Helmholz-Zentrum Berlin für Materialien und Energie
Light source BESSY II
Department Operation Accelerator (NP-ABS)

Technische Universität Berlin
Fakultät IV – Elektrotechnik und Informatik
Institut für Energie- und Automatisierungstechnik
Fachgebiet Regelungssysteme
Prof. Dr.-Ing. Jörg Raisch

Master Thesis

Localization and correction of orbit perturbations in BESSY II storage ring

Submitted by

Olivier Churlaud – Matriculation number: 366 964

In partial fulfillment of the requirements for
the Degree of Master of Science in Electrical Engineering

August 8, 2016

Supervisors

TU Berlin	Prof. Dr.-Ing. Jörg Raisch
Helmholtz-Zentrum Berlin	Prof. Dr. Andreas Jankowiak Dr. Andreas Schälicke

Disclaimer

I hereby declare that this dissertation is my own original work and has not been submitted before to any institution for assessment purposes. Further, I have acknowledged all sources used and have cited these in the reference section.

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin – August 8, 2016 Olivier CHURLAUD

Abstract

BESSY II is a 240 m electron storage ring in Berlin, Germany, aimed at producing stable and high-energy light rays by synchrotron radiation. This light is used by various scientific projects and must therefore provide a very stable and reliable light. This means that the electrons must be well focused in the center of their vacuum-chamber. Thus, one of the key requirements is to be able to control and remove perturbations from the electron beam.

The electron beam is described as an orbit, characterized by these parameters: the beta function β which defines the size of the beam, the tune Q which is the number of oscillations per revolution done by the beam around its ideal path and the phase Ψ of the orbit, which represents the magnet's influence on the orbit.

From these parameters, it is possible to derive several correction methods in conjunction with the activation of corrector magnets around the orbit. In this work, the inverse problem was solved thanks to the pseudo-inversion of the response matrix representing the changes of the orbit when a corrector magnet is activated. This, used with a PID correction, already provided an acceptable stable orbit. Additionally, to correct some specific harmonic perturbation, a finite impulse response (FIR) filter that generates the opposite harmonic was implemented, removing most of the targeted perturbation. A simulation environment was also designed to help improve the current correction.

Finally, algorithms were implemented to localize harmonic and non-harmonic static sources of the perturbations, at a precision of 2 to 4 m in the case of a vertical perturbation, and 10 m in the horizontal case. This makes it possible to manually isolate or remove the source, instead of correcting its effects.

Zusammenfassung

BESSY II ist ein Elektronenspeicherring mit einem umfang von 240 m in Berlin (DE), der sehr stabile und hoch energetische Lichtstrahlen durch Synchrotronstrahlung produziert. Dieses Licht wird von mehreren verschiedenen Forschungsprojekten benutzt und muss deshalb hochstabil und betriebssicher sein. Das bedeutet, dass die Elektronen innerhalb des Speicherrings auf einer definierten Bahn gehalten werden müssen. Eine der Hauptanforderungen ist deshalb, dass etwaige Störungen des Elektronenstrahls sowohl kontrollierbar, als auch behebbar sein müssen.

Der Elektronenstrahl wird durch einen sogenannten Orbit beschrieben, der durch die folgenden Parameter gekennzeichnet ist: die Betafunktion β definiert die Strahlgröße, der Tune Q ist die Anzahl von Schwingungen, die der Strahl entlang seines idealen Wegs in einem Umlauf durchläuft und die Phase Ψ stellt den Einfluss der Magnete auf den Orbit dar.

Verschiedene Korrekturmethoden können durch die jeweilige Aktivierung von Korrektormagneten entlang der Umlaufbahn verwendet werden. In dieser Arbeit wurde das inverse Problem durch die Pseudo-Inversion derjenigen Reaktionsmatrix gelöst, die Veränderungen der Umlaufbahn beschreibt, wenn Korrektormagneten aktiviert werden. Die Implementierung dieser Technik inklusive eines PID Reglers garantiert eine akzeptable und stabile Umlaufbahn. Zusätzlich wurden spezifische harmonische Störungen durch einen finite impulse response Filter (FIR) beseitigt.

Um die aktuelle Korrektur zu verbessern, wurde der Speicherring modelliert und eine Simulation entworfen.

Anschließend wurden Algorithmen implementiert, um harmonische und nicht harmonische statische Störquellen zu lokalisieren. Die Genauigkeit der Lokalisierung erreicht im Falle einer vertikalen circa 4 m und im Falle einer horizontalen Störung circa 10 m. Dadurch können diese Störquellen manuell isoliert oder entfernt werden, anstatt nur ihre Wirkungen zu korrigieren.

Acknowledgments

I would like to thank my supervisor Andreas Schälicke who has always been available to discuss the progress of my work, and managed to come on weekends and very early Monday mornings to attend the tests on the storage ring; Dennis Engel who has always had time to take me to the machines and explain to me in detail the history of such and such choices technology or process, and who took part in most implementation design deliberation. Both of them followed very closely the progress of the thesis, helping me understand the complex historical software and hardware stack involved in the correction, debug my programs, etc. They were also very patient with my German, which is still far from being perfect, but which I nevertheless really improved in this time.

I furthermore would like to thank Professor Raisch and Professor Jankowiak for providing me with this great opportunity and experience.

I would also like also to express my gratitude to the operators who were always kind enough to refill the storage ring when I defocused and threw the electron beam into the wall.

Lastly, I must warmly thank my friend Maxim who gave me some worthy resources in control theory, and some breaks when some algorithms refused to obey me. I finally cannot forget all the ones who read the drafts of my thesis, correcting my English and asking for more detailed explanations when I was too obscure.

Contents

List of Figures

Notations

The following notations are used:

- vector: $\underline{X}$, $\underline{x}$
- matrix: $\underline{\underline{A}}$
- matrix component, ith line, jth column: A_{ij} (same case)
- ith vector component: X_i, x_i (same case)
- Function evaluation:
  - continuous: x(t)
  - discrete: x[k]
- Operations:
  - multiplication: $x \cdot y$, $\underline{\underline{A}} \cdot \underline{x}$, xy, $\underline{\underline{A}}\,\underline{x}$,
  - vector product: $\underline{x} \times \underline{y}$
  - scalar/inner product: $\langle \underline{x}\,, \underline{y} \rangle$
  - point-wise multiplication: $\underline{x} \odot \underline{y}$
  - convolution: $x * y\,(t)$
  - pseudo-inverse: $\underline{\underline{A}}^*$
- Derivation:
  - over time: $\dot{x} = \frac{dx}{dt}$, $\ddot{x} = \frac{d^2x}{dt^2}$
  - over space variable: $x' = \frac{dx}{ds}$, $x'' = \frac{d^2x}{ds^2}$
- Control theory:
  - Fourier transform: $X(f) = \mathcal{F}\{x(t)\}$
  - Laplace transform: $X(s) = \mathcal{L}\{x(t)\}$
  - Z-transform: $X(z) = \mathcal{Z}\{x(t)\}$

Chapter 1

Introduction

1.1 BESSY II – General presentation

BESSY II (**B**erliner **E**lektronen **S**peicherring-Gesellschaft für **Sy**nchrotronstrahlung m.b.H.) is Berlin's electron storage ring, aimed at producing high energy light rays by synchrotron radiation. It emits extremely brilliant photons pulses ranging from Terahertz to hard X rays, with an emphasis on the soft X-ray range [2].

