \cos(\pi Q_x) \right], \quad (46)$$

from which we can immediately notice that for integer values of Q_x we have a pole [10]. Consequently, integer working points (or tunes) are normally avoided for accelerators, as *resonances* with chaotic particle motion of uncontrollable amplitude may lead to emittance growth. For integer tunes, the orbit kicks in every turn in a resonant manner, making the orbit unstable. The left side of Figure 6 schematically shows what happens in horizontal phase space when the tune Q_x is an integer in the presence of a dipole error. As the angular kick $\Delta x' = \theta$ occurs in the same direction for each revolution, the closed orbit does not exist. For this reason, the betatron tunes are normally designed to avoid integer values. If the tune is a half-integer, the angular kick from two consecutive revolutions will cancel each other, as illustrated to the right in the same figure [10]. In reality however, all kicks do not cancel each other. Quadrupole errors make the beam unstable at half-integer tunes and sextupoles at tunes that are multiples of $1/3$. All these values must be avoided in a circular accelerator.

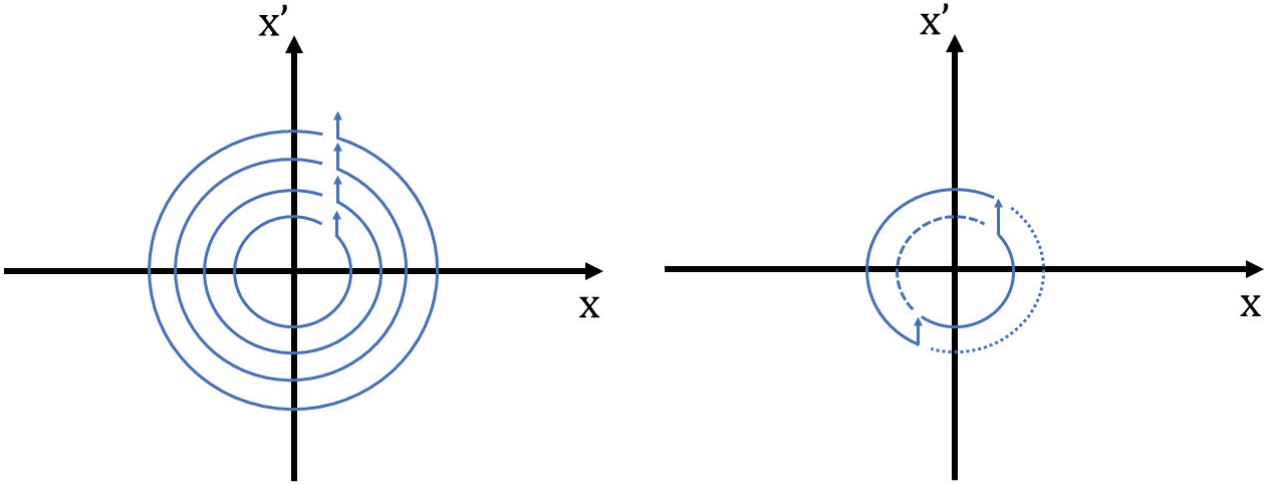


Figure 6: Illustration of the phase space effect of integer tune (left) versus half-integer tune (right).

In addition, such a dipole kick can originate from magnet misalignment of higher-order magnets, called *feed-down effects* [14]. For example, the kick from a misaligned quadrupole along the x -axis is found to be

$$\theta_q = K_x L \Delta x, \quad (47)$$

where K_x is the normalised quadrupole gradient in x , L is the element length and Δx is the displacement.

3 Beam transfer physics

3.1 Ring and transfer line beam dynamics

Depending on whether we have a circular accelerator in which a beam is transported over many turns or just a simple transfer line, the boundary conditions will be different. The one-turn map for a *circular machine* is

$$M_{0 \rightarrow L} = \begin{pmatrix} \cos(2\pi Q) + \alpha \sin(2\pi Q) & \beta \sin(2\pi Q) \\ -\frac{1}{\beta}(1 + \alpha^2) \sin(2\pi Q) & \cos(2\pi Q) - \alpha \sin(2\pi Q) \end{pmatrix}, \quad (48)$$

where Q is the tune [15]. A circular machine has a periodic solution, which imposes for one turn (closed ring): $\alpha_1 = \alpha_2$, $\beta_1 = \beta_2$ and $D_1 = D_2$. The condition of periodicity uniquely determines the solution $\alpha(s)$, $\beta(s)$, $\mu(s)$ and $D(s)$ around the whole ring. Mapping the coordinates of the single particles for each turn on any location, an ellipse forms in phase space. On the other hand, a *transfer line* describes a single pass from extraction to injection. The transfer map from point 1 to point 2 is given by

$$M_{1 \rightarrow 2} = \begin{pmatrix} \sqrt{\frac{\beta_2}{\beta_1}}(\cos \Delta\mu + \alpha_1 \sin \Delta\mu) & \sqrt{\beta_2 \beta_1} \sin \Delta\mu \\ \sqrt{\frac{1}{\beta_1 \beta_2}}[(\alpha_1 - \alpha_2) \cos \Delta\mu - (1 + \alpha_1 \alpha_2) \sin \Delta\mu] & \sqrt{\frac{\beta_1}{\beta_2}}(\cos \Delta\mu - \alpha_1 \sin \Delta\mu) \end{pmatrix}, \quad (49)$$

where the parameters at point 1 are given as input. For this case, no periodic boundary condition exists and the Twiss parameters are simply propagated from the initial point to the end. At any point in the transfer line, the optics functions $\alpha(s)$ and $\beta(s)$ are functions of the initial values α_1 and β_1 . Thus, for a single pass, the definition of the Twiss parameters is not unique. Also the dispersion function D depends on the initial values D_1 and D'_1 , and the line elements. Another significant difference with respect to a circular machine is that a change in the line elements only affects the downstream Twiss values, including the dispersion.

As beams have to be transported from the extraction point of one machine to the injection point of the next machine, the trajectories must be matched. An illustration of the Twiss parameters that have to be matched are illustrated in Figure 7. Additional constraints imposed by external factors include minimum bend radius, maximum available quadrupole gradient, magnetic aperture, cost and geology.

In the transverse plane, we can propagate for instance the horizontal position x and horizontal angle x' with respect to the reference particle trajectory when the transfer matrix M is known

$$\begin{bmatrix} x_2 \\ x'_2 \end{bmatrix} = M_{1 \rightarrow 2} \begin{bmatrix} x_1 \\ x'_1 \end{bmatrix} = \begin{bmatrix} C & S \\ C' & S' \end{bmatrix} \begin{bmatrix} x_1 \\ x'_1 \end{bmatrix}. \quad (50)$$

This step can analogously be done for the Twiss parameters [15]

$$\begin{bmatrix} \beta_2 \\ \alpha_2 \\ \gamma_2 \end{bmatrix} = \begin{bmatrix} C^2 & -2CS & S^2 \\ -CC' & CS' + SC' & -SS' \\ C'^2 & -2C'S' & S'^2 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \alpha_1 \\ \gamma_1 \end{bmatrix}. \quad (51)$$

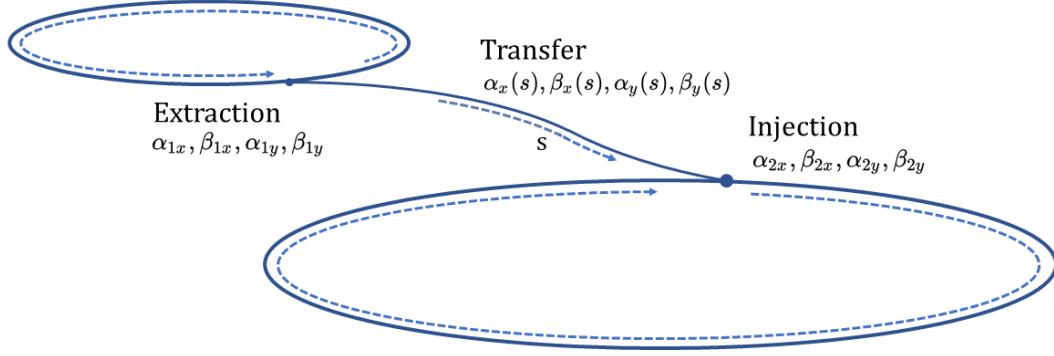


Figure 7: Parameters of extraction and injection that have to be linked.

In order to properly link two distant machines in longer transfer lines, one can facilitate the design by splitting the transfer line into two main categories:

- An initial and final matching section with independently powered quadrupoles, sometimes also with irregular spacing.
- A regular central section made up of e.g. FODO cells or doublets, where the quadrupoles occur at regular spacing and bending dipoles can be included, where the magnets are powered in series.

An example of the betatronic functions in such a lattice design is illustrated in Figure 8.

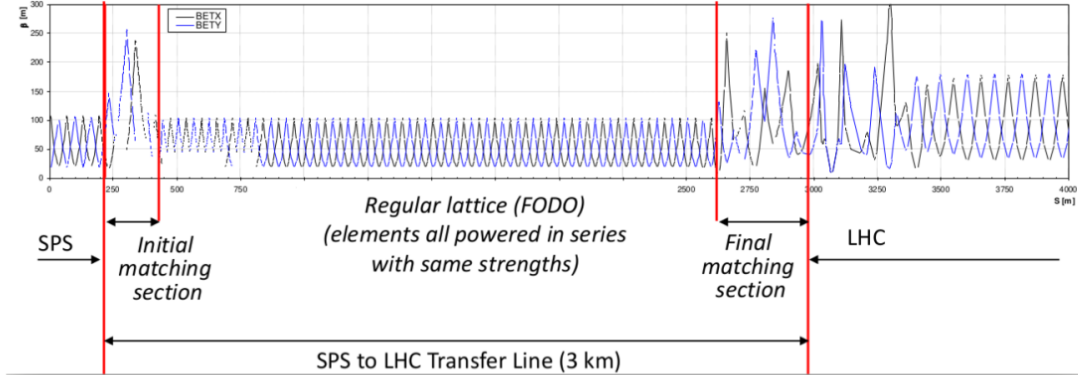


Figure 8: Example of betatronic oscillations in the SPS to LHC transfer line. The initial and final matching section are clearly customised depending on the beam parameter values in the SPS and LHC, whereas the middle section is a regular FODO-lattice [15].

3.2 Injection and extraction

Accelerators have limited dynamic range of energies to which particles can be accelerated, and therefore require a chain of stages to reach higher energies [16]. The beam needs to be properly transferred between each of these stages while minimising the beam loss, and placing the newly injected particles onto the correct trajectory with the correct phase-space parameters. The *injection* of a particle beam into a circular accelerator or accumulator ring at the right time, as well as the *extraction* the particles at the appropriate time, play a crucial

role for the performance of an accelerator complex. In addition, failed injection or extraction can cause severe machine damage. The injection and extraction can take place in either the horizontal or vertical plane (single-plane), or in both planes. An extraction or injection process that happens within one revolution is called *single-turn*, whereas a process that happens over multiple revolutions is called *multi-turn*. For instance, the SPS has a horizontal single-turn injection. The beam density at injection for hadrons can be limited either by space charge effects or by the injector capacity. If we cannot increase charge density, we can sometimes fill the horizontal phase space to increase overall injected intensity over a multi-turn process [17]. In this study, we focus on the single-plane single-turn injection, illustrated in Figure 9, which is relevant for the SPS.

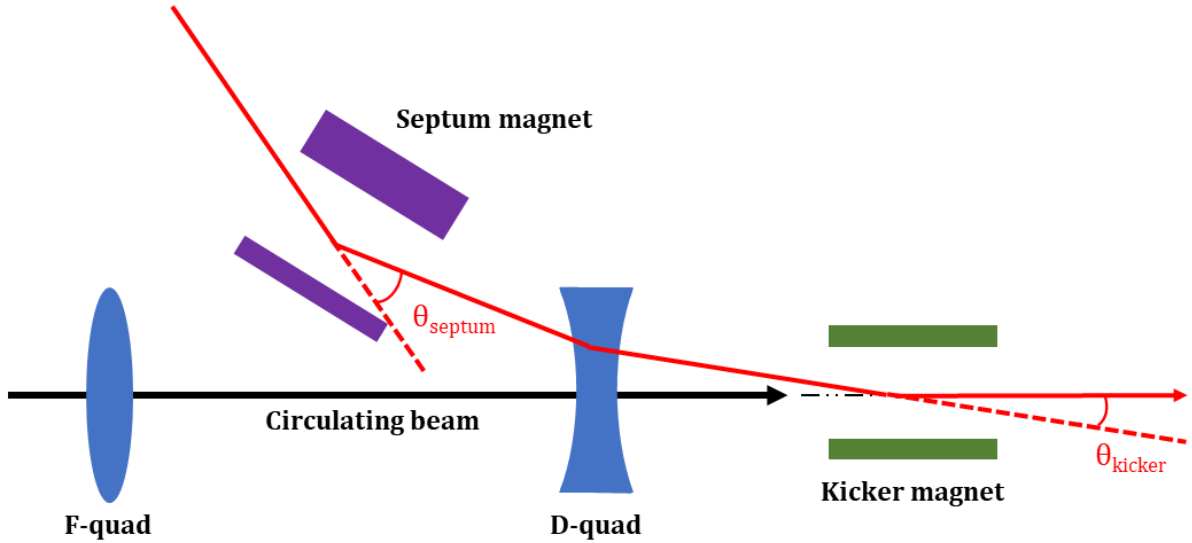


Figure 9: Illustration of the principle of single-turn injection.

Some special electromagnetic elements are used for the injection and extraction. A *kicker* magnet is a fast-pulsed dipole magnet that can reach the maximum field in less than one revolution, typically within hundreds of nanoseconds. A *septum* magnet is a special insertion magnet that define the borderline between the ring and the transfer line (TL). The septum magnet has an inner wall (often with 2-20 mm thickness) that separates two distinct field regions in order to selectively deflect particle beams where desired - the region of the circulating beam has ideally no field, and the region of the injected beam has full field strength for full deflection. An *electrostatic septum* is a DC electrostatic device with a very thin (about 0.1 mm) septum between zero field and high-field region. For a single-turn injection or extraction, the septum magnets are used rather than the electrostatic septum, as septum magnets can reach higher beam deflections. In order to match the injected beam trajectory to the machine closed orbit for a single-plane injection system, a system of kickers and septa is used. The septum deflects the beam onto the closed orbit at the centre of the kicker, whereas the kicker compensates for the remaining angle. The septum and kicker are placed on either side of a defocusing quadrupole to minimise the kicker strength needed for the injection.

4 Emittance growth after injection

Emittance is a key figure of merit both for circular and linear colliders, as the luminosity depends directly on the horizontal and vertical emittance [18]. In a collider such as the LHC, keeping the emittance as low as possible means higher likelihood for particle collisions and thus higher luminosity. Higher luminosity is desirable to achieve more collision data for the LHC experiments. The analysis of emittance growth at injection is of high importance for the rest of this study, as theoretical understanding of injection errors and tolerances for the injection error complements the development of the optimisation framework.

The emittance of an injected beam into a ring ultimately depends on the its initial position and angle at the point of injection, but also whether its Twiss parameters are equal to those of the ring. If these parameters do not match those of the circulating beam, the distribution of injected particles is distorted and decays into a distribution with larger emittance. An initially Gaussian beam degenerates into a distribution which is no longer Gaussian. This process is called *decoherence*. An example of decoherence after many turns is shown in Figure 10.

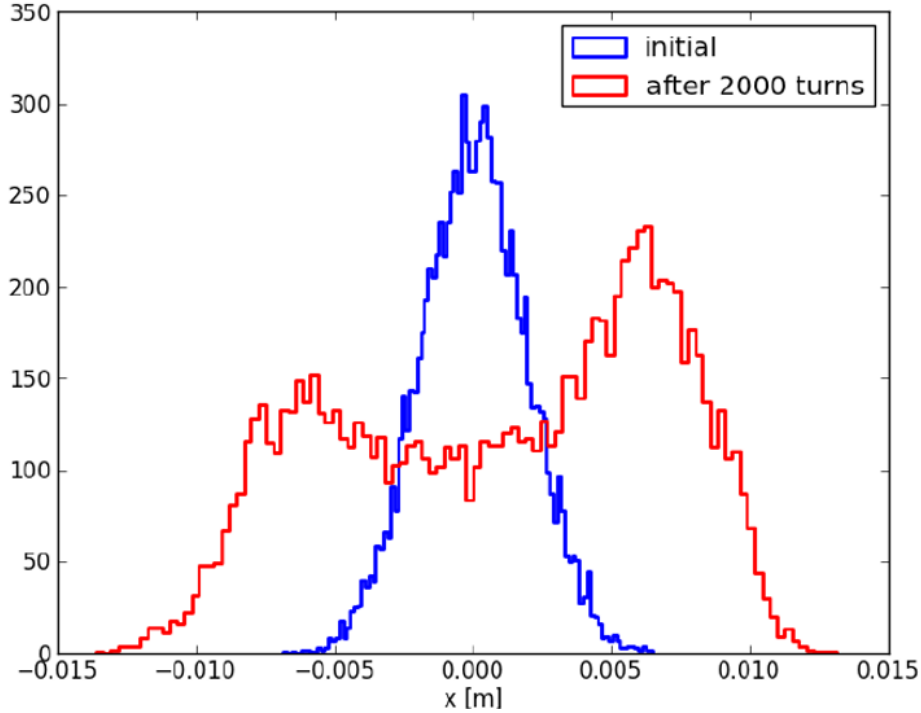


Figure 10: Illustration of the decoherence of a Gaussian beam due to filamentation from a steering error in X' , distorting the beam that is no longer Gaussian after many turns. Image credits: [18]

This effect is caused by a spread in betatron frequencies, and the oscillations of the injected beam is different to the betatron oscillations around the equilibrium orbit of the circulating beam. Decoherence may be due to many factors, including chromaticity, finite momentum spread, or amplitude-dependent tune shift. Amplitude-dependent tune shift caused by displacement (also called steering error) and by betatron mismatch is illustrated in Figure 11. Displacement, or error in the initial position X and angle X' at the point of injection, can lead to a tune shift ΔQ of the injected beam with respect to the circulating beam. Here $X = \langle x \rangle$ refers to the centroid (the first moment) of the beam, which is the expectation value in e.g. position of the distribution. Over many turns, this filamentation process causes the beam to degenerate into a non-Gaussian beam, as seen in Figure 10, with higher emittance. Also beta-

tron mismatch - Twiss parameters of the injected beam not matching those of the circulating beam - leads to the phase space ellipse of the injected beam not matching that of the circulating beam, resulting in tune shift, filamentation and ultimately larger emittance.