Scientific projects can freely apply for beam time at an experimental station, where they are able to adjust the wavelength, polarization and photon energy. More than 2000 scientists are using BESSY II equipment every year.

The storage ring has a circumference of 240 m and provides around 50 beamlines (paths of light rays between the storage ring and experimentation stations). The electrons are accelerated to an energy up to 1.7 GeV.

BESSY II was inaugurated in 1998 to let scientists study material structures and processes. Since 2009 it is a facility of the *Helmholtz-Zentrum Berlin für Materialien und Energie* (HZB)

In addition to the guest scientists, operators and researchers work to ensure the good functioning of the whole facility and work on refining the quality and the stability of the light rays.

1.2 BESSY II – General functioning

In the BESSY II accelerator, particles are circularly accelerated in the same direction. The goal is to produce light, instead of collisions.

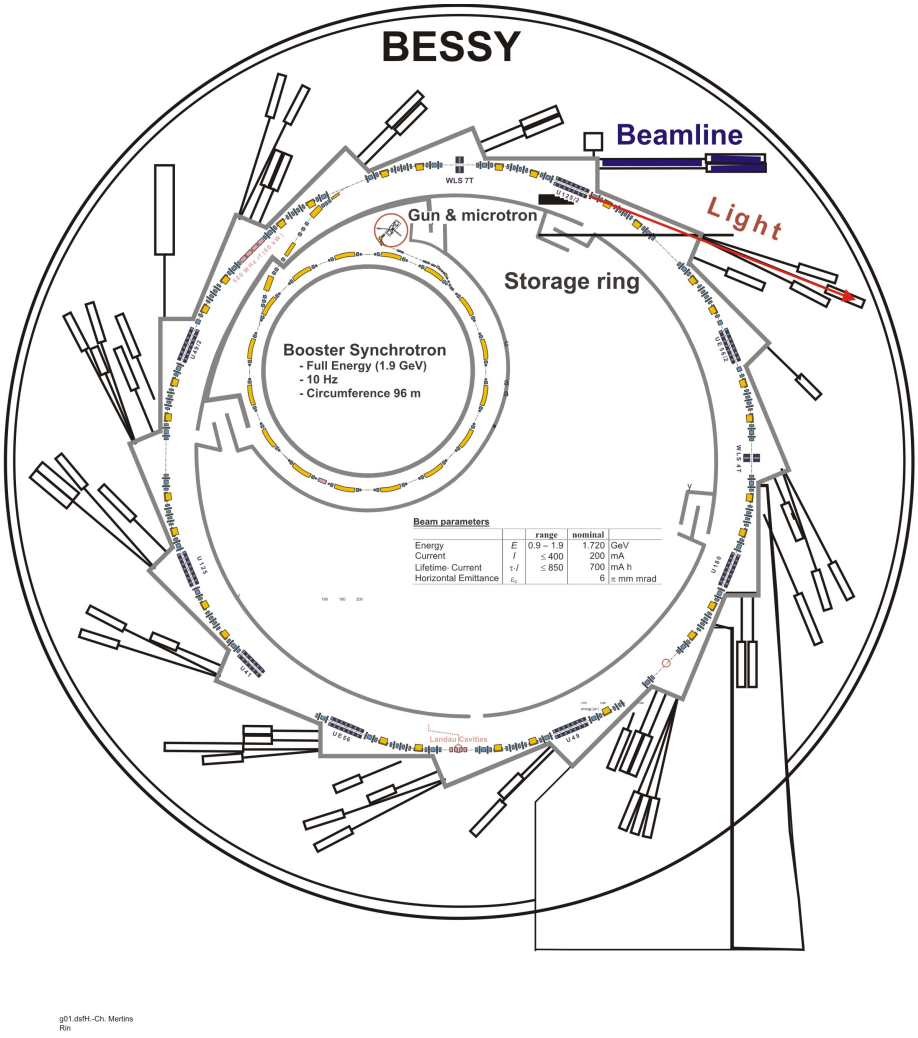


Figure 1.1: BESSY II – Accelerator chain (Source: [2])

BESSY II functions based on the synchrotron radiation phenomenon: any accelerated particle emits radiations (in the form of photons). It can be shown [1] that the radiated power in circular acceleration can be given as

$$P_s = \frac{e^2 c}{6\pi\varepsilon_0} \frac{1}{(m_0 c^2)^4} \frac{E^4}{R^2}. \tag{1.1}$$

where c is the speed of light, m_0 the rest mass (independent of the velocity) of the particle, e its charge, E its energy, R the bending radius and ε_0 the vacuum permittivity.

Considering their low mass, electrons are very good candidates to produce high energy radiation (in comparison, accelerated protons would provide 10^{13} less radiation than electrons), and are therefore used in BESSY II.

In order to reach an energy of 1.7 GeV, the electrons undergo several chained accelerations (see figure 1.2), namely:

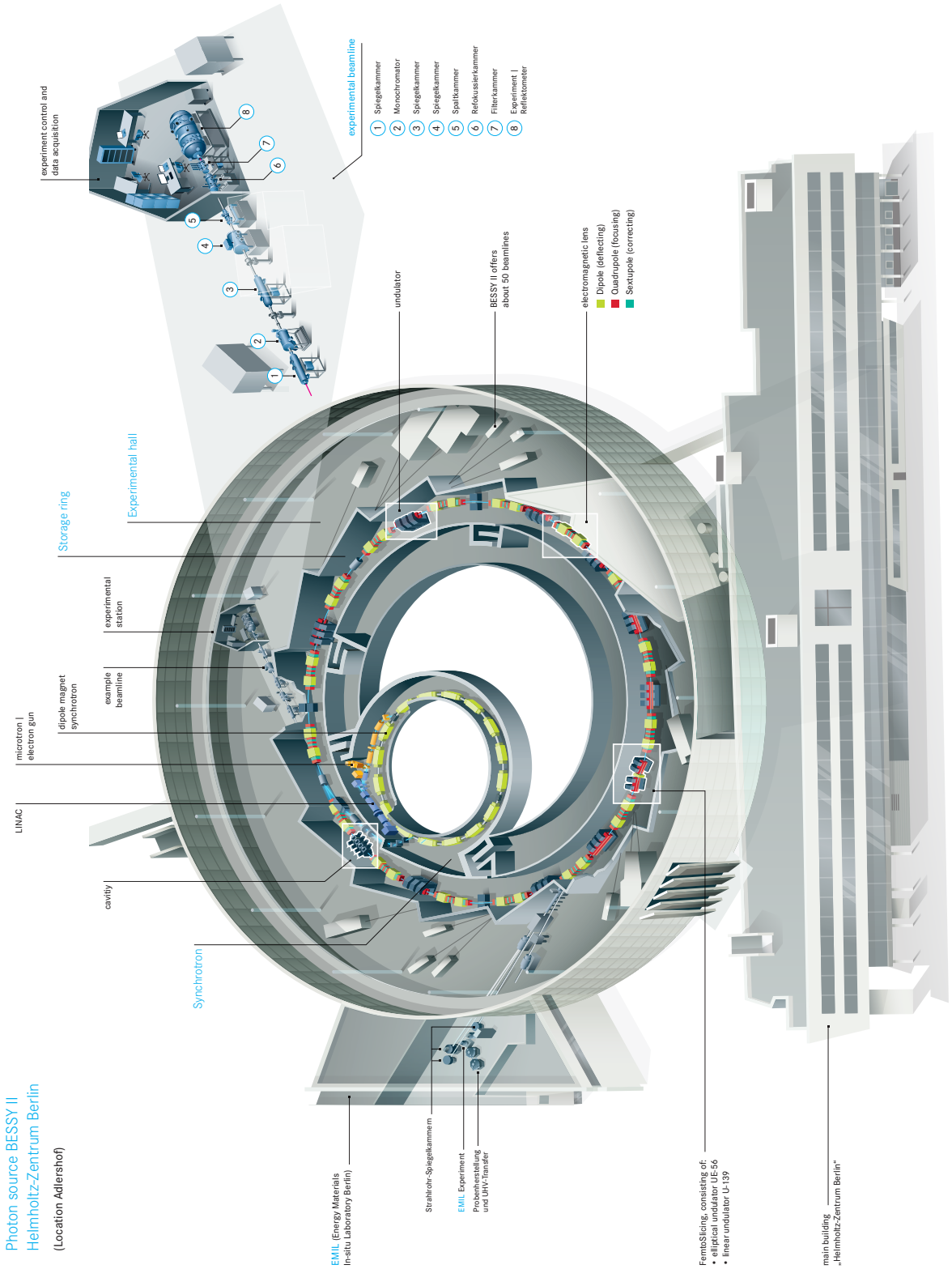


Figure 1.2: BESSY II facility (Source: [2])

1. DC electrons are provided by an electron gun at an energy of 90 keV,
2. a LINAC (**lin**ear **ac**celerator) increases the energy to 50 MeV,
3. the booster (a fast rumping synchrotron) accelerate the particles to their full energy of 1.7 GeV in not more than 30 ms.

When this energy is reached, the electrons are injected to the storage ring, which ensure that the electrons are kept at the same energy.

Another important property of BESSY II is that it is operated in *top-up mode*. This means that its current must be constant over time (in this case: 300 mA). However, electrons are likely to collide into each other or with the rest-gas atoms of the vacuum chamber. To achieve the top-up mode and counter the loss of electrons, new injections from the acceleration chain take place approximatively every 2 minutes to repopulate the storage ring.

1.3 Motivation

The most important properties of the synchrotron radiation are its brilliance and brightness, which represent the quality of the beam. The brightness describes the angular divergence of the beam, and the brilliance includes information about its transverse dimension: both are expected to be as small as possible to be in the configuration in which the beam is the most point-like. (See appendix A.1 for the exact definitions).

The main purpose of BESSY II is to provide a light radiation with high quality brilliance and brightness over time. To achieve this, the light source itself must be very stable and the electron beam very small. The stability of the orbit is required to be well bellow the transverse beam dimensions, which are for BESSY II storage ring 100 µm in the horizontal direction and 20 µm in the vertical direction.

Therefore, significant attention is drawn to the control of the storage ring orbit: the beam diameter must be as small as possible and therefore always well centered in the vacuum chamber. Although the storage ring is designed to achieve this, perturbations or misalignment in the accelerator optics occur, and must be corrected. Orbit feedback is used here to correct and stabilize the orbit and to keep it focused.

1.4 Summary

BESSY II is a powerful light source, with a nominal energy of 1.7 GeV, that produces a light of high quality, which is a wide spectrum and high brilliance. In order to keep the quality of these properties, removing perturbation sources or correcting the orbit is needed.

In the following, chapter 2 will provide the needed accelerator physics backgrounds. Chapter 3 will cover correction methods, first introducing the state of the art in the particle accelerator community, and more specially at BESSY II. Based on this, the improvement conducted during this work will be described. Chapter 4 will discuss how models and simulations where conducted in order to propose better controllers. In chapter 5, algorithms conceived and implemented during this thesis to localize perturbation sources will be presented. Finally chapter 6 will provide an overview of the results achieved with this work and some insights on how to go further.

Chapter 2

Basics of particle accelerator physics

In order to be able to localize and correct perturbation, the effects of a perturbation on a particle trajectory must be clearly understood. This section will describe the particle motion in a circular accelerator and define the parameters needed for the following explanations.

The beam trajectory can be studied in various ways. One easier and well used method stems from the field of linear beam optics (by analogy between beam and light focusing and steering). Here will be solely described what is needed in the next sections. A full reference can be found in [1]. This section is mostly inspired by the chapter 3 of this same reference.

2.1 Geometry – Frame of reference – Kinematics

The term *orbit* refers to the ideal trajectory of the particles, which is fixed by the construction of the accelerator.

The frame of reference is $K = (\underline{e}_x, \underline{e}_y, \underline{e}_s)$, with which origin follows the beam. $\underline{e}_x$ is the horizontal axis directed towards the exterior of the orbit and normal to its curve, $\underline{e}_y$ is the vertical axis and $\underline{e}_s$ the tangential axis to the orbit. Let φ be the azimuth angle, oriented by $\underline{e}_y$. Then, since $ds = -Rd\varphi$, it can be written

$$\frac{d\varphi}{dt} = -\frac{1}{R}\frac{ds}{dt}. \tag{2.1}$$

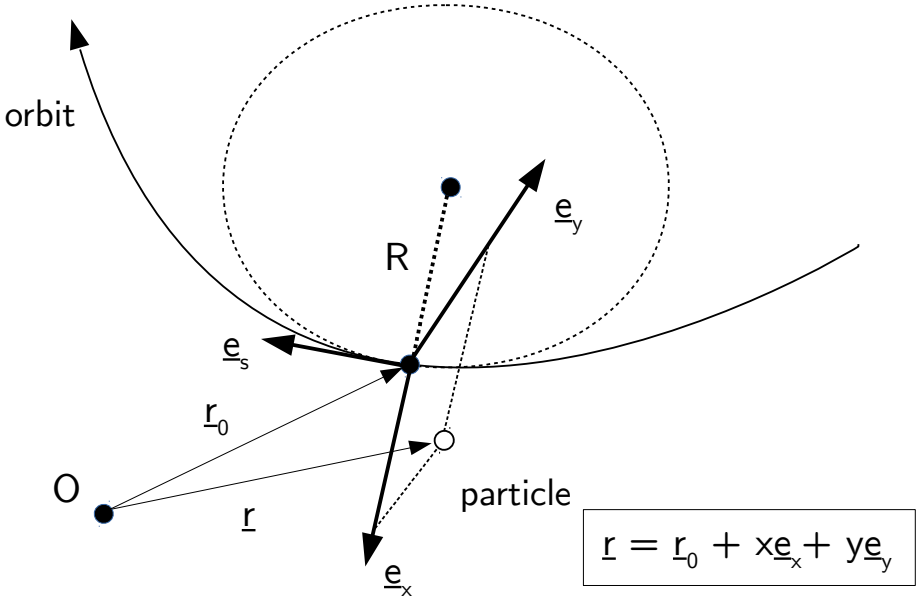


Figure 2.1: Description of the co-moving coordinate system (inspired by [1], Fig 3.2)