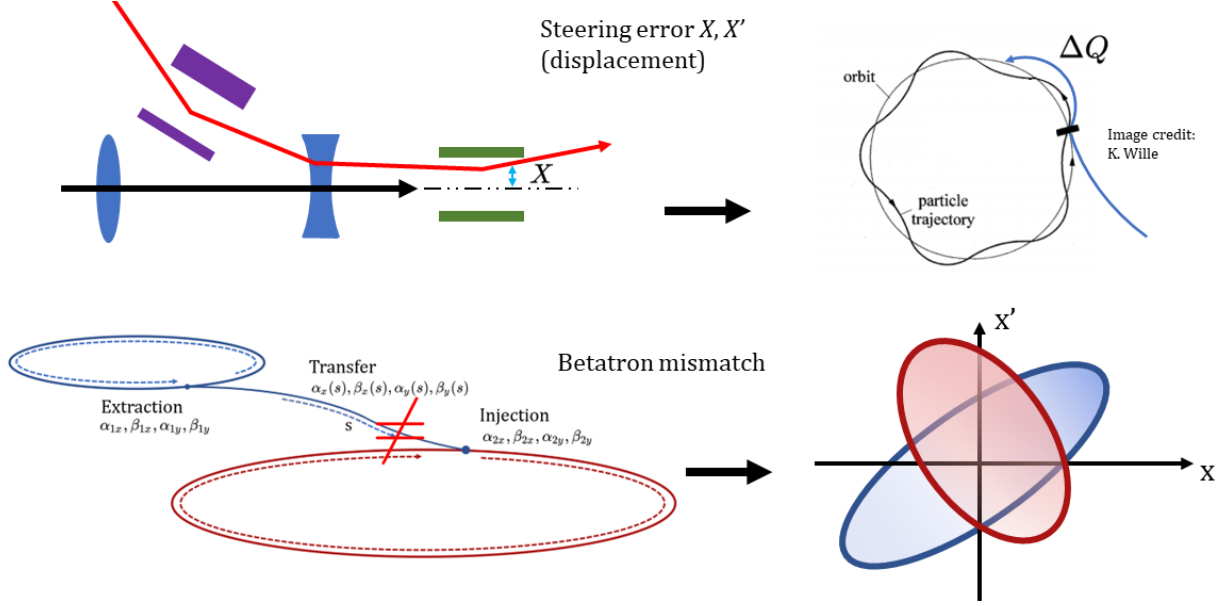


Figure 11: Illustration of two major injection error types - steering error (displacement) of the centroids X and X' , and betatron mismatch. Displacement causes tune shift ΔQ of the injected beam which leads to emittance growth. Betatron mismatch distorts the phase space ellipse of the injected beam with respect to the circulating beam, leading to tune shift and emittance growth.

The decoherence of displaced beams due to amplitude-dependent tune shift was previously analysed in [19, 20, 21]. In our recent published study [2], we extend this analysis of the decoherence by considering both the first and second moments of a beam over many turns - considering both displacement and betatron mismatch. We also follow the evolution of the beam matrix σ and the emittance ε as the beam decoheres. We start the analysis by adding an amplitude-dependent term to the phase shift per turn ϕ_x in the horizontal plane due to normal betatron phase advance $\mu_x = 2\pi Q_x$, assuming uncoupled optics

$$\phi_x = \mu_x + \kappa_{xx} (x_1^2 + x_2^2) + \kappa_{xy} (x_3^2 + x_4^2) = \mu_x + \vec{x}^\top \bar{\kappa}_x \vec{x}, \quad (52)$$

where $\vec{x}^\top$ is the transpose of $\vec{x}$ and $\bar{\kappa}_x = \text{diag}(\kappa_{xx}, \kappa_{xx}, \kappa_{xy}, \kappa_{xy})$ parametrises the amplitude dependence. $x_1, \dots, x_4$ denote the variables in normalised phase space, related to the position x and angle x' by the transformation $\mathcal{A}_x$ from Equation (22). The first moments ($\hat{X}_1, \hat{X}_2$) and one of the second moments $\langle \hat{x}_1^2 \rangle$ after n turns, identified by a caret, can be expressed as

$$\hat{X}_1 + i\hat{X}_2 = e^{-in\mu_x} \left\langle e^{-in\vec{x}^\top \bar{\kappa}_x \vec{x}} (x_1 + ix_2) \right\rangle \quad (53)$$

$$\langle \hat{x}_1^2 \rangle = \langle (x_1 \cos n\phi_x + x_2 \sin n\phi_x)^2 \rangle \quad (54)$$

where the angle brackets denote averaging over the initial d -dimensional Gaussian distribution from Equation (26). All the other first and second moments for d dimensions are calculated in a similar manner, described in detail in [2]. The integrals are solved with well-known multivariate Gaussian integral identities, enabling us to calculate the sigma matrix elements after n

turns. For instance, the first sigma matrix element $\hat{\sigma}_{11}$ turns out to be

$$\hat{\sigma}_{11} = \langle (\hat{x}_1 - \hat{X}_1)^2 \rangle = \langle \hat{x}_1^2 \rangle - \hat{X}_1^2, \quad (55)$$

where the other matrix elements are calculated in a similar manner - valid for any mismatched and transversely coupled beam that additionally is injected off-axis with $\vec{X} \neq 0$. We then explore the subsequent emittance growth in the two-dimensional case $d = 2$ when $\kappa_{xx} = \kappa$, both for displaced and/or mismatched beams. We prepared a MATLAB script, presented in [2], to follow the centroid motions $(\hat{X}_1, \hat{X}_2)$, the beam matrix $\hat{\sigma}$ from Equation (55), and the emittance $\hat{\epsilon} = \sqrt{\det \hat{\sigma}}$ for a number of turns n . The case of an initial displacement with a matched beam is shown in Figure 12, and a displaced and unmatched beam is shown in Figure 13. The vertical axes are normalised to appropriate powers of ε_0 .

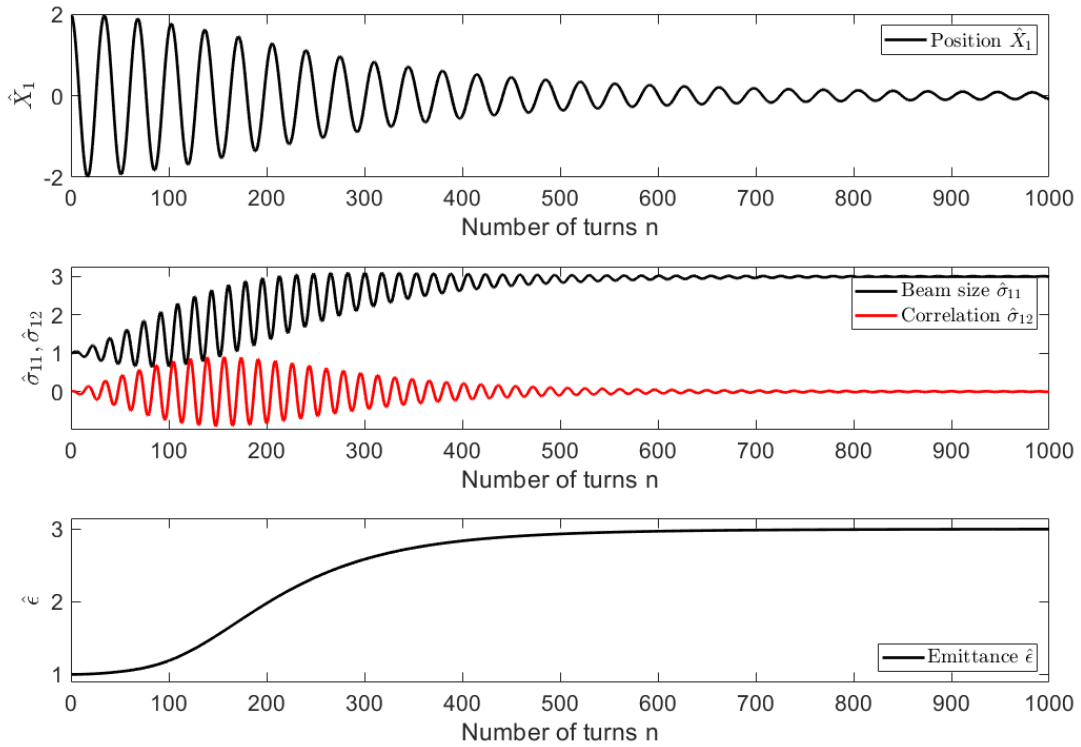


Figure 12: The centroid X_1 (top), the beam matrix elements σ_{11} and σ_{12} (middle), and the emittance (bottom) as a function of the turn number n for a matched beam that is injected with initial offset $X_1 = 2$. The parameters used are $\mu/2\pi = 0.028$ and $\kappa\varepsilon_0 = 0.001$ to illustrate the beam dynamics [2].

In both Figure 12 and 13, the beam offset oscillates and decreases slowly. The beam size $\hat{\sigma}_{11}$ oscillates at about twice the frequency of the centroid, before settling at its final value. The matched displaced beam has almost twice the asymptotic value $\hat{\sigma}_{11}$ with respect to the unmatched displaced beam. The correlation $\hat{\sigma}_{12}$ intermittently increases, albeit in different manners for Figure 12 and 13, but then settles to a zero-valued equilibrium value as the beam decoheres. The bottom plot shows the emittance $\hat{\epsilon}$, which has tripled compared to the initially injected beam in Figure 12, whereas $\hat{\epsilon}$ has increased by 70% for Figure 13. The equilibrium

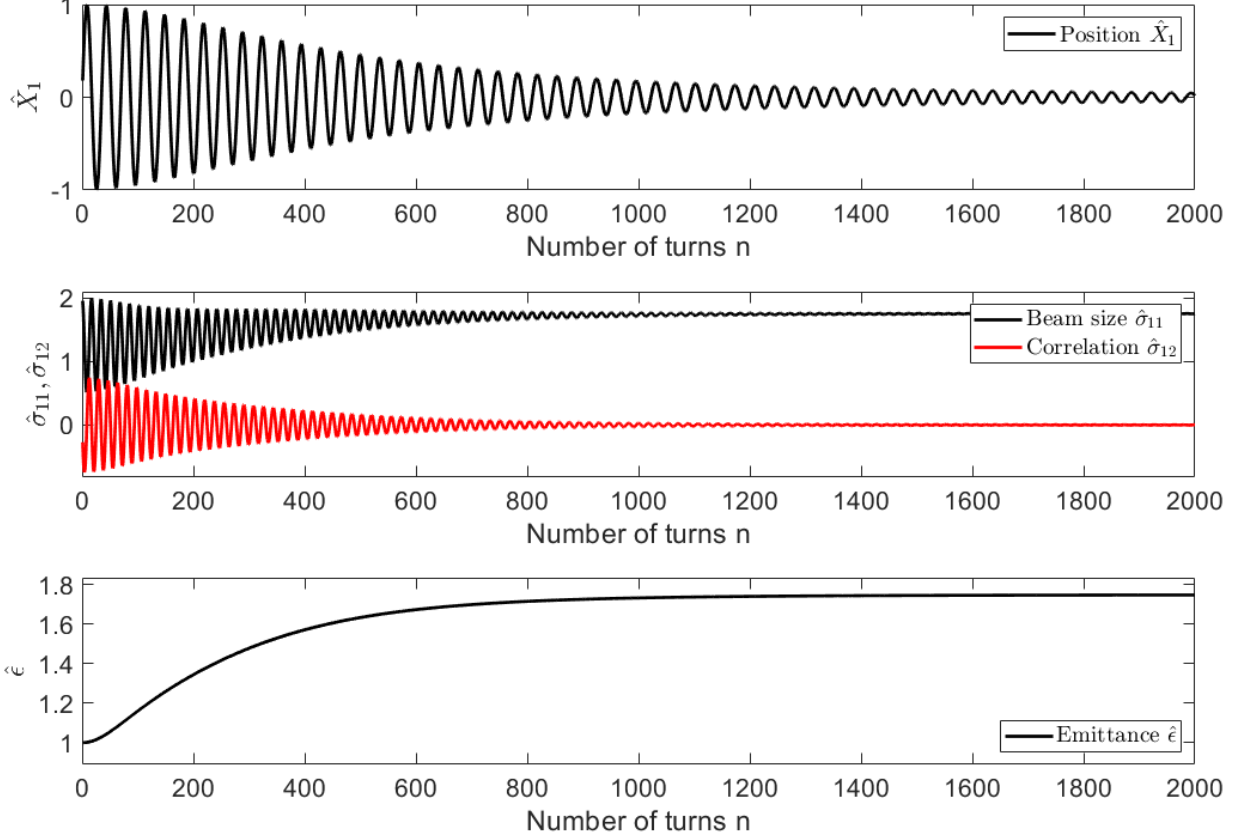


Figure 13: The parameters $\hat{X}_1$, $\hat{\sigma}_{11}$, $\hat{\sigma}_{12}$, and emittance as a function of the turn number n for a beam that is injected with a steering error at injection $X_2 = 1.0$, and a beta function β_0 that is twice the matched value β . All other parameters are equal to those used in Figure 12 [2].

emittance value that is reached after the decoherence has asymptotically played out its role is

$$\hat{\epsilon} = \varepsilon_0 B_{mag} + \frac{1}{2} (\gamma_x X^2 + 2\alpha_x X X' + \beta_x X'^2) \quad (56)$$

$$\text{with } B_{mag} = \frac{1}{2} \left[\left(\frac{\beta_0}{\beta_x} + \frac{\beta_x}{\beta_0} \right) + \beta_x \beta_0 \left(\frac{\alpha_x}{\beta_x} - \frac{\alpha_0}{\beta_0} \right)^2 \right],$$

where B_{mag} is the betatron mismatch factor by which the emittance asymptotically increases. α_x , β_x , and γ_x are the Twiss parameters of the ring. We also recall from Equation (19) that $\frac{1}{2} (\gamma_x X^2 + 2\alpha_x X X' + \beta_x X'^2) = (X_1^2 + X_2^2)/2$ is the Courant-Snyder invariant of the centroid due to the initial displacement. For a matched beam, $B_{mag} = 1$. If we set $\beta_0/\beta_x = 2$ $X_2 = 1$, we observe an increase of the emittance to $\hat{\epsilon} = B_{mag} \varepsilon_0 + X_2^2/2 = 1.75 \varepsilon_0$, in agreement with the final value of emittance growth shown on the bottom panel in Figure 13.

Interestingly enough, the result in Equation (56) shows that amplitude-dependent tune shift does not change the oscillation amplitude of individual particles in the distribution, as the asymptotic emittance growth agrees with the value caused by decoherence due to chromaticity and momentum spread (Section 8.2 in [13]). Amplitude-dependent tune shift only changes the initial behaviour of the oscillation, but reaches the same asymptotic value as in the case of chromaticity and momentum spread. In addition, we also present in [2] how to include transverse coupling, dispersion and chromaticity into the calculations of amplitude-dependent

tune shift. In particular for the case of chromaticity, we show that various form factors appear that affect the transient behaviour, but the final equilibrium values presented in Equation (56) remain unaffected, both for bunched and unbunched beams.

5 The Super Proton Synchrotron

The Super Proton Synchrotron (SPS) at CERN is the last accelerator in the Large Hadron Collider (LHC) injection chain. The SPS has a circumference of 6.9 km, and can accelerate protons up to 450 GeV. It was constructed in 1976, and has since its inception mostly provided beams for fixed-target experiments. SPS also plays an important roll in providing beams for the LHC. The beams need to be pre-accelerated before entering the LHC, in a so-called *injector chain*. SPS and its place in the CERN accelerator complex can be seen in Figure 14.

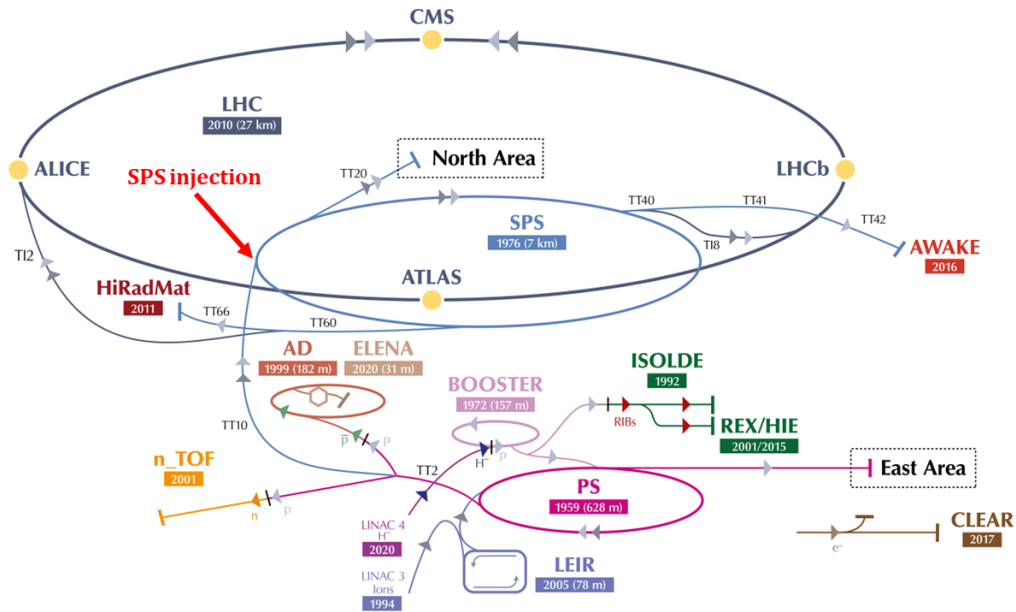


Figure 14: The CERN accelerator complex with experimental areas. The injection system of SPS is marked with a red arrow. Image credit: CERN

This process takes place in several stages through accelerators of various energies, connected through so-called *transfer lines* (TLs). The LHC injection chain for protons, which happens during the majority of operation time, is composed of

- LINAC4 - accelerates beams up to 160 MeV;
- Proton Synchrotron Booster (PSB) - consists of four superimposed rings and accelerates the beam to 1.4 GeV;
- Proton Synchrotron (PS) - accelerates beams up to 26 GeV, and is the first accelerator that is shared between proton and lead beam production;
- **Super Proton Synchrotron** - the second-largest accelerator at the CERN complex, accelerates proton from 26 GeV to 450 GeV;
- Large Hadron Collider - accelerates proton beam from 450 GeV up to about 7 TeV

The SPS was initially designed to achieve beam energies of 300 GeV, but reached 400 GeV on its initial commissioning in 1976. It is composed by 1317 normal-conducting electromagnets, including 744 main dipoles (6.26 m each in length) and 216 quadrupoles [22]. The SPS lattice is based upon a periodic FODO configuration, with a super-symmetry of six. Each of these six periods is equally spaced around the ring, and contains an arc with 16 FODO cells and a *long straight section* (LSS) of two cells. The LSS also contains the insertion elements. LSS1 contains the injection, the scraper (measuring how much the beam deviates from its original distribution) and dump systems. LSS2 focuses on the slow extraction channel for the North Area experiments, whereas LSS3 is dedicated to the RF system. LSS4 and LSS6 are responsible for the fast extraction systems to deliver beams to LHC, AWAKE and the HiRadMat (High Radiation to Materials) area, where the effect of high-energy and high-intensity beams on different materials for accelerator components is studied. LSS5 contains the UA9 experiment and other instrumentation. The UA9 collaboration explores how small bent crystals can perform better beam collimation for the LHC [23].

The SPS machine operates in cycles, meaning that each injected beam is only stored for a few tens of seconds. Depending on the purpose, different *optics* - how the beam-focusing magnets are distributed and configured in the lattice - can be used. The three main optics used during normal operations are **Q20**, **Q26** and **SFTPRO**. Q20 is used for beam delivery to LHC, whereas Q26 is the previous optics version for LHC but is now used together with Q20 for high-intensity bunched beams of LHC type. On the other hand, SFTPRO is used for slow extraction for the fixed-target experiments of NA. The parameters of these three SPS optics are displayed in the table in Table 1. The Q20 and Q26 optics logically take their name from the nearest integer value of their respective tunes. At the transition energy γ_t , the revolution period is completely independent from the individual particle energy. If $\gamma_t > \gamma$ (where γ is the Lorentz factor of the beam), particles with positive energy offset have larger revolution frequency.

SPS Optics	LHC Q20	LHC Q26	SFTPRO
Horizontal tune, ν_x	20.13	26.13	26.62
Vertical tune, ν_y	20.18	26.18	26.58
Natural chromaticity, Q'_x/Q'_y	-22.7/-22.7	-32.6/-32.63	-33.51/-33.46
Maximum betas, $\beta_x \approx \beta_y$ [m]	105	105	105
Minimum betas, $\beta_x \approx \beta_y$ [m]	30	20	20
Maximum dispersion, D_x [m]	4.5	8	4.4
Transition energy, γ_t	18	22.8	22.8
Phase advance per cell, $\mu_x \approx \mu_y$ [°]	67.5	90	90

Table 1: Summary of major parameters of the three main SPS optics types [14].

5.1 SPS physics users

The physics users of the SPS can be found in the CERN complex in Figure 14. SPS acts as a multi-purpose accelerator and provides beams to the LHC (450 GeV at extraction), to the North experimental Area (NA) fixed-target experiments, to HiRadMat, and to AWAKE. As of today, up to 4.5×10^{13} protons of 400 GeV can be extracted in the frame of 4.8 s to NA, and

to other user facilities. The SPS has been used to probe the inner structure of protons, the imbalance between matter and anti-matter and new exotic forms of matter. The Nobel Prize in Physics 1984 was awarded for the discovery of the W and Z bosons thanks to research carried out at the SPS [24]. The mode of operation was very different from today, as SPS was a $p\bar{p}$ collider at this time.

There are two main beam types for the users: LHC beam types, and SFTPRO beam types for the NA experiments. For this purpose, the SPS has two extraction modes: fast and slow. The fast extraction process ejects the circulating beam in one revolution, whereas the slow extraction process delivers a constant flux of particles at 400 GeV for the NA fixed-target experiments. The fixed-target experiments are the main users of the SPS. For the fast extraction, LSS4 and LSS6 contain the extraction elements. Normally, a kicker magnet is first triggered and the beam is deflected from its reference orbit towards the aperture of a septum magnet with a magnetic field.

The slow extraction allows for a continuous extraction from a synchrotron during a large number of turns. The slow extraction channel is installed in LSS2, and uses the one-third integer horizontal resonance by combining different phase space properties of the particles [25]. The slow extraction constituents are: a set of bumper magnets, an electrostatic septum (ZS), two magnetic septa (MST and MSE) and four extraction sextupoles (installed in LSS1, LSS3, LSS4 and LSS5). When the beam has been accelerated to its extraction energy, the slow extraction process starts. By turning on the extraction sextupoles in a quasi-adiabatic way, the extraction bump brings the beam closer to the electrostatic septum and the horizontal tune Q_x approaches the resonance condition

$$Q_x = N + p/3, \quad (57)$$

where N and p are integers. Particles located outside the stable regions will be pushed towards higher amplitudes, and the electrostatic septum wires slowly "cut out" the extracted beam from the circulating one. The magnetic septum gives the final deflection to make the beam exit the synchrotron.

5.2 SPS injection sequence

In the case of the SPS injection, the beam is delivered from the Proton Synchrotron (PS) via the TT10 transfer line as illustrated in Figure 14 of the CERN accelerator complex. The injection happens only in the horizontal plane during a single turn, as shown in Figure 9. The sequence with the main elements is illustrated below in Figure 15. There are four septa (denoted MSI) and two main types of kicker magnets: MKP-S and MKP-L, whose magnetic field and mechanical aperture differ slightly. The present top magnetic field is about 30 mT for MKP-S and 46 mT for MKP-L.

Figure 16 shows a three-dimensional CAD-model of the present kicker system for the SPS injection, which consists of three MKP-S units, powered in series, and an independently powered MKP-L (the rightmost tank). Each vacuum tank contains magnet modules. This configuration was installed in 1999 to reduce the SPS kicker magnets' rise time in order to adjust for the LHC bunch spacing [26]. There is also a dedicated beam dump denoted TBSJ, employed in the case of injection failure. For a safely dumped beam, the kicker magnets are turned off and the beam hits the mechanical aperture of the beam dump TBSJ such that no other machinery

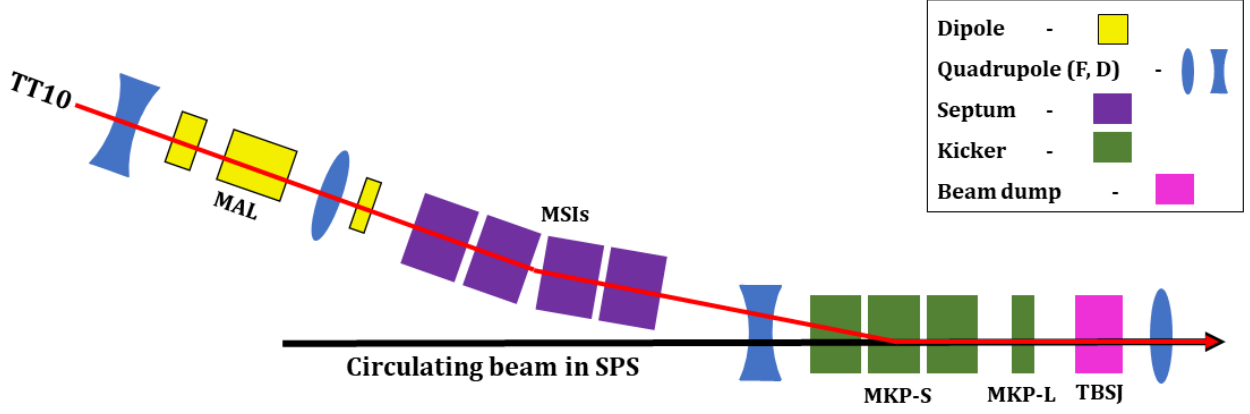


Figure 15: Illustration of the TT10/SPS injection sequence.

is damaged.

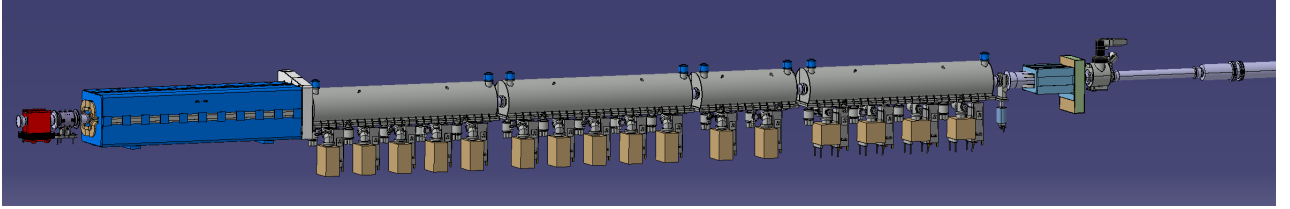


Figure 16: Three-dimensional schematic of the defocusing quadrupole (blue), the three MKP-S kicker tanks (grey cylindrical) and the MKP-L kicker tank (rightmost grey cylindrical tank) of the present SPS injection system. Image credit: Pavlina Trubacova

5.2.1 MSI and MKP injection elements

In order to place the newly injected particles onto the correct trajectory with correct phase space parameters to match the circulating beam, a combination of septa and kicker magnets are used. Septa have normally slower rise- and fall times than kicker magnets, but can achieve higher fields. In one septum region, the electromagnetic field is zero and no force is exerted on the circulating beam, as illustrated in Figure 9. In the other region, a homogeneous fields gives a deflection to the injected beam. Hence, a septum provides space separation between the injected/extracted beam and the circulating beam, whereas a kicker magnet provides time selection of the beam to be injected/extracted [27]. An illustration of a typical septum magnet is shown in Figure 17, where the thin separation between the deflected injected beam and the circulating beam is illustrated. In SPS, the four septa are illustrated in purple in Figure 15, with a total deflection angle of 4.2 mrad. In the present injection system in the SPS, the thickness of the separation between the circulating and injected beam is 40 mm. For economic reasons, parts of the septa from the PS Booster and parts from today's SPS injection system can be reused to form a new movable septum conductor (also called *blade*), where the second half of the blade is 20 mm thick instead of today's 40 mm for the whole blade. The reduced thickness of the proposed MSI blades enables more space for the beam envelope and for higher flexibility in the solutions.

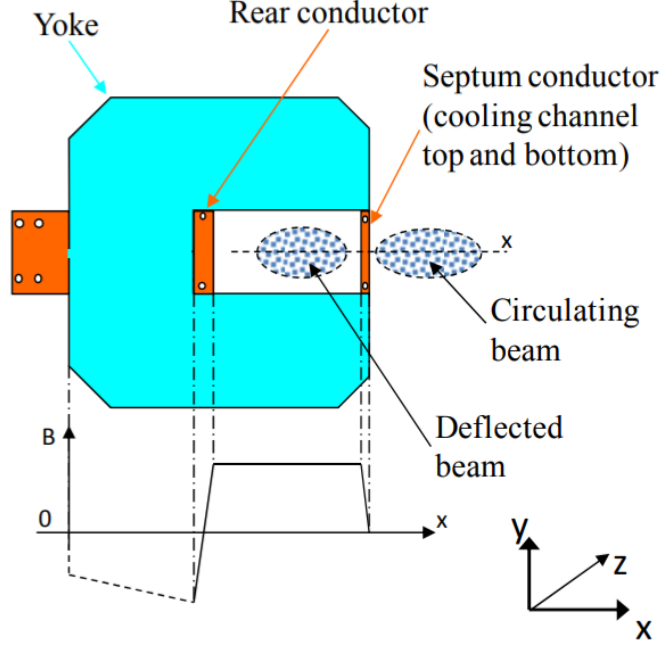


Figure 17: Illustration of example septum magnet, here the direct-drive pulse type [27].

In the SPS injection system, kicker magnet modules are placed under vacuum contained in tanks for improved performance. The total deflection angle for the MKP magnets is 4.7 mrad, for all beam types. A cross-section CAD drawing of the MKP-L type kicker magnet tank from the SPS is presented in Figure 18, whereas the MKP-S presently has no available CAD drawing. The four magnet modules inside the vacuum tank are clearly visible, coloured for the sake of illustration. Figure 19 shows the CAD drawing of one of these modules, where the beam passes through the slit behind the red horizontal red plate in the foreground. The horseshoe-shaped magnet is composed of a stack of several units, separated by the yellow-coloured plates in the drawing. At CERN, these kicker magnets normally have a very fast rise and fall time (100s of ns) in order to kick the injected beam batches onto the closed orbit of the circulating beam without disturbing it. The pulse of such a kicker magnet from the SPS injection, used for Q20 and Q26 optics, is illustrated in Figure 20.

One of the main limitations to the maximum achievable brightness of the SPS beam are instabilities caused by various *impedances*. Impedances are the Fourier transform of the wake fields arising from interactions between the machine aperture and the beam itself, as a result of the impulse response of an accelerator component. The wake field produced by a beam passing by a discontinuity in the machine aperture affects the motion of subsequent particles, significantly perturbing the beam motion above a certain threshold, denoted the instability threshold. The SPS kicker magnets are the main contribution to the total impedance budget, accounting for about 40% of the total vertical tune shift caused by impedance [28]. Due to difference in design and aperture, more beam-induced heating is caused by the MKP-L magnet than the MKP-S magnets. Consequently, the kickers for a new SPS injection system are preferably of the MKP-S type, or any possible new design that is more suitable in terms of beam-coupling impedance. In this report, we only use the MKP-S type module for this reason.

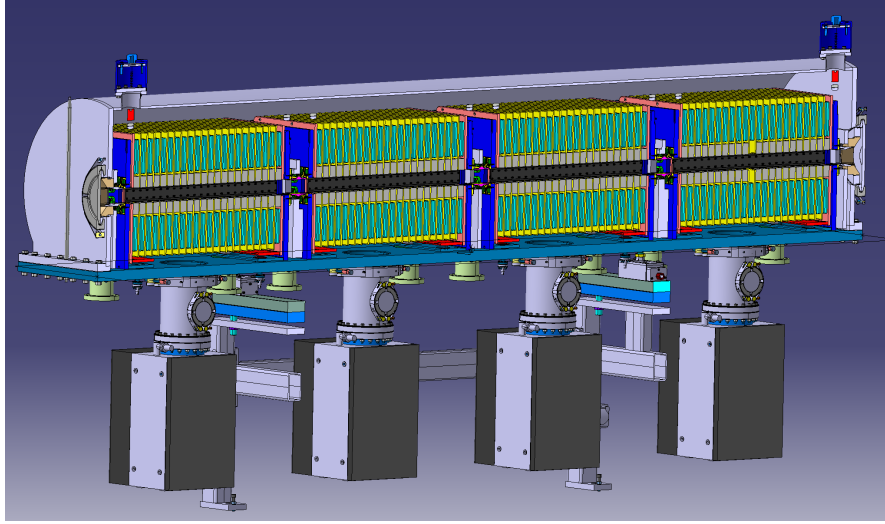


Figure 18: Cross-section of MKP-L kicker tank, where the four magnet modules are clearly visible. Image credit: Pavlina Trubacova

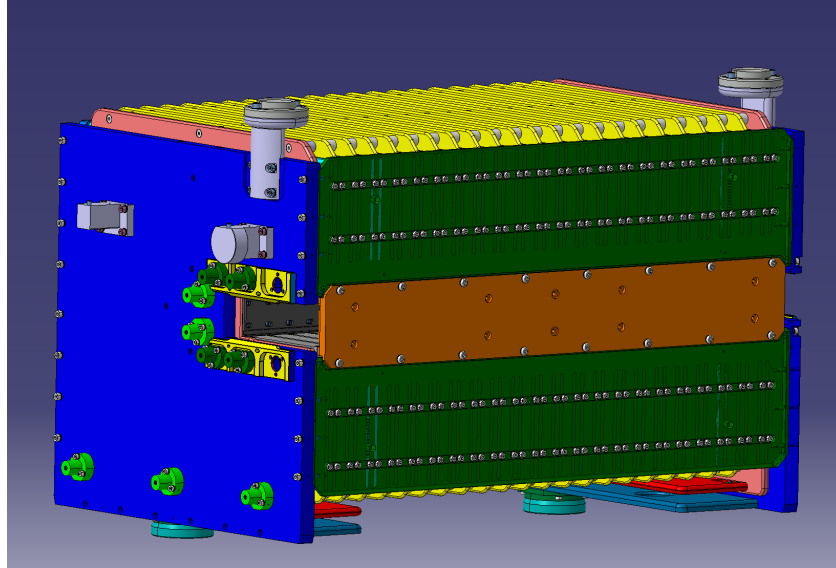


Figure 19: Three-dimensional technical CAD drawing of MKP-L kicker magnet, where the beam passes through the slit behind the horizontal red plate. Image credit: Pavlina Trubacova

5.3 The PS-SPS Transfer Line

The TT2/TT10 transfer beam line between the PS and SPS transfers both protons and heavy ions. Its location in the CERN accelerator complex is highlighted in Figure 14. The LHC imposes a strict emittance budget for the delivered beam with maximum allowed emittance blow-up of 17%, of which only a small fraction is assigned to mismatch at injection. The transfer line TT10, joining the PS with the SPS, was designed for a low-momentum beams (10 GeV/c) with a small momentum spread ($\text{rms} < 2.5 \times 10^{-4}$). The β -functions in the transfer line are comparable with the ones in the SPS machine, with a maximum value of around 120 m [1].

There are two main types of optics in this context: fixed-target optics (SFTPRO) and fast ejection optics (Q20/Q26). In fixed-target optics, a transverse phase oscillation exchange takes

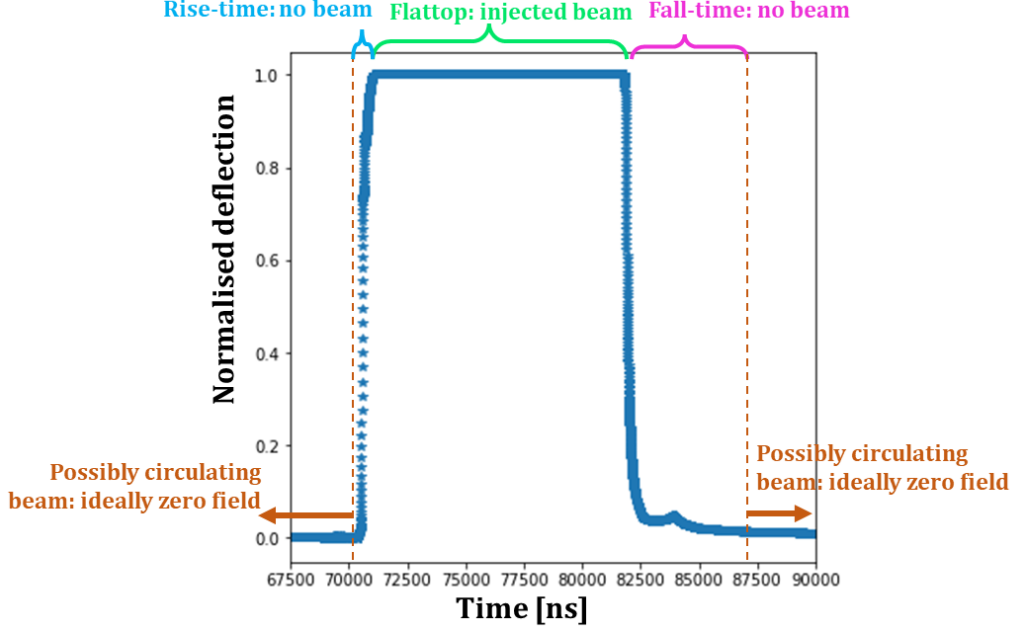


Figure 20: Example of the magnetic field deflection of a kicker magnet for the SPS injection.

place to put the smallest emittance (horizontal due to beam slicing in continuous extraction) in the vertical plane where the SPS aperture is most restrictive. In the fast ejection optics, this phase gymnastics is suppressed. This second type is mainly used for LHC beams, with higher momentum (26 GeV) but also with higher momentum spread (3.0×10^{-3}). The dispersion in TT10 reaches 5 m vertically and 6 m horizontally. The relatively large values in dispersion and momentum spread require multiple beam position monitors (BPMs) and beam steering elements for monitoring and adjustment, which have been installed in the straight section.

In addition, an injection optimisation system with transverse feedback damps injection oscillations and keep the emittance blow-up within tolerance if they are below 3.5 mm horizontally or 2.5 mm vertically. Both dispersion and the β -functions have to be matched in the transfer line in order not to blow up the emittance. An emittance monitoring system in the SPS stores the beam profile for many turns in order to monitor this process. Also multipolar errors in the magnets can cause higher-order resonances and lead to emittance blow-up. The working points of SPS are listed in Table 1.

Injection errors can also cause emittance blow-up of a transversely stable beam, caused by decoherence in the injection oscillations due to tune spread in the beam, a phenomenon called filamentation [13]. The time-varying part of these injection errors are due to fast and slow errors. The fast errors originate from *ripples* on the injection kicker pulse and are different for each injected bunch, when the flat top of the boxcar-shaped kick in Figure 20 is no longer flat. Slow variations of transfer line magnets and septa cause differences between successive injections. These factors require active damping, with feedback in place in order to stabilise the beam. Some studies in 1999 also looked in particular at the dispersion mismatch due to the small emittance and large momentum spread of the beam [29].