Using the derivation formula of polar coordinate yields

$$\begin{aligned}
\dot{\underline{e}}_x &= \frac{d\underline{e}_x}{dt} = -\dot{\varphi}\underline{e}_s = \frac{1}{R}\dot{s}\underline{e}_s \\
\dot{\underline{e}}_y &= 0 \\
\dot{\underline{e}}_s &= \frac{d\underline{e}_s}{dt} = \dot{\varphi}\underline{e}_x = -\frac{1}{R}\dot{s}\underline{e}_x.
\end{aligned} \tag{2.2}$$

Let $\underline{r} = \underline{r}_0 + x\underline{e}_x + y\underline{e}_y$ be the position of a given particle in a Galilean reference frame (with $\underline{r}_0$ the position of the origin of the moving coordinates, see figure 2.1) and let x' be the spatial derivative with respect to s so that:

$$\begin{aligned}
\dot{x} &= \frac{dx}{ds}\frac{ds}{dt} = x'\dot{s} \\
\ddot{x} &= x''\dot{s}^2 + x'\ddot{s}.
\end{aligned} \tag{2.3}$$

It follows, after some algebraic manipulations:

$$\begin{aligned}
\underline{r} &= \underline{r}_0 + x\underline{e}_x + y\underline{e}_y \\
\dot{\underline{r}} &= x'\dot{s}\underline{e}_x + y'\dot{s}\underline{e}_y + \left(1 + \frac{x}{R}\right)\dot{s}\underline{e}_s \\
\ddot{\underline{r}} &= \left[x''\dot{s}^2 + x'\ddot{s} - \left(1 + \frac{x}{R}\right)\frac{\dot{s}^2}{R}\right]\underline{e}_x + (y''\dot{s}^2 + y'\ddot{s})\underline{e}_y \\
&\qquad + \left[\frac{2}{R}x'\dot{s}^2 + \left(1 + \frac{x}{R}\right)\ddot{s}\right]\underline{e}_s
\end{aligned} \tag{2.4}$$

2.2 Assumptions

The following assumptions are taken to be valid.

A1 – The particles move essentially parallel to $\underline{e}_s$: in the first order

$$\underline{v} = v_s \underline{e}_s.$$

A2 – The magnetic field $VvecE$ has only transverse components:

$$\underline{B} = (B_x, B_y, 0).$$

A3 – The electric field is negligible with respect to the magnetic field.

A4 – The velocity of the particle varies very slowly in a magnet: $\ddot{s} \approx 0$.

A5 – The particles move at relativistic velocities, so the effect of the magnetic field is negligible on longitudinal components: only transverse components will be considered.

A6 – The momentum of the particles is $p = p_0 + \Delta p$, with the condition $\Delta p \ll p_0$ (which is well satisfied in accelerators).

2.3 Equation of motion

Newton's second law can be applied with the Lorentz force

$$\dot{\underline{p}} = e(\underline{E} + \underline{v} \times \underline{B}) \tag{2.5}$$

which according to (A3) becomes

$$\ddot{\underline{r}} = \frac{e}{m}(\dot{\underline{r}} \times \underline{B}) \tag{2.6}$$

According to (A2), the magnetic field is only transversal, which yields

$$\ddot{\underline{r}} = \frac{e}{m}(\dot{\underline{r}} \times \underline{B}) = \frac{e}{m} \begin{pmatrix} -\left(1 + \frac{x}{R}\right) \dot{s} B_y \\ \left(1 + \frac{x}{R}\right) \dot{s} B_x \\ x' \dot{s} B_y - y' \dot{s} B_x \end{pmatrix}. \tag{2.7}$$

Using (A5), the s-component does not need to be considered and using (A4) in equation (2.4) allows to remove every $\ddot{s}$ factors, so that equation (2.7) can be rewritten as

$$\begin{aligned} x'' \dot{s}^2 - \left(1 + \frac{x}{R}\right) \frac{\dot{s}^2}{R} &= -\frac{e}{m} \left(1 + \frac{x}{R}\right) \dot{s} B_y \\ y'' \dot{s}^2 &= \left(1 + \frac{x}{R}\right) \dot{s} B_x. \end{aligned} \tag{2.8}$$

Finally, because $p = mv$ and using equation (2.4) combined with (A1),

$$m = \frac{p}{v} = \frac{p}{\langle \underline{v}\,, \underline{e}_s \rangle} = \frac{p}{\left(1 + \frac{x}{R}\right) \dot{s}} \tag{2.9}$$

can be substituted to obtain

$$\begin{aligned} x'' - \left(1 + \frac{x}{R}\right)\frac{1}{R} &= -\frac{e}{p}B_y\left(1 + \frac{x}{R}\right)^2 \\ y'' &= \frac{e}{p}B_x\left(1 + \frac{x}{R}\right)^2 \end{aligned} \tag{2.10}$$

According to (A6), it can be written to first order

$$\frac{1}{p} = \frac{1}{p_0}\left(1 - \frac{\Delta p}{p_0}\right).$$

Furthermore the magnetic field can be approximated to first order with:

$$\frac{e}{p_0}B_y = \frac{1}{R} - kx \qquad \frac{e}{p_0}B_x = -ky \tag{2.11}$$

which yields

$$\begin{aligned} x'' - \left(1 + \frac{x}{R}\right)\frac{1}{R} &= -\left(\frac{1}{R} - kx\right)\left(1 - \frac{\Delta p}{p_0}\right)\left(1 + \frac{x}{R}\right)^2 \\ y'' &= -ky\left(1 - \frac{\Delta p}{p_0}\right)\left(1 + \frac{x}{R}\right)^2 \end{aligned} \tag{2.12}$$

Finally removing terms of second order ($\frac{x}{R} \ll 1$, $\frac{\Delta p}{p_0} \ll 1$) leads to the linear equations of motion in a magnetic structure:

$$\boxed{\begin{aligned} x''(s) + \left(\frac{1}{R^2(s)} - k(s)\right)x(s) &= \frac{1}{R(s)}\frac{\Delta p}{p_0} \\ y''(s) + k(s)y(s) &= 0 \end{aligned}} \tag{2.13}$$

2.4 Beta function and betatron oscillation

The beta function will provide a description of properties of a beam of many particles.

The only further assumption is that $\frac{1}{R} = 0$ and $\frac{\Delta p}{p_0} = 0$. The orbit curve is thus assumed almost flat and the momentum deviation negligible. Equation (2.13) turns into Hill's differential equation:

$$x''(s) - k(s)x(s) = 0 \tag{2.14}$$

which solution is the transverse oscillation of the orbit, or *betatron oscillation*

$$x(s) = Au(s)\cos\left(\Psi(s) + \phi\right). \tag{2.15}$$

The phase $\Psi(s)$ and the amplitude $u(s)$ depend of the position s; A and ϕ are integration constants.