5.4 SPS injection error tolerances

With the help of the SPS optics functions repository and the analytical expressions of the emittance growth from Section 4, we analyse the requirements for the steering errors and the Twiss parameters of an injected beam to keep the emittance growth lower than 1 % and 5 %, as presented in [2]. We expand Equation (56) up to second order in the deviations from their respective design values $\Delta\beta = \beta_0 - \beta$, $\Delta\alpha = \alpha_0 - \alpha$, ΔX , and $\Delta X'$. In particular, we look at the Q20 optics when SPS is used as an injector to the LHC, with the horizontal Twiss parameters [30] at the injection point being $\beta = 44.5$ m and $\alpha = -0.96$. The initial horizontal emittance is $\varepsilon_{0,x} = 1.26 \times 10^{-7}$ mrad. The error tolerances for the asymptotic emittances values below 1 % and 5 % respectively are displayed in Table 2. Although the relative errors for the Twiss parameters can be relatively high without causing too much emittance growth, the emittance is sensitive to steering errors $\Delta X'$ in momentum due relatively large value of β at the injection point. For instance, more than $20 \mu\text{rad}$ of $\Delta X'$ leads to increased emittances above the 5 % level. This sensitivity to injection errors motivates the high priority of achieving stably injected beams, especially for the displacement X and X' .

Table 2: The error tolerance levels for mismatch and steering errors for the injection into the SPS [2]. The nominal emittance is $\varepsilon_0 = 1.26 \times 10^{-7}$ mrad and the Twiss parameters at the injection point are $\beta = 44.5$ m and $\alpha = -0.96$ [30].

Tolerance level	$\Delta\beta/\beta$	$\Delta\alpha$	ΔX [mm]	$\Delta X'$ [μrad]
1 %	0.14	0.14	0.24	7.5
5 %	0.32	0.32	0.54	16.8

6 Injection system design with hierarchical optimisation

In this section, we present the construction of a new injection system for the SPS using a numerical hierarchical optimisation framework with realistic hardware assumptions, following closely the results already published in [7]. We use MAD-X [31], a general-purpose tool for charged-particle optics design and studies, as the main simulation software for the beam dynamics in the SPS injection system. For our study, we use `cpymad`, a Cython binding to MAD-X for full control and access to a MAD-X interpreter within Python environments. MAD-X simulations of the horizontal and vertical beam envelope with the machine aperture in the present SPS injection system are shown in Figure 21, with the MSI and MKP magnets marked.

To achieve on-axis injection, we desire zero-valued horizontal position x and momenta x' at the last quadrupole in the injection sequence to avoid oscillatory motion in the SPS. We also use the acceptance A - the number of standard deviations of the beam size that fits into the available space between the orbit and the mechanical aperture - as a figure of merit to determine whether the available space is sufficient

$$A_{x,\min} = \min\left(\frac{\text{aper}_x(s) - |x(s)|}{\sigma_x(s)}\right), \quad (58)$$

where $\text{aper}_x(s)$ is the horizontal mechanical aperture of the machine (circular radius), with y replacing x for the vertical case. For the horizontal plane, the beam envelope size $\sigma_x(s)$ is

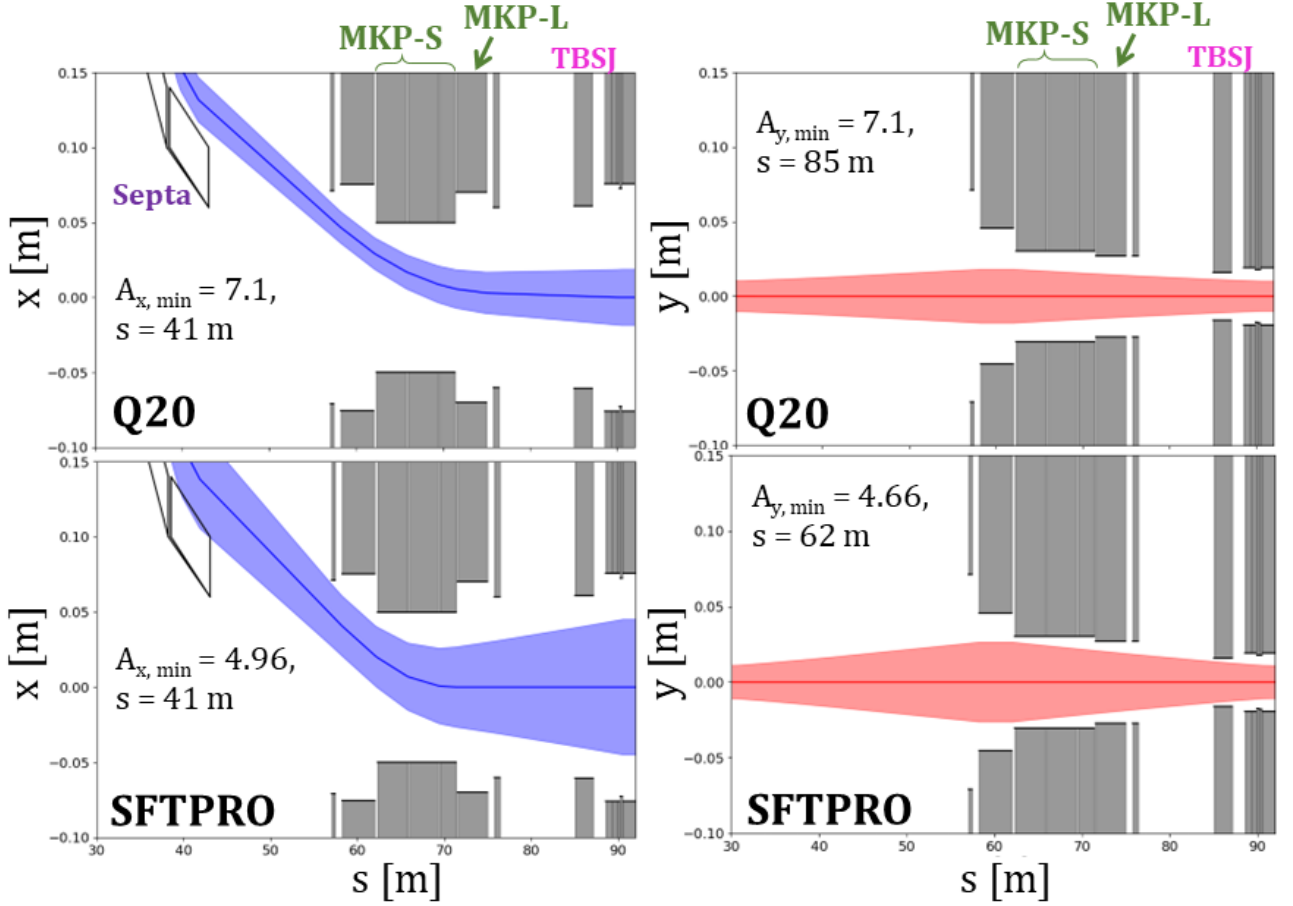


Figure 21: MAD-X simulations of today's horizontal and vertical envelope injection for Q20 (very similar to Q26) and SFTPRO optics, with the minimum acceptance $A_{\min}$ for each case and at which s it occurs. The plotted beam envelope widths $n_x = n_y = 5\sigma$ in terms of standard deviations are used for Q20/Q26, whereas $n_x = 6\sigma$, $n_y = 4\sigma$ are used for SFTPRO due to different emittance values ε_x and ε_y with respect to the other optics.

defined as

$$\sigma_x(s) = \sqrt{\beta_x(s)\varepsilon_x + (D_x(s)\Delta p/p)^2}, \quad (59)$$

where $D_x(s)$ is the dispersion function, ε_x is the geometric emittance, and $\Delta p/p$ is the fractional momentum difference [14]. If $A_{x,\min}$ or $A_{y,\min}$ are too low, the beam envelope will pass too close to the mechanical aperture and beam losses will occur.

When we construct the optimisation framework, we first define the *fixed* parameters: the incoming beams, the quadrupole (and higher-order) magnet strengths, and the number of septum units, some of which are re-used for economic reasons as described in Section 5.2.1. We then define the main *actors* (degrees of freedom) for the simulations: the positions and strengths of the given septa and kickers, but also the number of kicker magnets. We can vary the number of MKP magnet modules n_1 and of vacuum tanks n_2 , in which the modules are contained. The *objectives* and *constraints* in our optimisation studies are: stable outgoing beam ($x = x' = 0$), preserved beam ($A_{x,\min}$ at least larger than today), and a safely dumped beam if the MKP magnets fail. Once these basic requirements are fulfilled, we also minimise the kicker magnet resources as a secondary objective: the top magnetic field B (for easier magnet design)

and integrated kicker magnetic field $\int Bdl$ for minimal power consumption.

Thus, the lower level of optimisation consists in tuning the MSI and MKP magnet strengths and positions to obtain a stable preserved beam for a given magnet configuration for all optics. On the other hand, the higher level of the optimisation consists of building up the sequence by finding the ideal combination of n_1 and n_2 . In the present injection system, the MKP-S vacuum tank 1 and 2 contain five magnet modules each. We adopt the modular approach also for simulating the MKP length as a free parameter - we divide the current MKP-S vacuum tank length and strength by five to create a modular element, similar to the elements displayed in Figure 19. An n_1 number of such modules will then form one vacuum tank, with a total of n_2 tanks, as illustrated in Figure 22.



Figure 22: Illustration of the magnet modules contained in one MKP vacuum tank.

This approach of dividing a problem into several levels is called *hierarchical optimisation* [32] - a complex system is divided hierarchically into several simpler systems, where each level is optimised. The simplest case is called *bilevel optimisation* and consists of two levels, where one low-level optimisation problem is embedded within the outer high-level optimisation problem. To represent each level, we create a Python class object to form an *environment*, where the essential injection elements and parameters are initialised in the sequence for each optics. This object-oriented approach has the advantage that magnet configurations, beam dynamics and other parameter values as well as the MAD-X process itself are stored as attributes to the environment, enabling straight-forward gradual iteration of the injection system optimisation. More construction details and pseudo-algorithms can be found in [7].

The main flow of this study's multi-layer hierarchical optimisation is illustrated in Figure 23. The meta-optimiser (located in a wrapper function) first creates a global environment object, whose `step` method takes the number n_1 of MKP magnet modules and of tanks n_2 as input and returns objective function 1. Hence, the actors on this higher level are n_1 and n_2 . As a second step, the global environment creates an environment object for each optics with the given n_1 and n_2 for its injection sequence. The global environment then optimises the objective function 2 on a lower level for each optics, by varying the MSI and MKP positions and deflection strength for ideally delivered beams. As a third step, the global environment selects the optics environment with the highest value of $\int Bdl$, - this approach guarantees that we first satisfy the case requiring the highest integrated field. In our study, the magnet strengths for the other optics could always be re-optimised with these new magnet positions. The magnet positions of this optics environment (Q20 in our example) are returned, and the individual MSI and MKP strengths of the other optics environments are then individually optimised by the global environment for this magnet position configuration. As the last step, the global environment returns the value of objective function 1 to the meta-optimiser.

This whole iteration corresponds to one step of the high-level optimisation, with which the meta-optimiser progresses until the termination conditions have been fulfilled - normally when the relative improvement of the high-level objective function is below a certain threshold. To find the best combination of n_1 and n_2 at the higher level, we use a *genetic algorithm* (GA)

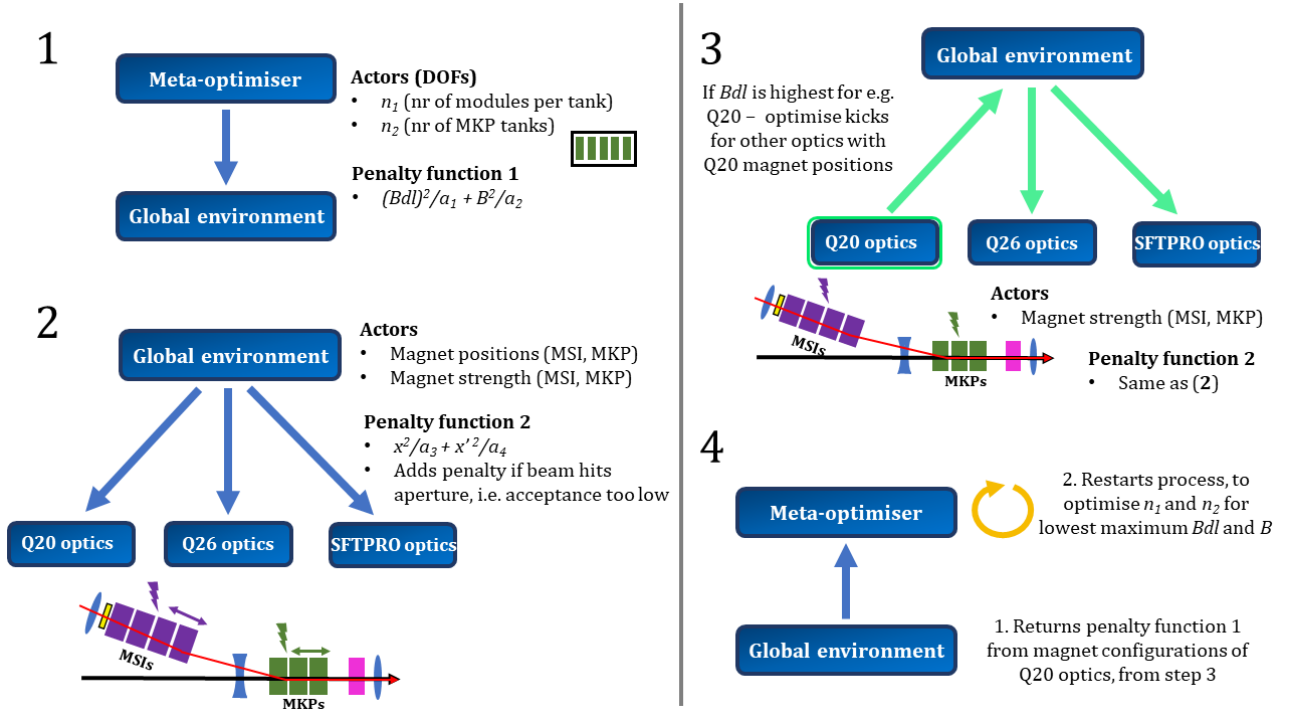


Figure 23: Illustration of the hierarchical optimisation scheme, where a_i are unit normalisation constants.

provided by the Python library `pymoo` [33]. GAs are inspired by the process of natural selection and belong to the category of so-called evolutionary algorithms. The main advantage is easy customisation with different evolutionary operators applied to a broad category of problems, making it straightforward to adapt the algorithm to our existing Python class methods. Figure 24 shows the generic flow of the GA.

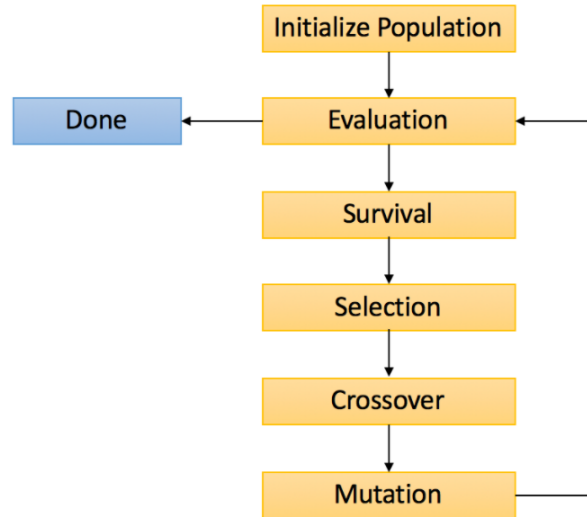


Figure 24: Illustration of the generic flow of a genetic algorithm [33]

A starting population of candidate solutions is sampled and their objective functions are evaluated. The fittest candidates survive according to the objective criteria and are selected for the crossover, combining the candidates into offspring where some mutations may occur according

to pre-defined probabilities for higher diversity. This flow represents one iteration, which is repeated until some termination conditions are fulfilled. In our case, the population consists of different (n_1, n_2) tuples that evaluated against the high-level objective function, where the fittest tuples survive for the next iteration and produce offspring in a crossover process. Even with a starting population size of 10 randomly sampled tuples, the population converges already after 10-15 iterations to the same candidate solutions.

The new injection system generated by the hierarchical optimisation is presented in Figure 25, with the ideal combination $n_1 = 3$ modules per tank, and $n_2 = 5$ tanks in total. The last two tanks are shifted horizontally by 1.9 cm to make the dumped beam fit better, which is possible as the dipole kicker field is uniform. Both the minimum horizontal and vertical acceptances $A_{x, \min}$ and $A_{y, \min}$ are higher than today, as desired. The relative increase in integrated field is 5-6% for Q20/Q26 with respect to the current system, but the injected dumped (un-kicked) beam reaches the TBSJ without the need of an extra dipole used today - a great simplification for future operations. The main drawback of this configuration is the fact that the external side of the last MKP tank is exposed to direct beam impact in case of self-triggering of the MKP while beam is circulating already. More studies are needed to evaluate alternative solutions.

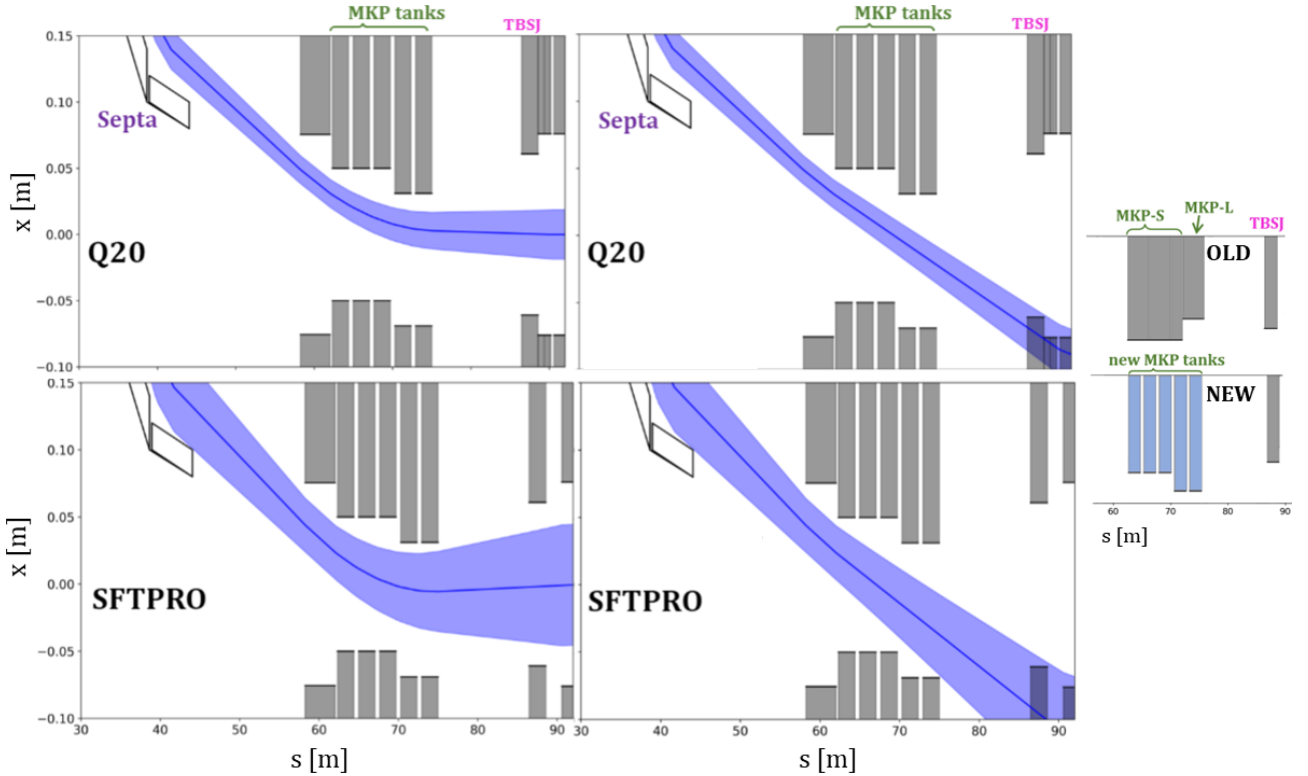


Figure 25: New injection system design with simulated beam envelope for injected beam (left) and dumped beam (right) for Q20 and SFTPRO optics, given by the hierarchical optimisation. A comparison between the old and the new proposed layout of the MKP tanks is shown to the very right.

In terms of physics, the optimisation process helps us to find the best balance between minimal total kick (i.e. magnet resources) versus obtaining a steeper injection angle for the beam to safely hit the beam dump TBSJ. The optimisation framework finds the best possible compromises between these conditions, by also maximising the usage of space around the MSI blade. A great advantage of the proposed new injection design is the uniform kicker tank design, where a malfunctioning unit can relatively easily be replaced as there is only one standard. In the present system, there are many different kicker tanks that each needs a separate spare. The pro-

posed new system also avoids the MKP-L type that generates more beam-induced heating than the MKP-S type, due to difference in aperture design between the magnet module types and to higher top fields for the MKP-L. Although the SPS injection system hardware components and magnet sequence are very specific, the main algorithmic flow of this study's hierarchical optimisation in Figure 23 can be applied to other accelerator systems where sequences need to be constructed at a higher level and optimised at a lower level. The object-oriented approach of the optimisation framework enables practical storage of the sequence elements, magnet configurations, Twiss command values and other parameters from the MAD-X process over many iterations. A detailed outline and generic pseudo-algorithm are presented in [7].

7 SPS injection kicker delay optimisation

7.1 Offline analysis and optimisation simulations

The SPS injection system, illustrated in Figure 9, happens for the injected beam within one revolution by placing the injected beam just after the already circulating beam. The kicker magnet is responsible to deflect the injected beam onto the orbit of the circulating beam, without interfering with the circulating beam. The main features of the SPS kicker magnets are described in Section 5.2.1, whose fast-pulsed waveform profile is shown in Figure 20. It is crucial that the flat top of the kicker waveform is timed precisely to act when the injected beam arrives, but at the same time keeping the magnetic field as low as possible when the circulating beam arrives. Ideally, this means placing the injected beam time stamp on the flat top of the waveform and the circulating beam time stamp where there is no field. However, due to technical limitations in the fast-pulsed regime of nanoseconds, the MKP flat top is not perfectly flat but contains ripples.

In addition, the rise and fall time are not perfect vertical steps. The so-called *batch spacing* dt_c of the beam - meaning the separation in time between the injected and circulating beam - is often required to be of the order 200-500 ns in the SPS, which means that the injected beam will often not receive full deflection and that the circulating beam will receive non-zero deflection. This mismatch at injection causes the beam - both circulating and injected - to experience *injection oscillations*. This process can potentially lead to emittance blow-up, detrimental for the beam quality and luminosity delivered to the SPS physics users. Theoretical calculations of emittance growth due to injection errors were presented in Section 4. In the present injection system set-up, the timing trade-off between distributing as much of the nominal kick as possible to the injected beam versus letting as little as possible of the deflection spill over to the circulating beam has been a laborious process done by hand. The three MKP-L type kicker tanks and 1 MKP-S type kicker tank contain in total 16 magnet modules, which are serially connected in pairs to 8 electrical switches for individual timing. These 8 individual switches (denoted *pulsers*) are visible in Figure 26, along with which MKP magnet modules they are connected to. The switches are also connected to a general delay $d\tau_{general}$ for all the waveforms. Thus, an operator in the SPS needs to consider $8 + 1 = 9$ degrees of freedom, where a slight change of one individual delay may completely alter the total kick.

As a first step to lay the foundations for an ideal balance of minimal injection oscillations, we extract kicker waveform data for the LHCPILOT cycle taken in October 2021 for the kicker magnet modules. This data set provides valuable samples to test out a pilot optimiser environment in Python. We convert the waveform data unit from voltage to actual deflection

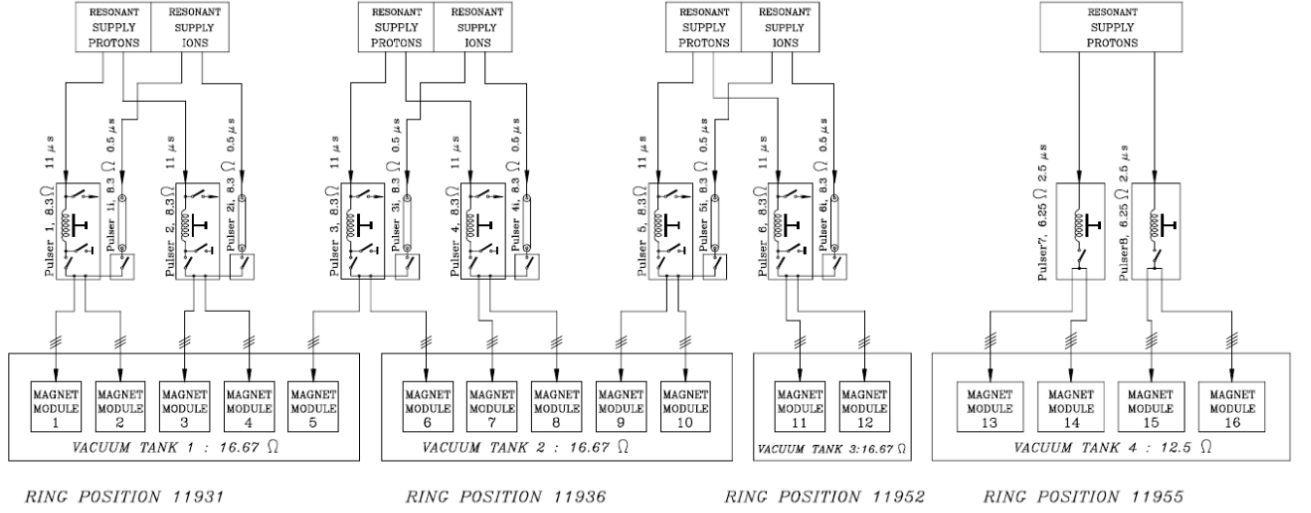


Figure 26: Schematic of the SPS kicker system layout with switches, installed in 1999 [26].

(in radians), using the assumption that the sum of the flat tops corresponds to the total kicker deflection as in the ideal case, without any injection oscillations. For a given batch spacing dt_c , the optimiser finds a general delay for the injected and circulating beam, and can also adjust individual delays by up to 50 ns. The objective is to minimise the squared sum of total injection error of the circulating and injected beam. The optimised general and individual delays are shown in Figure 27 for $dt_c = 250$ ns and $dt_c = 500$ ns. The corresponding simulated injection oscillations in MADX for the SPS injection, both for the injected and circulating beams, are shown in Figure 28.

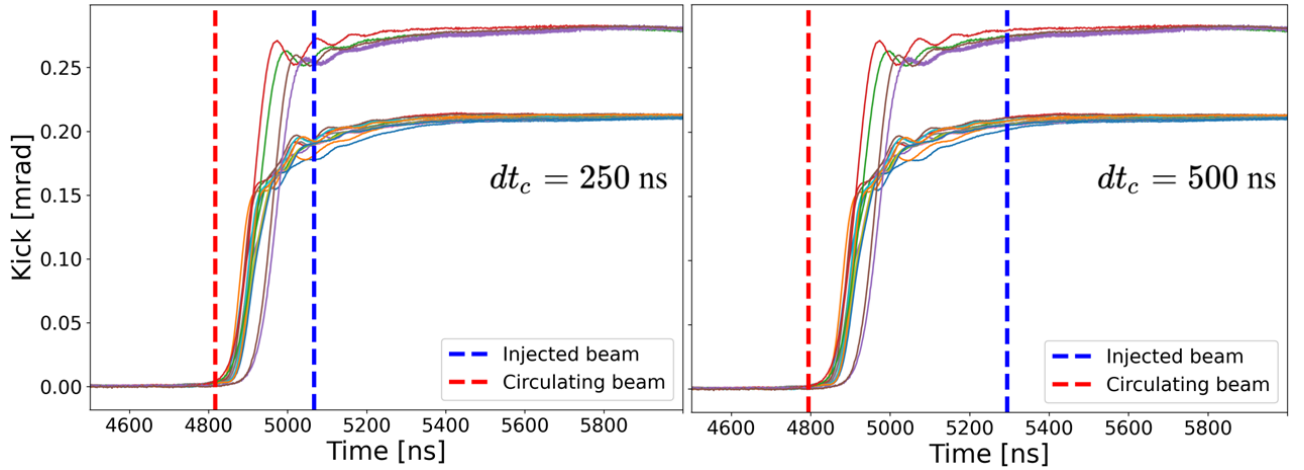


Figure 27: Optimised general and individual delays for standard SPS batch spacing $dt_c = 250$ (left) ns and $dt_c = 500$ ns (right), together with the kick waveforms for all kicker magnets.

A clear observation is that more generous batch spacing of 500 ns allows the injected beam to be placed almost entirely on the flat top, whereas the circulating beam can arrive almost untouched with enough delay, resulting in relatively small injection oscillations. On the other hand, the smaller batch spacing of 250 ns forces the optimiser both to give a small kick to the circulating beam and not enough kick to the injected beam, resulting in larger injection oscillations, as seen in Figure 28.

The increasing effect of injection errors on the action J_x in the horizontal plane - defined in

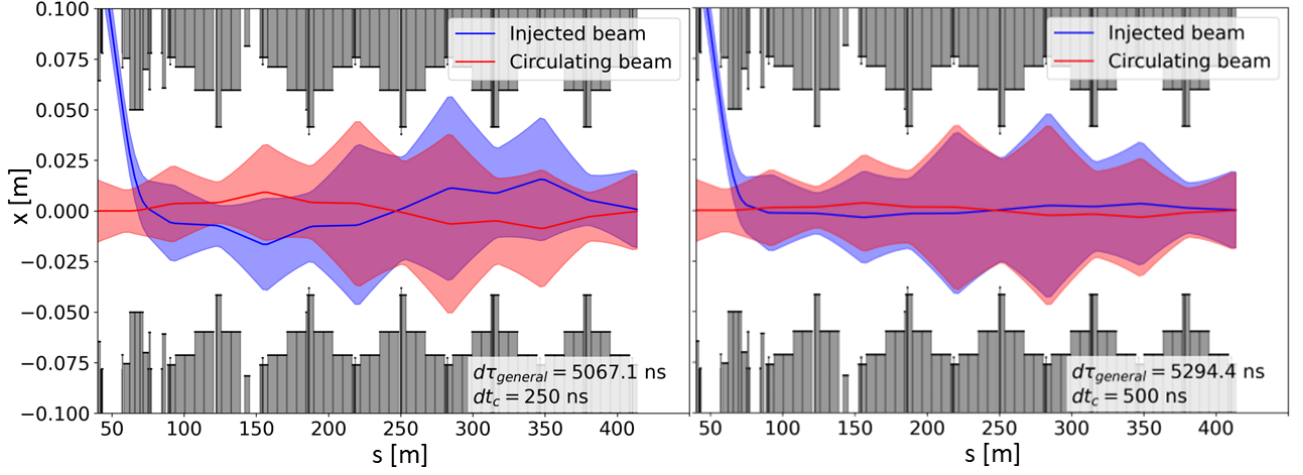


Figure 28: Simulated injection oscillations in the SPS of the injected and circulating beam, with batch spacing $dt_c = 250$ (left) ns and $dt_c = 500$ ns (right), using Q20 optics. Note the decreased injection oscillation amplitude when the batch spacing is increased, both for the circulating (red) and injected (blue) beam.

Equation (19) - is visible in Figure 29, depending on the MKP injection time stamp t_{MKP} . The minimum of the sum $|J_{x, \text{inj}}| + |J_{x, c}|$ occurs at the very same timestamp as the optimised t_{MKP} in Figure 27. An increased value of J_x also means an increase emittance ε_x .

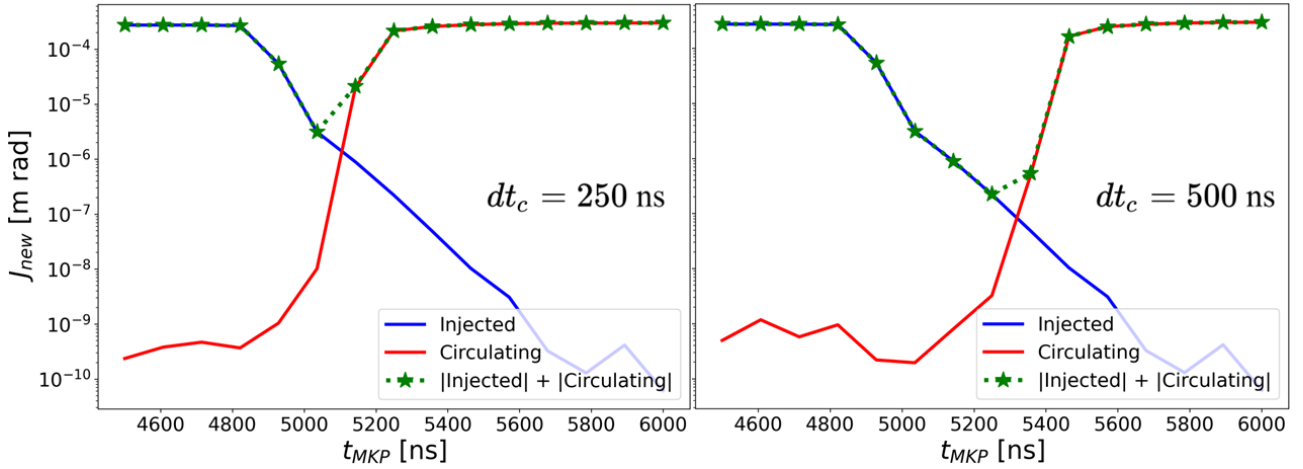


Figure 29: Simulated value of Courant-Snyder invariant J_x in the horizontal plane of injected and circulating beam due to injection errors, as a function of the general MKP delay t_{MKP} .

7.2 Online kicker delay optimisation

Based upon the notions and insights gained from the offline simulation tests with waveform data, we develop a Python optimisation environment following the standards of CERN-ML Common Optimisation Interfaces (COI) [34]. COI defines common interfaces that facilitate using numerical optimisation and reinforcement learning (RL) on the same optimisation problems. This makes it possible to unify both approaches into a generic optimisation application in the CERN Control Center (CCC). This object-oriented approach makes the use of the optimiser relatively straightforward for operators, and to test out different optimisation algorithms via the Graphical User Interface (GUI). The class methods are defined in such a way that

both numerical optimisation (minimising a penalty) and reinforcement learning (maximising a reward) can be tested with the same environment. In a similar manner to the hierarchical optimiser described in Section 6, the `step` method of the environment takes some action and returns the penalty or reward.

To communicate with the accelerator and access machine parameter values, we use the Java API for Parameter Control (JAPC), in particular PyJapc. PyJapc is a Python library to interact with the accelerator control system via JAPC, allowing for easy get and set commands of parameters [35]. We define a helper class that subscribes to the relevant variables via a callback function, such that we access the variables at injection of the right cycle. In this case, we test the optimiser environment on the LHCINDIV type beam cycle, which is a type of proton beam delivered to the LHC. For a given action (vector of 9 MKP time delays), the helper class also registers the beam position at the BPM monitor SPS.BPMB.51503 over many turns to investigate beam oscillations. We define the loss function

$$\mathcal{F}(\bar{x}, \bar{x}_c) = \bar{x}^2 + \bar{x}_c^2 + (\bar{x} - \bar{x}_c)^2, \quad (60)$$

where $\bar{x} = \frac{1}{2}(|\max(x)| - |\min(x)|)$ is the maximum injection oscillation amplitude for the injected beam, and analogously $\bar{x}_c$ represents the maximum injection oscillation amplitude for the circulating beam. The BPM electronics in the SPS can distinguish between the injected and circulating beam, and evaluates the beam until the oscillations have decayed, usually for 300 turns or more. The last term in Equation (60) ensures that the optimiser does not only focus on either the injected or circulating beam, but reduces the injection oscillations for both beams. This loss is recorded for each action, where a larger loss indicates higher beam oscillations for the injected and/or circulating beam. The optimiser normalises the action to $[-1, 1]$ from the given bounds, which we start by setting to 50 ns away from the initial value for the individual shifts in either direction, and 100 ns for the general shift. The first run with the optimiser using the BOBYQA algorithm is shown in Figure 30 for batch spacing $dt_c = 225$ ns.

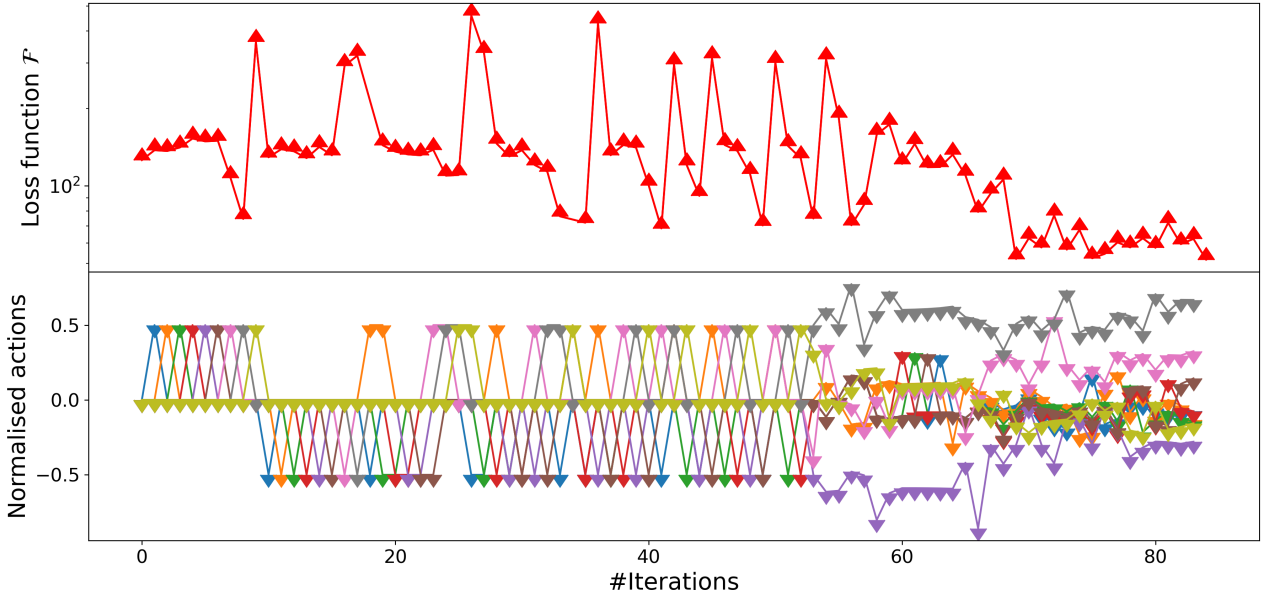


Figure 30: First results of the optimiser, for batch spacing $dt_c = 225$ ns of LHCINDIV cycle type.

In all the online implementation runs in this section, there was a single bunch per batch. However, we also generalised the framework to handle multiple bunches per batch. The circulating beam is then considered as the first circulating bunch after the injected bunch.