Inserting equation (2.15) in (2.14) results in

$$A\left[u'' - u\Psi'^2 - ku\right]\cos\left(\Psi + \phi\right) - A\left[2u'\Psi' + u\Psi''\right]\sin\left(\Psi + \phi\right) = 0. \tag{2.16}$$

As $A \neq 0$, $u(s) \neq 0$ and $\Psi(s)$ has different values for every s, the previous equation provides two conditions

$$u'' - u\Psi'^2 - ku = 0 \tag{2.17}$$

$$2u'\Psi' + u\Psi'' = 0. \tag{2.18}$$

Equation (2.18) leads to

$$2\frac{u'}{u} + \frac{\Psi''}{\Psi'} = 0 \tag{2.19}$$

which can be integrated as

$$\Psi(s) = \int_0^s \frac{d\sigma}{u^2(\sigma)}. \tag{2.20}$$

The *beta function* $\beta(s)$ is defined as

$$\beta(s) := u^2(s). \tag{2.21}$$

Substituting $A = \sqrt{\varepsilon}$ provides the final trajectory equation

$$\boxed{x(s) = \sqrt{\varepsilon\beta(s)}\cos\left(\Psi(s) + \phi\right)} \tag{2.22}$$

The constant ε is termed *emittance* and the envelop of the orbit at each position s is $E(s) = \sqrt{\varepsilon\beta(s)}$, which defines the size of the beam.

Finally, deriving $x(s)$ leads to

$$\begin{aligned} x'(s) &= \frac{\sqrt{\varepsilon\beta'(s)}}{2\sqrt{\beta(s)}}\cos\left(\Psi(s) + \phi\right) - \sqrt{\varepsilon\beta(s)}\frac{1}{\beta(s)}\sin\left(\Psi(s) + \phi\right) \\ &= -\sqrt{\frac{\varepsilon}{\beta(s)}}\left[\alpha(s)\cos\left(\Psi(s) + \phi\right) + \sin\left(\Psi(s) + \phi\right)\right] \end{aligned} \tag{2.23}$$

with

$$\alpha(s) := -\frac{\beta'(s)}{2} \tag{2.24}$$

Since the beta function controls the size of the beam, this is the function that must be finely controlled. A matrix calculation method is presented in [1] in order to calculate $\beta(s)$ and $\alpha(s)$ step by step around the orbit. Once these are known, the particle trajectory can be easily calculated through the matrix multiplication

$$\begin{pmatrix} x(s) \\ x'(s) \end{pmatrix} = \underline{\underline{M}} \begin{pmatrix} x(0) \\ x'(0) \end{pmatrix}. \tag{2.25}$$

From the equations of $x(s)$ and $x'(s)$, some trigonometrical manipulations and choosing $\phi(0) = 0$ it can be shown that (with simplified notations)

$$\underline{\underline{M}} = \begin{pmatrix} \sqrt{\frac{\beta}{\beta_0}}\left(\cos\Psi + \alpha_0\sin\Psi\right) & \sqrt{\beta\beta_0}\sin\Psi \\ \frac{(\alpha_0 - \alpha)\cos\Psi - (1 + \alpha_0\alpha)\sin\Psi}{\sqrt{\beta\beta_0}} & \sqrt{\frac{\beta_0}{\beta}}\left(\cos\Psi - \alpha\sin\Psi\right) \end{pmatrix}. \tag{2.26}$$

The two functions Ψ and β will be extensively used in the next sections as the perturbation localization process and several correction methods rely on them.

Note — The calculation done for the x direction can be done similarly for the y direction, which defines a $\beta_x(s)$, $\beta_y(s)$, ε_x, ε_y, $\Psi_x(s)$ and $\Psi_y(s)$

2.5 Tune

In circular accelerators, the particles undergo periodically the same magnetic fields and therefore the same force. This can lead the amplitude of the beam transverse oscillation to increase, putting particles at risk to be lost in the vacuum chamber walls. This phenomenon is called optical resonance, and it must be avoided.

As exposed in sections 2.3 and 2.4, if $\frac{\Delta p}{p_0} = 0$ and $K(s) = \frac{1}{R(s)} - k(s)$, the trajectory is given by

$$x''(s) + K(s)x(s) = 0. \tag{2.27}$$

Since $K(s)$ is periodic with the circumference L of the ring, then the trajectory is again (according to Floquet's theorem)

$$x(s) = \sqrt{\varepsilon\beta(s)}\cos\left(\Psi(s) + \Phi\right) \tag{2.28}$$

with $\beta(s)$ periodic of period L as well. The resonant behavior depending on the betatron phase $\Delta\Psi = \Psi(s+L) - \Psi(s)$, the tune Q of the accelerator is defined as

$$Q = \frac{\Delta\Psi}{2\pi} = \frac{1}{2\pi}\oint\frac{ds}{\beta(s)}. \tag{2.29}$$

The tune is a constant for a circular accelerator: it describes the *number of betatron oscillation* undergone by a particle per revolution.

It can be shown that tune values that satisfy

$$mQ_x + nQ_y = p \qquad (m,n,p) \in \mathbb{Z} \tag{2.30}$$

produce coupled resonance of order $|m| + |n|$. A pair of values (Q_x, Q_y) is used as a working point so that it either does not satisfy the equation (2.30) or with a high order (above the 5th order for instance).

2.6 Summary

Two very important functions were defined here: $\beta(s)$ and $\Psi(s)$, defined in equations (2.20) and (2.21), respectively the variation of the trajectory amplitude and the betatron phase.

These functions highly depend on the accelerator optics and when determined, knowing the real objects that produce them (e.g. magnets, coils) is no longer needed.

The tune Q is a constant parameter defined as the number of betatron oscillation per revolution.

The following sections will rely on these three variables as they fully represent the orbit and its variations.

Chapter 3

Orbit correction

3.1 Motivation

The accelerators are designed so that the particles follow a given path, which is defined in the case of synchrotrons by the successive bending imposed by dipole magnets. Quadrupole magnets are additionally needed to provide focusing and guarantee the closing of the orbit. As the precision of the positioning of the magnets is limited, some errors may destabilize the orbit and increase the deviation of the particles around the theoretical orbit.

In addition, the environment produces perturbations: for instance the 50 Hz of the main power, some imperfectly isolated magnetic sources (like the booster at 10 Hz). Figure 3.1 shows the spectrum of the beam displacement from the reference orbit before correction.

In order not to lose electrons in the walls of the vacuum chamber but also to increase the brightness of the synchrotron radiations (and therefore to have focused electron beams), all these residual misalignment and magnetic field errors must be corrected.

The most classical correction methods will be presented here, before explicitly describing the case of BESSY II and the improvement proposed during this thesis.

3.2 Monitoring and correction instruments

To be able to correct the orbit or localize perturbations, some tools must be used.

3.2.1 Beam Position Monitors

The position of the beam in a given direction is monitored with *beam position monitors* (BPM).