The BOBYQA algorithm (Bound Optimization BY Quadratic Approximation) is an iterative numerical optimisation algorithm forming quadratic models by interpolation to create a trust region of the search space [36]. The algorithm solves bound constrained optimisation problems without using derivatives of the objective function, which is suitable for computationally expensive non-linear functions where the derivative is not known. As seen in Figure 30, the optimiser first explores a trust region for each actor until about iteration 50 in order to form an adequate model before it starts to minimise the objective function - greatly reducing the loss function $\mathcal{F}$ within about 20 iterations before the beam cycle changed, and we had to stop the optimisation. Two cases when the beam was lost during initial exploration were assigned a high penalty, but places them outside the limits in Figure 30. Before the LHC commissioning, we had 4 of these optimisation sessions of 923 iterations in total with beam cycle relevant for our studies. In the last session, the used batch spacing was $dt_c = 200$ ns. The effect of the optimiser on the injection oscillations for the batch spacings $dt_c = 225$ ns and $dt_c = 200$ ns are shown in Figure 31.

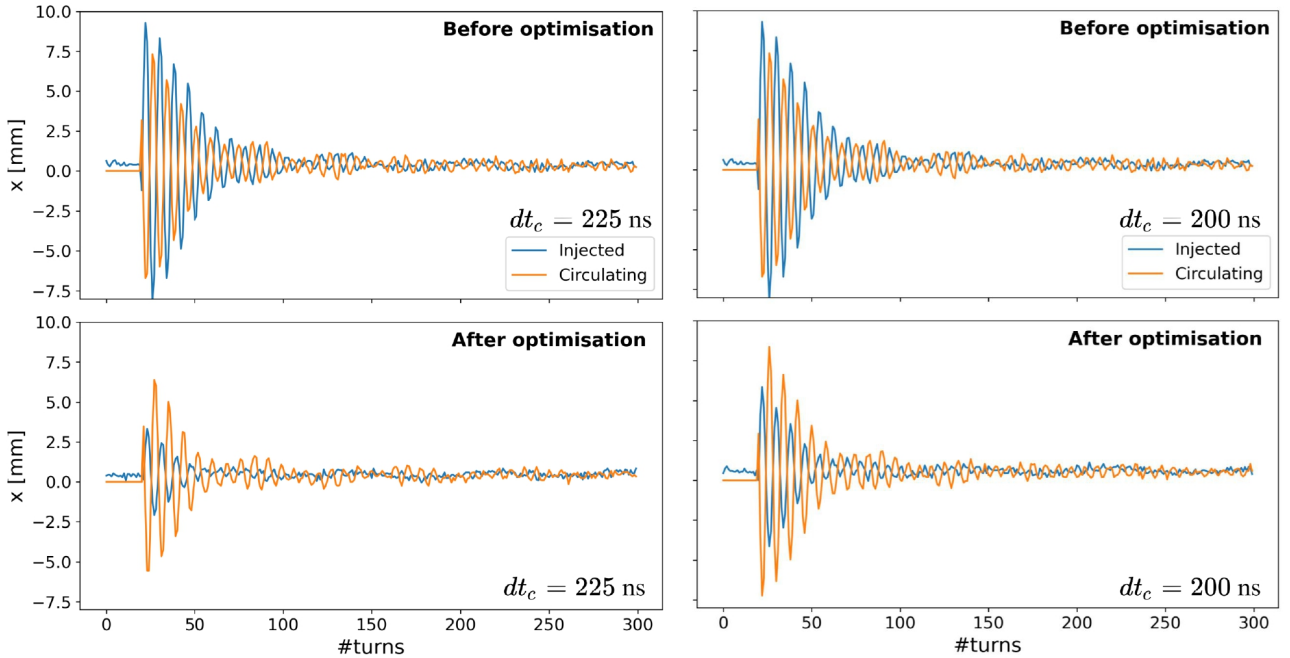


Figure 31: Injection oscillations in the SPS, recorded at the monitor SPS.BPMB.51503, before and after implementing the optimiser, using batch spacing $dt_c = 225$ ns (left) and $dt_c = 200$ ns (right).

For $dt_c = 225$ ns, the maximum injection oscillation amplitude for the injected beam decreases by almost 70%, and by more than 20% for the circulating beam. The case $dt_c = 200$ ns proves to be more difficult with tighter batch spacing, where the injection oscillations on the circulating beam increases by a few %, but is reduced by more than 40% for the injected beam.

7.3 Neural network model of the SPS injection

Due to the high number of activities and machine development at the CCC, we only had limited time to implement the optimiser environment described in Section 7.2. In total, there was time to test 923 objective function evaluation with the environment, where the BOBYQA algorithm was used. In order to compare different optimisation algorithms, we create a simple *neural network* (NN) to create a computationally inexpensive surrogate model and approximate the

behaviour of the beam position for different MKP kicker delays, based upon the recorded data. An artificial neural network (ANN) is a series of algorithms mimicking the biological neural network of animal brains. ANNs consist of a pool of simple processing units (“neurons”) which communicate by sending signals to each other over a large number of weighted connections [37]. For every neuron k , there is a state of activation a_k , which equivalent to the output of the neuron and determines whether a neuron is active or not. The connections between the neurons are defined by weight elements w_{jk} that determines the effect which the signal of neuron has on a neuron in the next layer. $w_{jk}^{(l)}$ denotes the weight for the connection from the k^{th} neuron in the $(l-1)^{\text{th}}$ layer to the j^{th} neuron in the l^{th} layer [38]. There is also a bias (external input, or offset) $b_j^{(l)}$ of the j^{th} neuron in the l^{th} layer. Thus, $a_j^{(l)}$ is the activation of the j^{th} neuron in the l^{th} layer. An activation function $f^{(l)}$ determines the new level of activation based on the weighted sum of the activations in the previous layer

$$a_j^{(l)} = f^{(l)}\left(z_j^{(l)}\right), \quad (61)$$

where we define $z_j^{(l)} = \sum_k w_{jk}^{(l)} a_k^{(l-1)} + b_j^{(l)}$ as the weighted input. The activation function f transforms the weighted sum of the input into output from the neurons in one layer of the network. f can both be linear and non-linear. For the non-linear case, some type of non-linear threshold function is typically used, for instance a sign function, a hyperbolic tangent function, a sigmoid function, or a Rectified Linear Unit (ReLU) function. Some commonly used activation functions are shown in Figure 32. In this case, the non-linearity of f between the linear operations of the weighted sum in Equation (61) can improve the performance and capability of the network. We can also write Equation (61) in a compact vectorised form using a weight matrix $W^{(l)}$ with elements $w_{jk}^{(l)}$

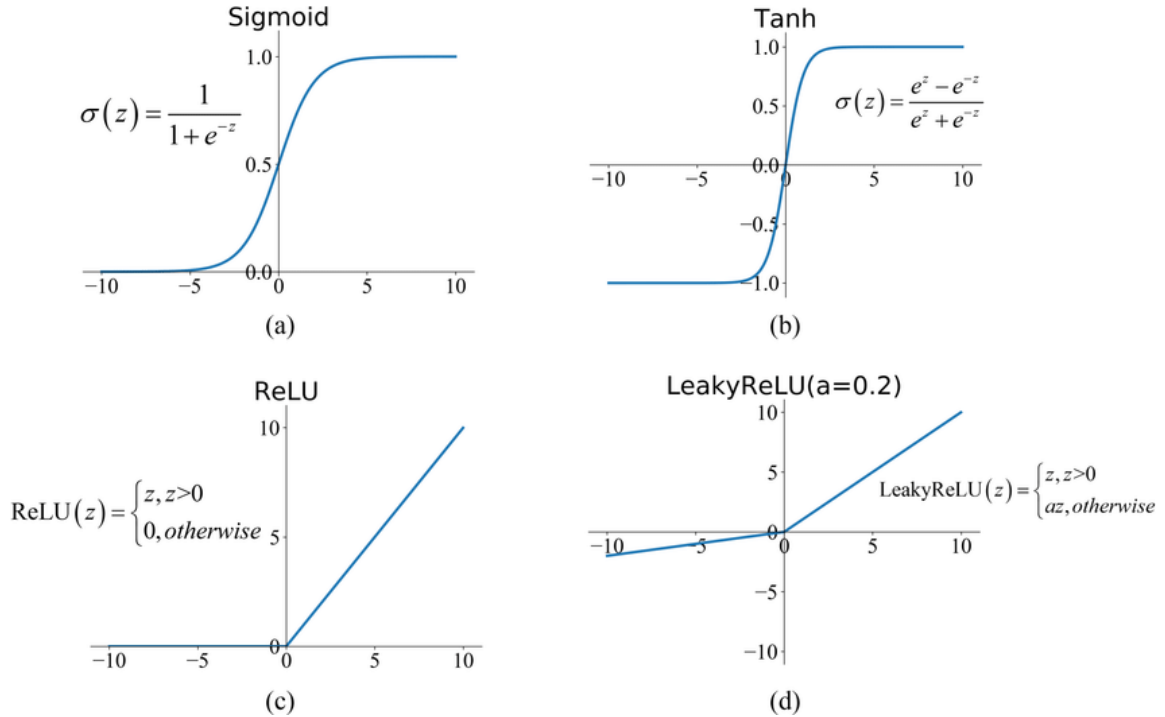


Figure 32: Figure of commonly used activation functions: (a) Sigmoid, (b) Tanh, (c) ReLU, and (d) LReLU. Image reproduced from [39].

$$\mathbf{a}^{(l)} = f^{(l)}\left(W^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}\right) \quad (62)$$

An illustration of a simple neural network is shown in Figure 33. Some input vector $\vec{x}$, normalised between $[0, 1]$, constitute the input layer activations $\mathbf{a}^{(0)}$. The output vector $\vec{y}$ contains the output layer activations $\mathbf{a}^{(l)}$ ($l = 3$ in the example). In this manner, the neural network can be interpreted as a function $\mathcal{M} : \vec{x} \rightarrow \vec{y}$ that maps any normalised input $\vec{x}$ to the normalised output $\vec{y}$ in the range $[0, 1]$. In our case, the input $\vec{x}$ corresponds to the nine MKP delays and the output $\vec{y} = y$ to the loss function $\mathcal{F}$. The map $\mathcal{M}$ can be very large with thousands (or more) of weights for the linear matrix operations, with non-linear activation function operations in between.

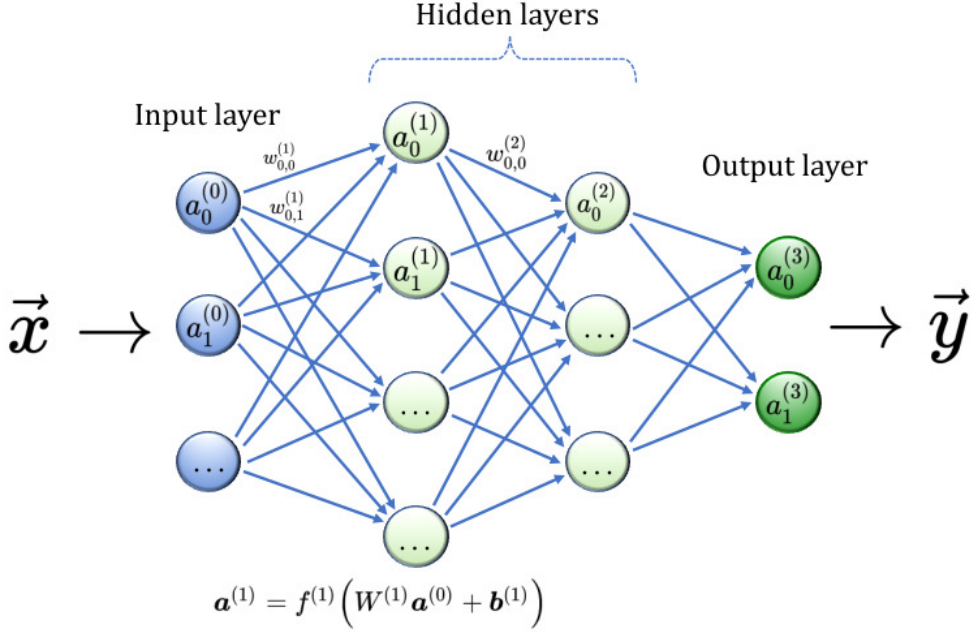


Figure 33: Illustration of a simple neural network.

The next step after constructing the model is to find weights and biases such that the output $\vec{y}_{\text{model}}$ from the network model approximates the real output $\vec{y}_{\text{real}}$ from the accelerator for all training inputs $\vec{x}$. We quantify how well the model approximates real data by defining the mean squared error (MSE) cost function C

$$C(w, b) = \frac{1}{2n} \sum_x \|y(x) - \mathbf{a}^{(l)}\|^2 \quad (63)$$

where w denotes all the weights in the network, b all the biases, n is the total number of training inputs, and $\mathbf{a} = \vec{y}_{\text{model}}$ is the output vector from the model for the given input x . The cost function $C(w, b)$ becomes small when the final output $\mathbf{a}$ of the model approximates $y(x)$ well, where the quadratic difference heavily penalises any offset. To find weights and biases for the model that minimises C , we use an algorithm known as *stochastic gradient descent*, an iterative optimisation algorithm to find a local minimum estimating the gradient $\nabla C = (\partial C / \partial w, \partial C / \partial b)$ (where w and b denote all the weights and biases) of the cost function using randomly selected subsets of the data. In order to find the weights and biases minimising the cost function, we use a method called *backpropagation*. Instead of computing the gradient directly with respect

to all the weights, backpropagation instead computes the gradient of the cost function with respect to each weight by the chain rule, calculating the gradient one layer at a time from backwards iteration. This method is computationally more efficient, as intermediate redundant terms do not need to be calculated [38]. These calculations are performed by keeping track of small perturbations to the weights and biases as they propagate through the network, reach the output, and then affect the final cost C .

In our studies, $\vec{x}$ is the vector that contains the 9 kicker delays. The output $\vec{y} = y$ is a scalar containing the squared sum of the beam positions, from the loss function $\mathcal{F}$ described in Equation (60). However, this is done with a slight modification: we define $\mathcal{G}$ to be the normalised logarithm of the loss function $\mathcal{F}$

$$\mathcal{G}(\bar{x}, \bar{x}_c) = \frac{1}{N} \log \left(\mathcal{F}(\bar{x}, \bar{x}_c) \right) \quad (64)$$

where N is a normalisation constant, and the oscillation amplitudes $\bar{x}, \bar{x}_c$ are defined in Equation (60). We stress that the unit $[\text{mm}^2]$ of $\mathcal{F}$ formally has to be normalised by some constant to enter the logarithm argument. The mapping $\mathcal{G}$ ensures that the actual output values y given to the neural network model lies in the range $[0, 1]$, and where the logarithmic operation greatly reduces the numerical distance to outliers caused by the beam losses. Thus, the output training data y provided to the neural network model is first transformed by $\mathcal{G}$ to improve numerical stability, while still preserving the shape of the search landscape. One could think of the optimisation search space in our problem as a large multidimensional version of “Grand Canyon”, where the continuous part of the objective function $\mathcal{F}$ (when the beam is preserved) constitutes the lower smooth part of the valley in which the algorithm finds the heuristic minimum. On the other hand, the manual penalty given to the optimisation when the beam is lost represents the tall flat plateau on top of the canyon. This steep wall between the continuous low valley and the tall plateau might cause numerical instabilities, and makes it harder for optimisers or neural network models to estimate the function accurately. In other words, $\mathcal{G}$ reduces the height of the numerical “Grand Canyon” effect, flattening the tall multidimensional plateaus caused by the high manual penalty of a lost beam while preserving the shape of the lower continuous valley. The mapping $\mathcal{G}$ thus makes it easier for the neural network to adjust the weights $w_{jk}^{(l)}$.

The architecture of our simple neural network is presented in the code snippet below. We use the extensive and easily-implemented features of the `pytorch` library [40]. The subclass `nn` contains many useful methods to easily define the layers and the number of neurons for the network. In the snippet below, we include four layers, ranging from 200 to 100 neurons, that maps an input vector of 9 normalised MKP delays to one loss function value.

```

1 class NeuralNetwork(nn.Module):
2     def __init__(self):
3         super().__init__()
4         self.layers = torch.nn.Sequential(
5             torch.nn.Linear(9, 200),
6             torch.nn.BatchNorm1d(200),
7             torch.nn.ReLU(),
8             torch.nn.Linear(200, 150),
9             torch.nn.BatchNorm1d(150),
10            torch.nn.ReLU(),
11            torch.nn.Linear(150, 100),

```

```

12     torch.nn.BatchNorm1d(100),
13     torch.nn.ReLU(),
14     torch.nn.Linear(100, 1)
15 )
16 self.apply(self._init_weights)

```

We use the ADAM (Adaptive Moment Estimation) optimiser, one of the most popular algorithms to find the weights and biases of neural networks. To facilitate the optimisation procedure, we perform parameter initialisation by randomising the weights in snippet line 16 from the normal distribution $\mathcal{N}(0.0, 1 \times 10^{-4})$. Without initialising weights - possibly zero-valued by default - the gradients might diverge or vanish entirely. For each layer, we also perform batch normalisation. The distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This process slows down the training by requiring lower learning rates and careful parameter initialisation. This phenomenon, called internal co-variate shift, makes it hard to train models with saturating nonlinearities. Performing batch normalisation while training each mini-batch allows us to use much higher learning rates and be less careful about initialisation [41]. This normalisation is done after each layer with the built-in function of `pytorch`.

To train the model, we divide the recorded data set into 70% training data, 20% validation data and 10% test data. Training is performed during a chosen number of iterations, called *epochs*, in which the algorithm scans through the whole data set. However, we allow the model to update the weights before having scanned through the whole data set. The number of samples the training algorithm goes through before updating parameters is called *batch size*. Stochastic gradient descent is the iterative algorithm used to find the ideal weights and biases to minimise the cost function given in Equation (63). After testing some different hyperparameter values (the parameters that control the learning process), we set the batch size to 64 samples and the learning rate to 1×10^{-4} . The learning rate controls how much the algorithm should change the model in response to the estimated error each time the model weights and biases are updated, where a too low learning rate may cause the learning process to get stuck, whereas a too high learning rate can lead to unstable learning processes [42]. We program the learning rate to decrease by half after 200 epochs and to yet again decrease by half after 300 epochs, to fine-tune the learning process. After each epoch, the updated model is tested against Equation (63) with the yet unseen validation data set to tune the parameters of the networks. We stop the training when the loss function for the validation flattens and the training losses keep on decreasing, which means that the learning algorithm overfits the model parameters for the training dataset. The training and validation loss curves are displayed to the left in Figure 34.