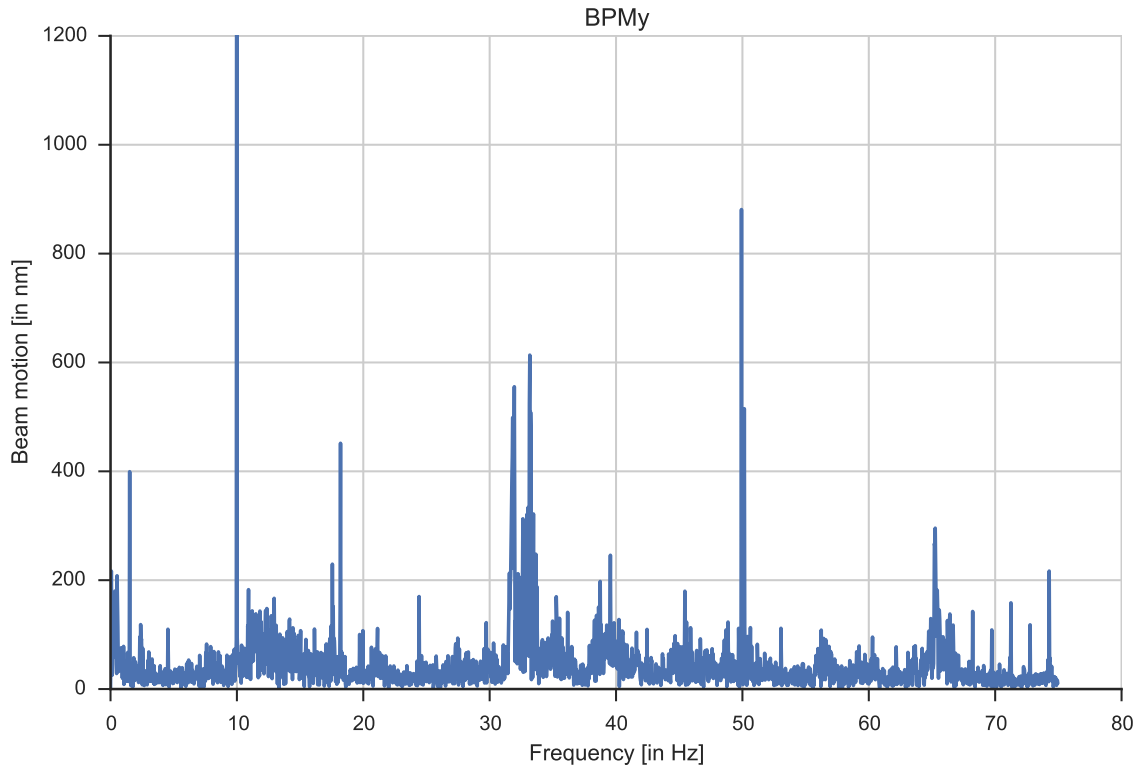


Figure 3.1: Spectrum of the beam motion, without correction (vertical)

Several types of BPMs exist, but the important characteristics of the type used at BESSY II are that

- it is a non invasive method (it does not affect the beam or in a negligible way)
- it outputs an electric signal proportional to the distance of the beam from an arbitrary point.

This last property means that the measured value must always be subtracted by a reference value.

The raw values output by the BPMs are thus never considered and every orbit position value given (also called BPM value by misuse of language) is always at position s and time t

$$\Delta x_{\mathrm{BPM}}(s,t) = x_{\mathrm{BPM}}(s,t) - x_{\mathrm{BPM,ref}}(s) \tag{3.1}$$

3.2.2 Correctors

To correct the orbit, BESSY II uses dipole magnets, also called *corrector magnets* (CM) positioned around the storage ring. Each dipole contributes to correcting in a given direction, at a specific position.

3.2.3 Monitor and corrector numbers

The number of monitors and correctors is quite important. Because the correction method used is based on an inversion problem (see section 3.3.3), it is important to have an over-constrained problem. In the opposite case, a perfect correction would be always reached at monitor positions but unconstrained (and thus potentially arbitrary bad) elsewhere.

Therefore, the number of BPMs is set quite greater than the number of correctors.

At BESSY II there are

- 108 BPMs, measuring both horizontal (BPM_x) and vertical (BPM_y) direction,
- 48 CMs for the horizontal direction (CM_x) and 64 for the vertical direction (CM_y).

3.3 Orbit correction methods

Several global correction methods are well documented in the literature. Local orbit bumps (presented in section 3.3.1) allow local correction and are used to change the path of the orbit (during the injection time for example). The most common ones are the *best corrector method* and the *response matrix method* (see sections 3.3.2 and 3.3.3), as they provide a global correction over the whole orbit.

3.3.1 Local orbit bumps

Using local orbit bumps is a basic method that gives the operator a total control on local orbit modifications.

3.3.1.1 Principle

Adding a simple dipole on the path of a particle will bring it closer or push it away. As the particles must eventually return to the planned orbit, a series of dipoles can be set one after the other to design an arbitrary path. Figure 3.2 shows a minimal example, where the black points represent the magnets, and the dashed line the original orbit. Each magnet M_i has the position s_i, where the phase is Ψ_i and the beta function β_i. The orbit motion at this position is described by the vector $\underline{x}_i = (x_i, x_i')$.

The strengths κ_1, κ_2, κ_3 of each magnets can be calculated with the formalism introduced in section 2.4, equation (2.26) in order to achieve the right displacement and respect the closed orbit condition. The following conditions must be fulfilled (closed orbit):

$$\underline{x}_1 = \begin{pmatrix} 0 \\ \kappa_1 \end{pmatrix} \qquad \text{and} \qquad \underline{x}_3 = \begin{pmatrix} 0 \\ -\kappa_3 \end{pmatrix} \tag{3.2}$$

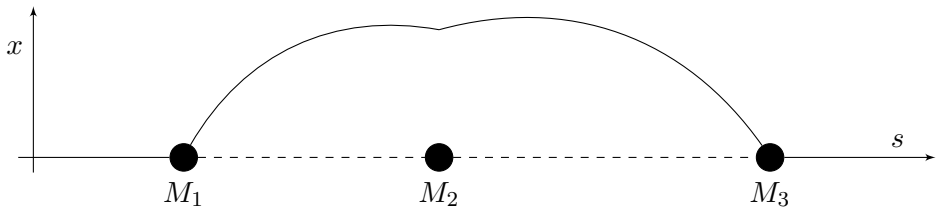


Figure 3.2: Local bump with 3 magnets (inspired by [1], Fig 3.47)

The orbit motion at s_3 is the superposition of the effects of M_1 and M_2 alone. It can thus be written

$$\underline{x}_3 = \underline{\underline{M}}_{1\to3}\underline{x}_1 + \underline{\underline{M}}_{2\to3}\begin{pmatrix}0\\ \kappa_2\end{pmatrix} \tag{3.3}$$

or

$$\begin{pmatrix}0\\ -\kappa_3\end{pmatrix} = \begin{pmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{pmatrix}\begin{pmatrix}0\\ \kappa_1\end{pmatrix} + \begin{pmatrix}b_{11} & b_{12}\\ b_{21} & b_{22}\end{pmatrix}\begin{pmatrix}0\\ \kappa_2\end{pmatrix}$$

Solving this system of equations and replacing the coefficients of the matrix by their expression in equation (2.26) leads to the values of κ_2 and κ_3 in function of κ_1 (see [1], p.130 for the exact equations). This can allow an arbitrary displacement of the orbit at a given position. This can be extended to a version with N magnets, which allows the free setting of $N-2$ parameters (x or x'). This is extensively described in [1].