We stop the training process at 200 epochs (green dashed line), after which the learning process starts overfitting. As a last step, we test the model with a final unbiased evaluation against unseen test data, and make a regression plot of predicted values $y_{\text{predicted}}$ versus the actual test set values y_{actual} , displayed to the right in Figure 34. To test the validity of the linear regression of the trained model, we use the statistical tool goodness-of-fit R^2 [43]. R^2 is the fraction of how much the fitted model can explain the variation around the average $\bar{y}$ of the observed data, defined as

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i. \quad (65)$$

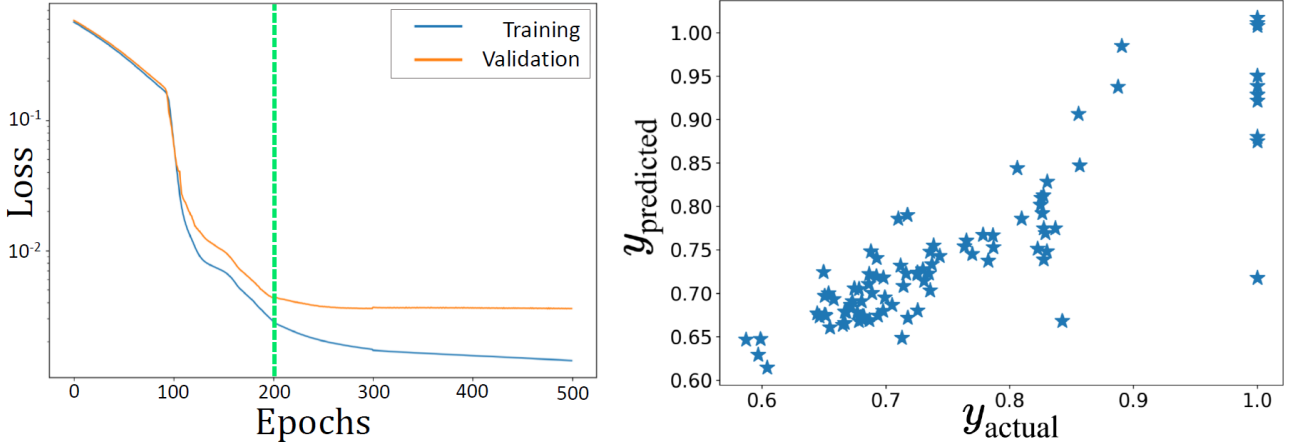


Figure 34: Left: learning curve of the neural network on the training dataset and on the validation dataset. We stop training the model after 200 epochs (dashed green line), after which the model becomes overtrained on the training dataset and starts diverging significantly from the validation. Right: prediction curve of the NN regression model on test dataset. $y_{\text{predicted}}$ given by the model is compared to the actual value y_{actual} , with goodness-of-fit $R^2 = 0.76$.

The goodness-of-fit R^2 , also called coefficient of determination, is defined as

$$R^2 = 1 - \frac{\text{SSR}}{\text{SST}}, \quad (66)$$

where SST is the total sum of squares

$$\text{SST} = \sum_i (y_i - \bar{y})^2, \quad (67)$$

and SSR is the sum of squared residuals

$$\text{SSR} = \sum_i (y_i - y_{\text{predicted}})^2. \quad (68)$$

Although R^2 is not a complete statistical tool by itself and does not tell e.g. whether there are enough data points for valid conclusions, it does give an indication of the goodness of fit for linear regression. For instance, $R^2 = 1$ means that the model explains all the variability of the response data around its mean, and $R^2 = 0$ means that the model explains none of it. To the right in Figure 34, $R^2 = 0.76$ for a model trained for 200 epochs. Instead if we only train the model for 150 epochs, $R^2 = 0.34$. Once the neural network model of the accelerator has been trained, validated and tested, we want to implement other optimisation algorithms on the model to compare performances. In this manner, we test out different algorithms on a trained model that imitates the response of the SPS injection system. As an example, the convergence plot of the BOBYQA algorithm tested on the neural network model is shown in Figure 35. The performance plots for the other algorithms (Nelder-Mead, Powell and Bayesian) are found in the Appendix.

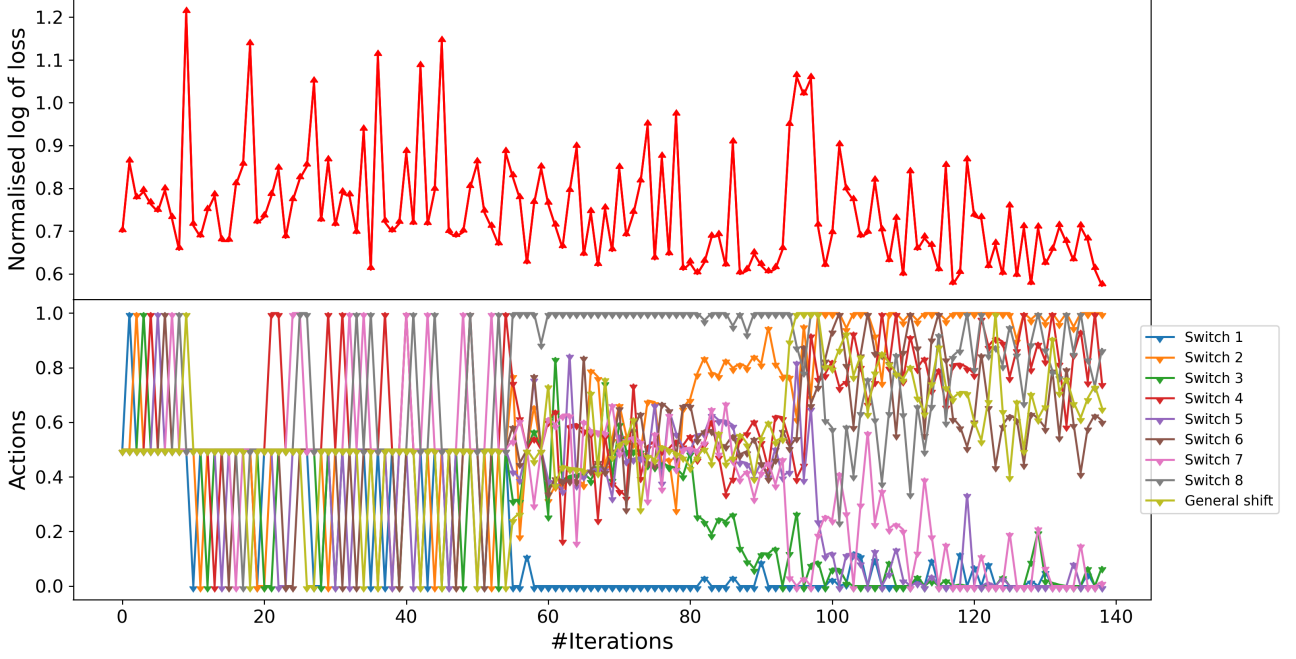


Figure 35: Convergence plot of the BOBYQA optimisation algorithm with the neural network model, with normalised logarithm of the losses $\mathcal{G}$ in the upper pane, and the normalised actions of the different switches in the lower pane. The algorithm explores the maximum limits of each action for the first 55 iterations to create a trust region, before minimising the objective function.

7.4 Optimisation algorithm selection using the surrogate model

We want the algorithm that runs faster, uses less memory, and returns a better final loss function value from Equation (60) - smallest injection error possible - in the lowest number of function evaluations possible, as time in the CCC is limited. In order to compare the performance of different optimisation algorithms, we use a visual tool called *performance profile*. Performance profiles have become a gold-standard in modern optimisation benchmarking (formalised comparison of methods using different metrics), and should be included in optimisation benchmarking analysis whenever possible with an appropriate interpretation [44]. A performance plots visualises the probability that an algorithm solves a problem within a given time frame or function evaluation budget k . Figure 36 shows the cumulative distribution of the optimisation results of various algorithms for 100 different normalised initial conditions $\vec{X}_0$ for the kicker delays, sampled from the normal distribution $\mathcal{N}(0.5, 0.1)$. For 100 different initial conditions, the y -axis shows the fraction that takes a loss function value lower or equal to $\mathcal{G}$ from Equation (64) on the x -axis. Thus, we want an optimisation algorithm where as many of the different initial conditions result in the lowest possible loss function value - a steep curve as much to the left on the x -axis as possible. We compare the performance of different algorithms for different function evaluation budgets k . We observed from the online optimisation described in Section 7.2 that 100 function evaluations of Equation (60) require about an afternoon of beam operation, where each evaluation of a set of MKP kicker delays affect the beam position. Depending on the beam time available in the CCC, different function evaluation budgets k have to be considered.

In Figure 36, we compare the BOBYQA, Nelder-Mead, Powell, and Bayesian algorithms. For the Bayesian optimisation, we use the Python library `bayesian-optimization` [45]. Bayesian optimisation constructs a posterior distribution of the desired functions $f(x)$ with a Gaussian process that best describes f . As we increase the number of observations, the posterior distri-

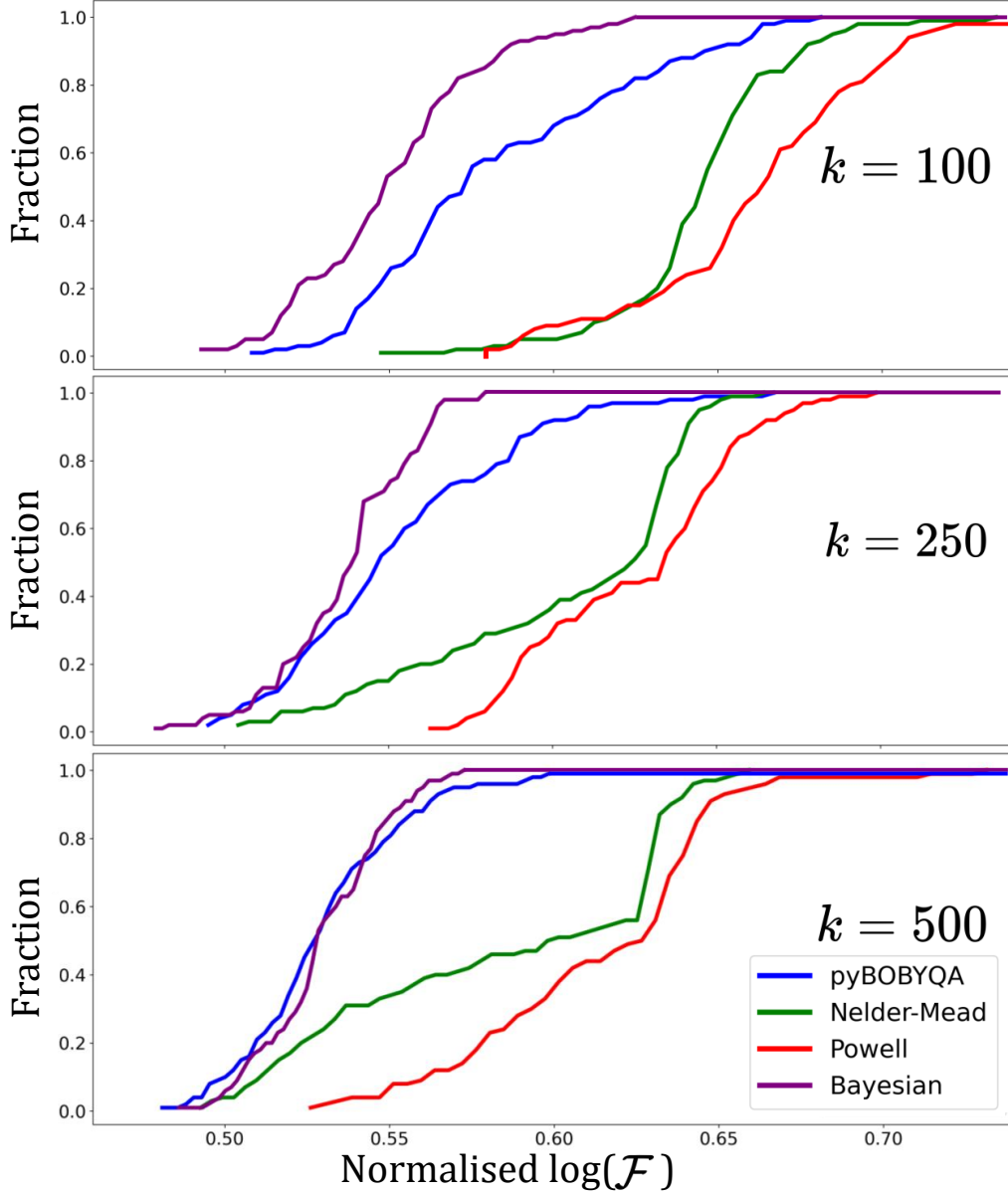


Figure 36: Performance plot of different optimisation algorithms for 100 different initial conditions $\vec{X}_0 \leftarrow \mathcal{N}(0.5, 0.1)$ sampled from a normal distribution, with budget k of maximum function evaluations $k = 100$ (top), $k = 250$ (middle) and $k = 500$ (bottom).

bution improves. The algorithm becomes more certain of which regions in parameter space that are worth exploring and which are not, and where to find the optimum. The Bayesian process is designed to minimise the number of steps required to find a combination of parameters that are close to the optimal combination, keeping the number of function evaluations low. On the other hand, every function evaluation with the Bayesian optimiser is computationally more expensive than the other algorithms. However, this does not necessarily impose a problem in the CCC where beam is delivered every few minutes, and optimisation function evaluations are scarce. For a budget of $k = 100$ function evaluations, we observe that the Bayesian optimiser reaches the lowest absolute value, but its early steep curve also indicates that a larger fraction of its function evaluations finds lower loss function values. Not far behind lies BOBYQA with a similar trend - Nelder-Mead and Powell do not perform as well for $k = 100$, with higher loss function values and larger fractions of iterations located at higher values. As we increase

the budget k to 250 and 500 function evaluations, we observe that the Bayesian curve shape remains almost identical, whereas BOBYQA improves towards lower values and even surpasses Bayesian at $k = 500$. Powell does not improve much as we increase k , whereas Nelder-Mead reaches lower and lower loss function values with a higher fraction at lower values, albeit not performing as well as BOBYQA or the Bayesian. Thus, the Bayesian algorithm seems the most promising for budgets up to about $k = 250$ function evaluations, whereas BOBYQA slightly overtakes the performance for higher budgets. Although the neural network is an approximate model of the accelerator, this optimisation algorithm comparison gives a clear indication of what options to consider for different situations and time limits. Figure 36 clarifies how different algorithms compare for a given function evaluation budget k , but how can we quantify how well the performance of an algorithm improves as we increase k ?

As an additional instrument, data profiles $d(k)$ illustrate the percentage of problems - for a given tolerance α_a - that can be solved within a budget of k function evaluations [44]. A data profile is a cumulative distribution, just like Figure 36, but showing the fraction that succeeds in finding a loss function value lower than the tolerance α_a , as a function of k . The choice of α_a - the loss function value when the problem is considered “solved” - is arbitrary and individual for each algorithm a . As a benchmark, we select α_a for each algorithm as the loss function value that at least 50% of the optimisation iterations reach (or the data point closest to 50% in Figure 36) for $k = 100$. The data profiles in Figure 37 show the fraction of “solved” problems - i.e. fraction of optimisation iterations finding a value lower or equal to α_a - as we increase the budget k .

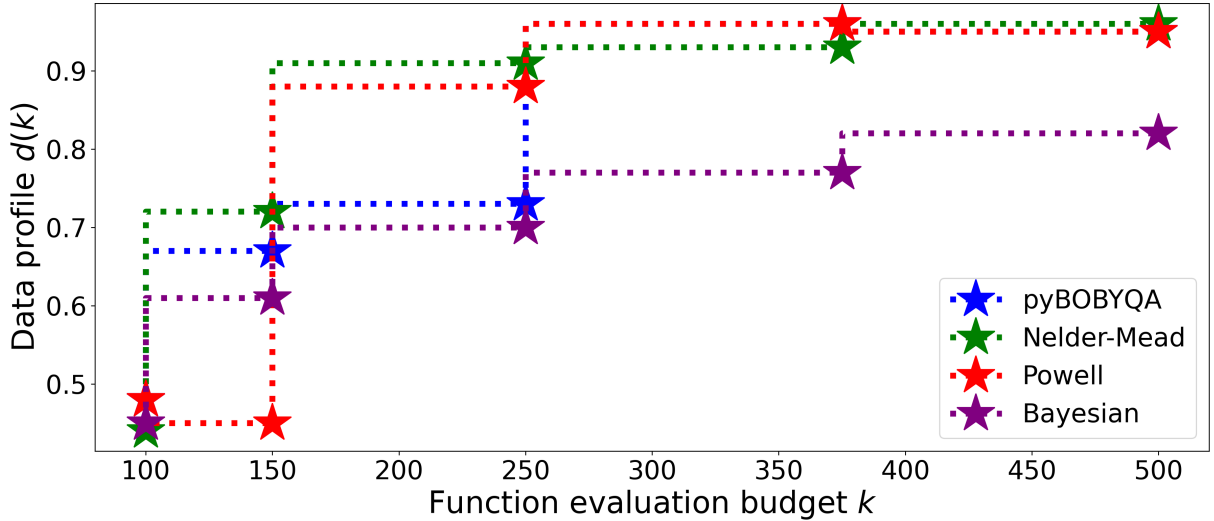


Figure 37: Data profile $d(k)$ plot with fraction of successfully solved optimisation given a tolerance α_a (unique to each algorithm a) within a function evaluation budget of k iterations. This plot was generated from the performance profiles in Figure 36.

We observe that for Nelder-Mead and BOBYQA, increasing k has a larger effect on the data profile $d(k)$, corresponding to the problem solving rate. Also Powell, albeit a smaller drop at $k = 150$, sees its data profile improve as k exceeds 250. However, the improving effect for the data profile of the Bayesian algorithm is smaller, meaning that increasing k has a smaller effect on the Bayesian optimisation. We once again underline that data profiles are not a complete tool and cannot be used to find the absolute loss function minimum or to compare algorithms in absolute terms. The data profiles rather constitute a complimentary visual instrument to the performance profiles in Figure 36 to compare each individual algorithm with itself and what

effect a higher budget k has. Figure 36 and 37 are valuable indications for operations in the CCC, as they provide guidance to what optimisation algorithm to choose depending on the beam time available.

8 Conclusions

The current SPS injection system has seen a radical evolution since its first design, and the characteristics of the injected beam have changed significantly. A full redesign may be needed to solve long-standing issues, like the different kicker types and the bulky septa. In this study, we first performed theoretical studies on emittance growth due to amplitude-dependent injection errors for a generic circular accelerator. In Section 4, we derived evolution equations for the first and second moments of an coupled arbitrary Gaussian phase-space distribution that initially is mismatched, displaced, and has mismatched dispersion under the influence of decoherence due to amplitude-dependent tune shift. The established results from [19] and [46] for the amplitude dependence of the first and second moments after an initial displacement of a matched beam were reproduced. Our results in this study go beyond [19] and [46], because the initial beam can have an arbitrary Gaussian distribution, which includes transverse coupling, and does not need to be matched. We then calculated the temporal evolution of the second moments, the beam sizes, and the emittance. Moreover, we calculated the emittance in the asymptotic limit and found it to agree with the emittance growth due to chromatic effects. In Section 5, we analysed tolerances for the injection and used the SPS parameters as an illustration. This work was published in [2].

In Section 6, we presented a hierarchical optimisation framework that gradually constructs a new injection sequence and generates ideal candidate solutions for all SPS optics - finding the ideal balance between minimal magnet resources and dumped beam trajectory, while considering realistic hardware. The developed framework is rather generic and provides a basis for designing other accelerator systems, even with higher complexity. We stress that although the SPS injection is a specific case, the high-level discrete design of magnet modules and configurations can in principle be extended to almost any accelerator sequence, as described in a pseudo-algorithm in [7].

In Section 7, we presented an optimiser to find the ideal individual and general timing of the MKP kicker delays. We first performed pre-study simulations in Section 7.1 to observe the effects on optimising the timing to reduce the injection errors, followed by the implementation of a real-time optimisation environment on the actual SPS injection kicker magnets, described in Section 7.2. The impact on the residual injection oscillations was significant both on the circulating and injected beam, reducing the injection oscillation amplitude by up to 70%. We then developed a simple neural network and a surrogate model of the injection system to compare different optimisation algorithms in Section 7.3 and 7.4, demonstrating what algorithms would be suitable depending on the beam time available in the CCC. The neural network model approximated the real accelerator surprisingly well, which allows for testing and fine-tuning offline also when accelerator operation time is not available. For future studies, it would be interesting to extend the simple neural network model and compare between new architectures. A potential upgrade is to construct a dual neural network model that first classifies whether the beam was lost or not, and then performs regression of the ideal kicker delays. This would eliminate the “Grand Canyon effect” of penalising the lost beam, although we evaded this problem successfully with the transformation $\mathcal{G}$ from Equation (64). Another aspect to investigate

is whether we could reduce the number of actors from nine time delays to eight for smoother optimisation, as each individual delay is coupled to the general delay. However, an operator in the CCC still needs to provide one general and eight individual kicker delays to the machine, for which our optimiser served its purpose.

These studies show the importance and impact of optimising the present SPS injection system, implementing relatively recent tools such as numerical optimisation and neural networks, while exploiting these resources to design a future injection system - ultimately increasing beam intensity and luminosity beneficial to many experiments at CERN.

9 Acknowledgements

Coming to CERN and taking part in its exciting fundamental research with its international community has been my dream since childhood, but never could I have envisioned how much I would be able to learn here. As a technical student in applied physics, one is constantly exposed to the challenge and joy of undergoing a long steep learning curve. Less than nine months have passed since I started, yet it feels like my notion of what physics is all about and where it can have an impact has radically changed with respect to before. The phases of the project have been numerous: from the beam dynamics simulations of complex non-linear system to the actually testing out my optimisation software in the CERN Control Room and observing its improving effect on the SPS beam stability on the monitor, from publishing theoretical studies of the beam emittance growth to learning to implement methods such as neural networks and genetic algorithms, that previously were merely technological buzzwords to me. I want to thank everyone who has been involved and taught me so much, empowering me and inspiring to pursue the long-standing dream that I am about to embark upon: pursuing a PhD at CERN.

First of all, I want to thank my CERN supervisor Francesco Velotti for guiding me through the learning process, for his encouragement and all the responsibility to allow me to test out my optimisers in the CCC and observe the effects. Thanks to you, I know a bit more what it means to take part in operations, what it means being a researcher in accelerator physics, but also what exciting tools and methods there are out there. I want to express gratitude to the members of ABT section for their hospitality, teaching me all the tricks not found in the manuals and posing new research questions that keep on challenging the intellect. I also want to thank my local supervisor Volker Ziemann, who first led me onto the track of accelerator physics throughout many courses and projects, but with whom I published my first article. From many points of view, you have become a mentor for me. I express gratitude to Roger Ruber, Maja Olvegård and all the members of the FREIA Laboratory for their support. Last but not least, I want to give my sincere thanks to my family members from the coastal forests of northern Sweden down to the Italian Alps, who have been solid pillars of support through the most turbulent times. I remain speechless of gratitude and wonder towards my co-sailor and soulmate Sveva Castello, with whom I have already been able to build and experience more than one could imagine. Our home is where we are together, overlooking the splendid beauty of Alps and boreal forests...

References

- [1] P. Collier, “SPS as injector for LHC: Conceptual design report,” Tech. Rep. CERN-SL-97-07 DI, CERN, 1997.
- [2] E. Waagaard and V. Ziemann, “Emittance growth of kicked and mismatched beams due to amplitude-dependent tune shift,” *Phys. Rev. Accel. Beams*, vol. 25, p. 054001, May 2022. Available at <https://link.aps.org/doi/10.1103/PhysRevAccelBeams.25.054001>.
- [3] G. Apollinari, L. Rossi, and O. Brüning, “High luminosity LHC project description,” 2014. Tech Report No. CERN-ACC-2014-0321.
- [4] A. Scheinker, S. Hirlander, F. M. Velotti, S. Gessner, G. Z. Della Porta, V. Kain, B. Goddard, and R. Ramjiawan, “Online multi-objective particle accelerator optimization of the AWAKE electron beam line for simultaneous emittance and orbit control,” *AIP Advances*, vol. 10, no. 5, p. 055320, 2020.
- [5] V. Kain, S. Hirlander, B. Goddard, F. M. Velotti, G. Z. Della Porta, N. Bruchon, and G. Valentino, “Sample-efficient reinforcement learning for CERN accelerator control,” *Physical Review Accelerators and Beams*, vol. 23, no. 12, p. 124801, 2020.
- [6] Y. Dutheil, F. Velotti, and B. Goddard, “The blind linemaker: Genetic algorithms for multi-objective optimisation.” Available at <https://indico.cern.ch/event/964060/> [Accessed 18/01/2022].
- [7] E. Waagaard, “Designing a new sps injection system with numerical optimisation.” Uppsala University Publications. Available at: <http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-468033>.
- [8] E. Shaposhnikova, S. Mataguez, D. Manglunki, A. Funken, J. Coupard, M. Meddahi, M. Vretenar, H. Damerau, B. Goddard, A. Lombardi, *et al.*, “LHC Injectors Upgrade (LIU) Project at CERN,” 2016. Paper ID MOPOY059: Proceedings of IPAC 2016, Busan, Korea.
- [9] A. Wolski, *Beam dynamics in high energy particle accelerators*. World Scientific, 2014.
- [10] S.-Y. Lee, *Accelerator physics*. World Scientific Publishing Company, 2018.
- [11] R. Tomás García, *Direct measurement of resonance driving terms in the super proton synchrotron (SPS) of CERN using beam position monitors*. PhD thesis, Universitat de València, 2003.
- [12] Y. Papaphilippou, “Transverse motion.” United States Particle Accelerator School, University of California - Santa-Cruz (2008). Available at: <https://yannis.web.cern.ch/teaching/transverse.pdf> [Accessed 19/10/2021].
- [13] V. Ziemann, *Hands-On Accelerator Physics Using MATLAB®*. CRC Press, 2019.
- [14] F. M. Velotti, *Higher brightness beams from the SPS for the HL-LHC era*. PhD thesis, EPFL, 2017.

- [15] W. Bartmann, “Transfer lines - paper studies.” Shool Proceedings of the CERN Accelerator School (CAS) 2017. Available at: https://indico.cern.ch/event/451905/contributions/2159098/attachments/1425372/2194111/CAS_transferlines_1.pdf [Accessed 30/09/2021].
- [16] B. Goddard, “Injection and extraction.” Introduction to Accelerator Physics 2009 Course, Divonne, France. Available at: <https://indico.cern.ch/event/43703/contributions/1086895/attachments/939252/1331885/Goddard.pdf.pdf> [Accessed 10/01/2022].
- [17] C. Bracco, “Injection: Hadron beams.” Shool Proceedings of the CERN Accelerator School (CAS) 2017. Available at: https://indico.cern.ch/event/451905/contributions/2159032/attachments/1425216/2188137/CAS_ERICE_Hadrons_Injection.pdf [Accessed 10/01/2022].
- [18] V. Kain, “Emittance preservation.” Shool Proceedings of the CERN Accelerator School (CAS) 2017. Available at: https://indico.cern.ch/event/451905/contributions/2159070/attachments/1423256/2191040/EmittancePreservation_v1.pdf [Accessed 23/05/2022].
- [19] R. Meller, A. Chao, J. Peterson, S. G. Peggs, and M. Furman, “Decoherence of kicked beams,” *SSCN-360*, 1987.
- [20] S. Lee, “Decoherence of kicked beams II,” *Internal Rep. SSCL-N-749*, 1991.
- [21] A. Sargsyan, “Transverse decoherence of the kicked beams due to amplitude and chromaticity tune shifts,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 638, no. 1, pp. 15–18, 2011.
- [22] CERN, “The 300 GeV Programme,” 1972. Tech Report No. CERN/1050.
- [23] “UA9 experiment official website.” Available at <https://ua9.web.cern.ch/> [Accessed 08/06/2022].
- [24] R. Billinge, “1984 Nobel Prize in Physics,” *Europhysics News*, vol. 15, no. 11-12, pp. 1–2, 1984.
- [25] M. Benedikt, *Optical design of a synchrotron with optimisation of the slow extraction for hadron therapy*. PhD thesis, Vienna, Tech. U., 1997.
- [26] J. Uythoven, “The new SPS injection channel,” tech. rep., 1999. No. CERN-OPEN-99-078.
- [27] M. Barnes, J. Borburgh, B. Goddard, and M. Hourican, “Injection and extraction magnets: septa,” *arXiv preprint arXiv:1103.1062*, 2011.
- [28] C. Zannini, *Electromagnetic simulation of CERN accelerator components and experimental applications*. PhD thesis, EPFL, 2013.
- [29] G. Arduini, M. Giovannozzi, K. Hanke, D. Manglunki, M. Martini, and G. Métral, “Measurement and optimisation of the PS-SPS transfer line optics,” in *Proceedings of the 1999 Particle Accelerator Conference (Cat. No. 99CH36366)*, vol. 2, pp. 1282–1284, IEEE, 1999.

- [30] “CERN Optics Repository, Q20 optics function for SPS injection.” available at https://acc-models.web.cern.ch/acc-models/tls/2021/sps_injection/tt2tt10_lhc_q20/stitched/ [retrieved 2022/03/15].
- [31] H. Grote, F. Schmidt, L. Deniau, and G. Roy, “The MAD-X program (methodical accelerator design) version 5.07.00: User’s reference manual.” Available at: <http://madx.web.cern.ch/madx/> [Accessed 15/12/2021].
- [32] S. Ishikawa, K. Horio, and R. Kubota, “Effective hierarchical optimization using integration of solution spaces and its application to multiple vehicle routing problem,” in *2015 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 406–411, 2015.
- [33] J. Blank and K. Deb, “pymoo: Multi-objective optimization in python,” *IEEE Access*, vol. 8, pp. 89497–89509, 2020.
- [34] “CERNML-COI Gitlab Repository.” Available at <https://gitlab.cern.ch/be-op-ml-optimization/cernml-coi/-/tree/master/> [Accessed 25/05/2022].
- [35] “PyJAPC Official Website.” Available at <https://acc-py.web.cern.ch/gitlab/scripting-tools/pyjapc/docs/stable/> [Accessed 25/05/2022].
- [36] M. J. Powell, “The BOBYQA algorithm for bound constrained optimization without derivatives,” *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, vol. 26, 2009.
- [37] B. Krose and P. v. d. Smagt, *An introduction to neural networks*. Faculty of Mathematics & Computer Science, University of Amsterdam, 2011.
- [38] M. A. Nielsen, *Neural networks and deep learning*, vol. 25. Determination press San Francisco, CA, USA, 2015.
- [39] J. Feng, X. He, Q. Teng, C. Ren, H. Chen, and Y. Li, “Reconstruction of porous media from extremely limited information using conditional generative adversarial networks,” *Physical Review E*, vol. 100, no. 3, p. 033308, 2019.
- [40] “PyTorch Official Website.” Available at <https://pytorch.org/> [Accessed 27/05/2022].
- [41] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, pp. 448–456, PMLR, 2015.
- [42] J. Brownlee, *Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras*. Machine Learning Mastery, 2016.
- [43] V. Ziemann, *Physics and Finance*. Springer, 2021.
- [44] V. Beiranvand, W. Hare, and Y. Lucet, “Best practices for comparing optimization algorithms,” *Optimization and Engineering*, vol. 18, no. 4, pp. 815–848, 2017.
- [45] “Bayesian-optimization library GitHub repository.” Available at <https://github.com/fmfn/BayesianOptimization> [Accessed 31/05/2022].
- [46] M. G. Minty, A. Chao, and W. Spence, “Emittance growth due to decoherence and wakefields,” in *Proceedings Particle Accelerator Conference*, vol. 5, pp. 3037–3039, IEEE, 1995.

A Appendix

Figure 38, 39 and 40 show the convergence plots of different optimisation methods used on the neural network model presented in Section 7.3.

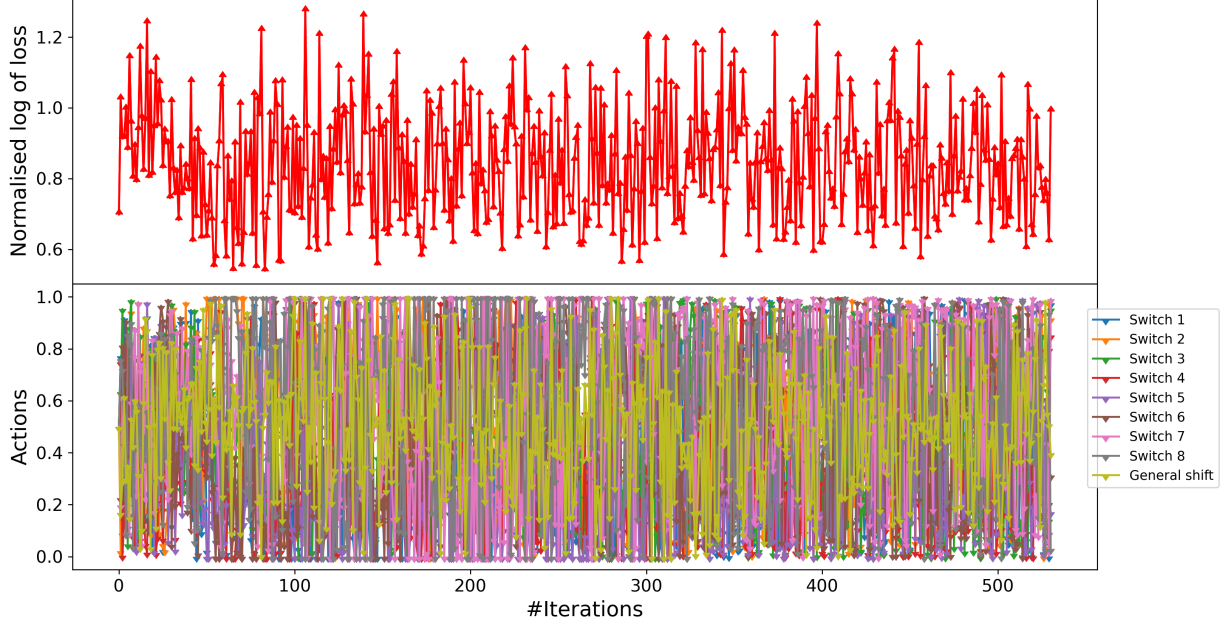


Figure 38: Convergence plot of the Bayesian optimisation algorithm with the neural network model, with normalised logarithm of the losses $\mathcal{G}$ in the upper pane, and the normalised actions of the different switches in the lower pane. One can observe how the loss function decreases until the first 80-100 iterations but not further, in agreement with the performance plots in Figure 36.

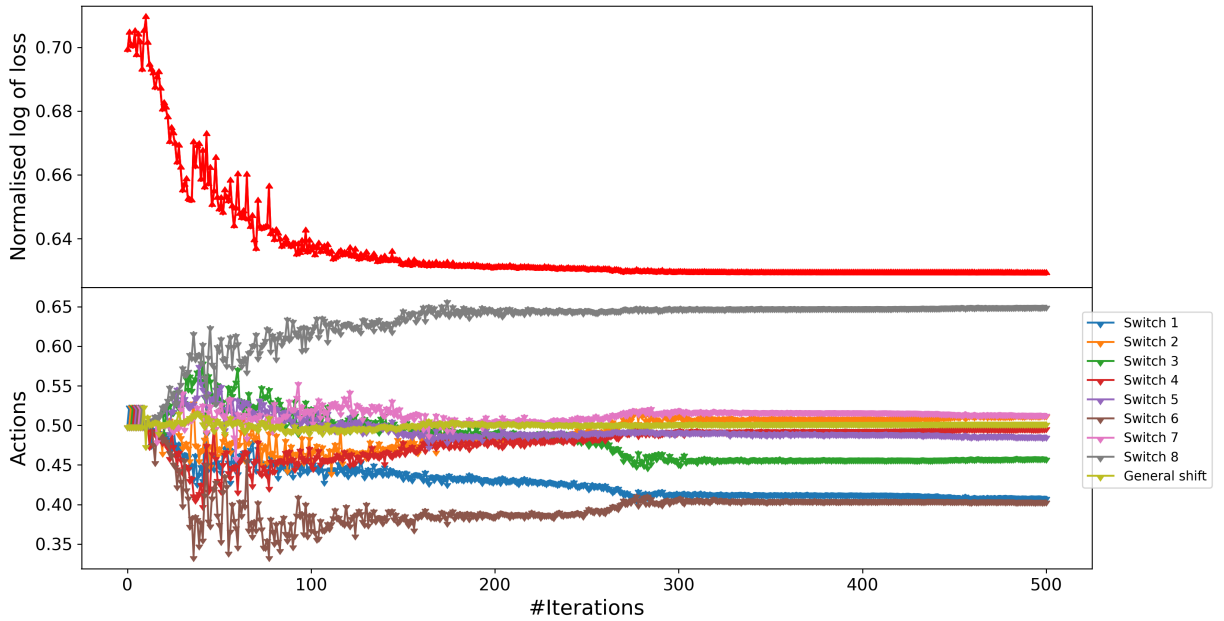


Figure 39: Convergence plot of the Nelder-Mead optimisation algorithm with the neural network model, with normalised logarithm of the losses $\mathcal{G}$ in the upper pane, and the normalised actions of the different switches in the lower pane.

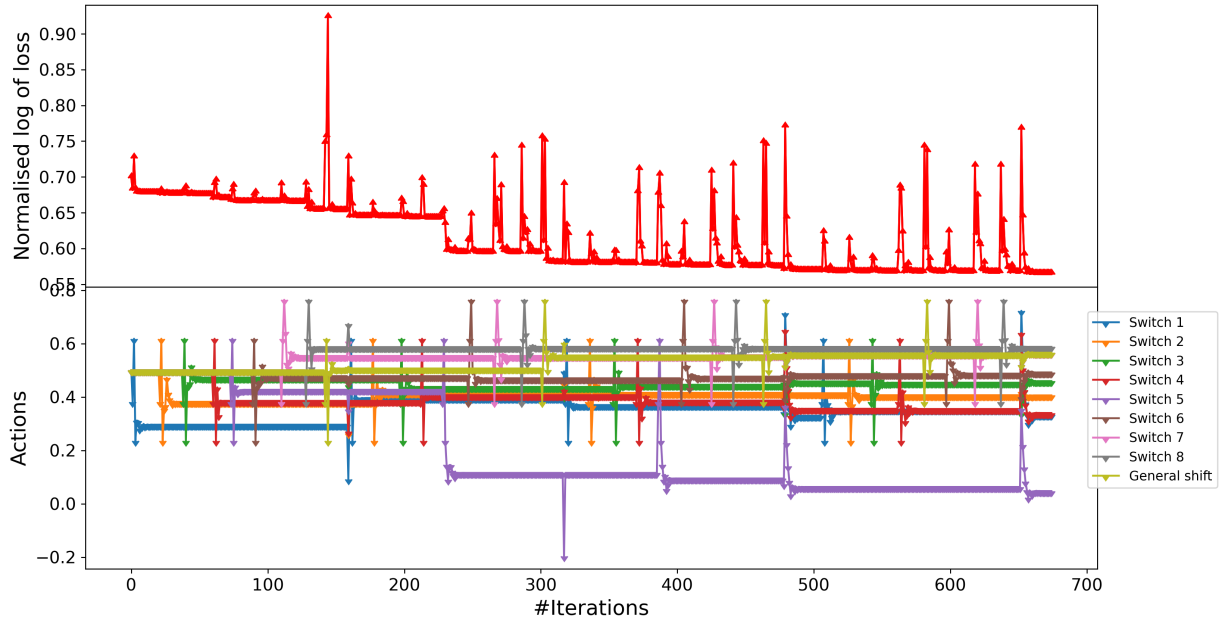


Figure 40: Convergence plot of the Powell optimisation algorithm with the neural network model, with normalised logarithm of the losses $\mathcal{G}$ in the upper pane, and the normalised actions of the different switches in the lower pane. The frequent “spikes” are probably due to the relatively large step sizes Powell take in its actions - a large step size that would lose the beam by changing the delays too much. Comparing Powell with the other algorithms in Figure 36, we observe that it performs the worst in terms of achieving the lowest loss value with respect to the other algorithms.