3.3.1.2 In practice

This solution is used, for instance, to shift a part of the orbit during particle injections, in order not to disturb already stored particles. It can also be used to counter a known localized perturbation of which source cannot be removed.

It is indeed very efficient to locally shift the orbit. However, it cannot be used for correction, as that needs a precise knowledge of the optics of the ring, which is not available (some devices have no theoretical optic model). Further more, it would be an iterative method as the correction would need to be calculated and applied for each orbit segment.

3.3.2 Most effective corrector method

This method is based on the fact that orbit shifts are often caused by strong localized disturbances. Its goal is to correct each particular disturbance.

3.3.2.1 Principle

Given a distorted orbit, the optimal gain for each corrector is calculated by a mean square error algorithm (see equation (3.4) and [1]). The corrector which provides the best correction

is selected: it is the most effective corrector.

Let's assume that the ith corrector, at position s_i, has the optical parameters β_i, α_i and Ψ_i, and that m monitors are set around the orbit with parameters β_j, α_j and Ψ_j, and read a displacement u_j from the reference orbit ($1 \leq j \leq m$).

The strength κ_i of the field at the position s_i is obtained minimizing the function

$$f_i(\kappa_i) = \sum_{j=1}^{m} (u_j - x_{ij}(\kappa_i))^2 = \sum_{j=1}^{m} (u_j - \kappa_i h_{ij})^2 \tag{3.4}$$

with, if $\Delta\Psi_{ij} := \Psi_j - \Psi_i$,

$$\begin{aligned} x_{ij}(\kappa_i) &= \kappa_i h_{ij} \\ &= \kappa_i \frac{\sqrt{\beta_i \beta_j}}{2} \left[\frac{\cos(\Delta\Psi_{ij}) - 2\alpha_i \sin(\Delta\Psi_{ij})}{\tan(\pi Q)} + \sin(\Delta\Psi_{ij}) \right]. \end{aligned} \tag{3.5}$$

It follows that

$$\kappa_i = \frac{\sum_{j=1}^{m} u_j h_{ij}}{\sum_{j=1}^{m} h_{ij}^2} \tag{3.6}$$

The ith corrector is attributed the gain $-\kappa_i$ to compensate the field.

3.3.2.2 Iterative version

When the most effective corrector is found, the process is reiterated on the corrected orbit with the remaining correctors. By doing this until all correctors are used (or until adding a correction does not improve the orbit), a comprehensive correction is reached.

3.3.2.3 Practical issue

The problem of this method is that each corrector must be tested once, and this for each iteration: the initialization of the correction is long and is then fixed. Moreover, the correction is less efficient than the other ones presented here [1].

3.3.3 Response matrix

This section is mainly based on [1], [3] and [4].

3.3.3.1 Inverse problem

This correction is based on solving the following inverse problem:

With the expected orbit being known, how must the correctors be set in order to achieve this orbit?

To solve this problem, the response matrix $\underline{\underline{S}}$ is introduced. It is defined by the equation $\underline{X} = \underline{\underline{S}}\,\underline{\Theta}$ where $\underline{\Theta}$ is the *kick vector* (i.e. the vector of strength of the field generated by each corrector) and $\underline{X}$ the vector of *orbit change*. If the accelerator has M monitors and N correctors, then $\underline{\underline{S}}$ is a $M \times N$ matrix. $\underline{\underline{S}}$ is often termed *forward* or *observation matrix* because it describes the effects of a given phenomenon. Indeed, each coefficient S_{ij} of the matrix is the orbit change at the position s_i (of the ith monitor), for a kick of unity 1 at the position s_j (of the jth corrector).

Inversing the response matrix will provide the correction to apply. Since it's very common to have more monitors than correctors, the matrix is not square. A *singular value decomposition* (SVD) is used on the matrix to provide a pseudo-inversion $\underline{\underline{S}}^*$ of $\underline{\underline{S}}$.

Using the SVD also makes it possible to use only the most significant singular values in the correction. This prevents the orbit from being over-corrected, by not considering less significant values that can result of noise in the monitors during the measuring, as well as numerical artifacts or incorrections in the response matrix.

The correction can then be performed

$$\underline{\Theta} = \underline{\underline{S}}^* \underline{X}. \tag{3.7}$$

3.3.3.2 Acquisition of the response matrix

Before solving the inverse problem, the response matrix must first be determined. This can be achieved in two different ways: by describing the magnet structure and explicitly calculating the matrix, or by acquiring it in an experimental way.

Calculation of the response matrix The matrix can be theoretically calculated by using the accelerator model and physics. This was already used in section 3.3.2 and using the same calculations leads to setting, for the ith monitor and jth corrector,

$$S_{ij} = h_{ij} \tag{3.8}$$

with h_{ij} as defined in equation (3.5).

The main flaw of this method is that it is only a model which may not exactly represent the reality. Some misalignments of magnets or external magnetic perturbations would not be taken intp account. It is, however, a good first approach.

Experimental acquisition of the response matrix A more common and precise way of constructing the response matrix is empirical: it suffices to give a unitary value to a corrector and to read the monitors to obtain a column $\underline{S}$. Doing so for each corrector provides the whole matrix.

3.4 State of the art at BESSY II

BESSY II currently implements a Fast Orbit Feedback (FOFB), which is based on the response matrix inverse. The correction algorithm will be first described in section 3.4.1, and current choices will be discussed by references found in literature. A technical overview will then be provided in section 3.4.2, presenting the actual processes taking part in the correction.

3.4.1 The correction

Using the response matrix to correct the orbit is quite common [1, 4, 5]. However, as already highlighted in section 3.3.3, the matrix never perfectly represents the orbit and because of noises in the environment, it is not sufficient. In general only the largest singular values are used and some additional corrections are needed in the frequency domain.

To cope with these additional requirements, a PID correction (proportional response with gain K_p, integral response with gain K_i and derivative response with gain K_d) is used at BESSY II. It also deals with the delays introduced by the process between the time the orbit is read by the BPMs and the moment the correction is actually applied [4].

The full process goes as presented in figure 3.3.

The control value at correction cycle n is the differential orbit

$$\Delta\underline{X}[n] := \underline{X}[n] - \underline{X}_{\text{ref}} \tag{3.9}$$

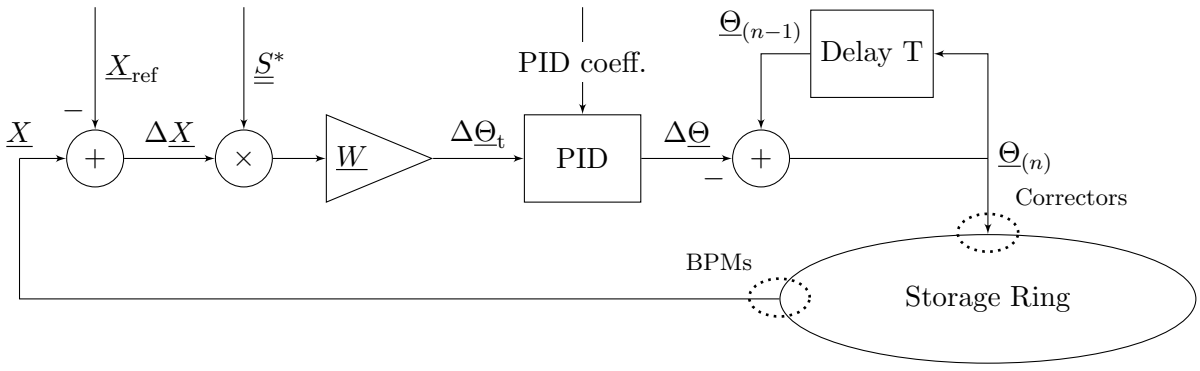


Figure 3.3: The correction process used at BESSY II

which is expected to be close to 0. The correction provided by the weighted response matrix is then processed

$$\Delta\underline{\Theta}_{\mathrm{t}}[n] := \left(\underline{\underline{S}}^{-1} \cdot \Delta\underline{X}[n]\right) \odot \underline{W} \tag{3.10}$$

(with $\odot$ the point-wise multiplication and $\underline{W}$ a vector of weights) and modulated by the PID correction. The ideal PID in the Laplace form is described with

$$H_{PID}(s) = K_p + \frac{K_i}{s} + K_d \cdot s \tag{3.11}$$

or, with the z transform (calculated with Euler's method),

$$H_{PID}(z) = K_p + \frac{K_i}{1-z} + K_d \cdot (1-z) \tag{3.12}$$

which yields

$$\Delta\underline{\Theta}[n] = K_p \cdot \Delta\underline{\Theta}_{\mathrm{t}}[n] + K_i \cdot \sum_{k=0}^{n-1} \Delta\underline{\Theta}_{\mathrm{t}}[k] + K_d \cdot \left(\Delta\underline{\Theta}_{\mathrm{t}}[n] - \Delta\underline{\Theta}_{\mathrm{t}}[n-1]\right). \tag{3.13}$$

The real correction is eventually the accumulation of all corrections

$$\underline{\Theta}(n) = \underline{\Theta}[n-1] - \Delta\underline{\Theta}[n]. \tag{3.14}$$

For stability reasons, the PID was always implemented as a pure proportional corrector (K_i and K_d being set to zero). Indeed, because of the last accumulation, it was of course behaving like a proportional integrator (PI), which forbade the use of the other coefficients in the PID correction. In the simulations described in section 4.3, a corrector PID can be implemented with more efficient coefficients. It is, however, important that the correction starts from an initial value known to be stable, so that the beam stays focused.

3.4.2 Technical overview

The correction is naturally automatized. Because the read/write actions should be really fast, a specific infrastructure was designed. This is represented on figure 3.4.

All elements are connected to a reflective memory (RFM), which provides a high speed and low latency interface. This memory space is split in specified divisions to prevent data collisions: one part contains the configuration, a second the orbit values and the last one the correction values.

Actors The following actors are playing a role in the process:

- The **analog-to-digital converter (ADC)** reads the orbit displacement values from the BPMs and makes them available to the other actors.

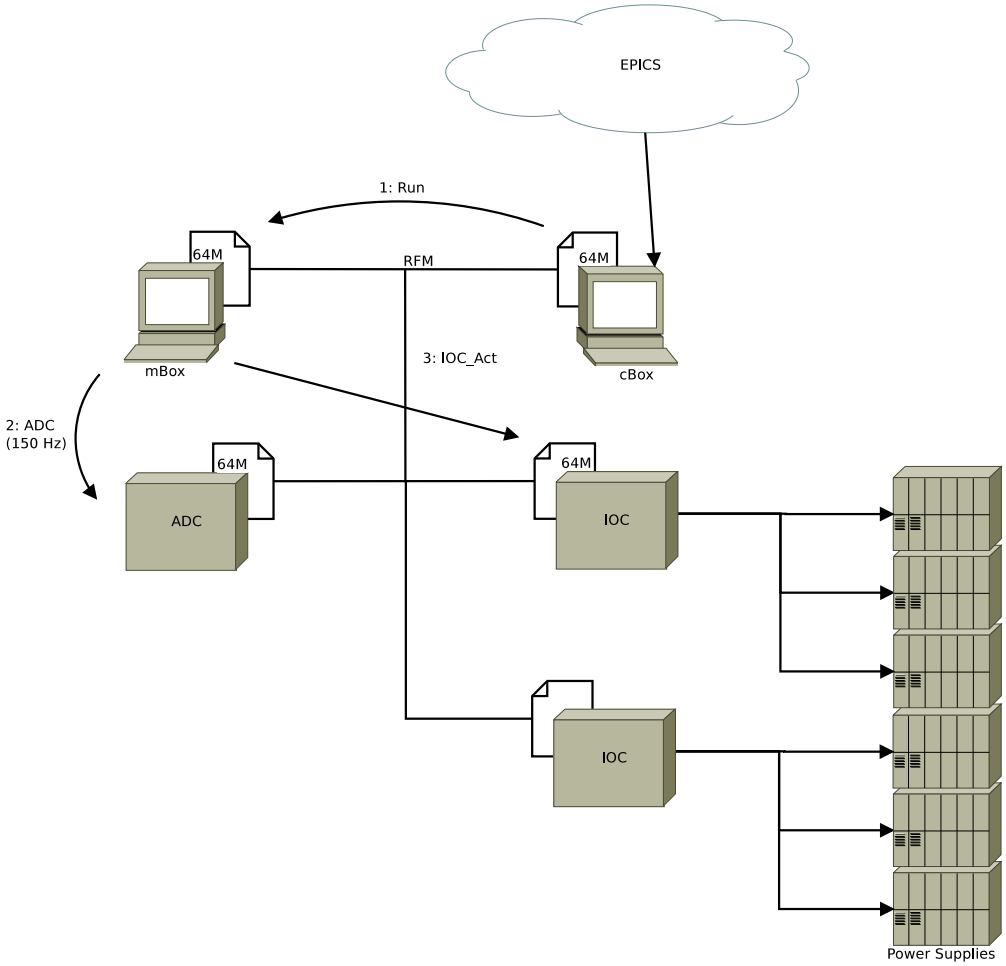


Figure 3.4: cBox and mBox: the correction infrastructure at BESSY II

- The **cBox** controls (= c) the correction. It defines when to read the values of the BPMs and when to write the new correction values, and it provides initializations values. The operators are communicating with this process to configure the correction.
- The **mBox** does the maths (= m) of the correction. When allowed by the cBox, it reads the BPMs values, does the maths to define the new correction values (multiplication with the response matrix, PID correction) and writes them to the communication bus. This process also publishes the values it reads and writes so that client programs can subscribe to this data stream and reuse the values internally.
- The **input-output controllers (IOCs)** write the correction values to the power supplies commands when they are available.
- The **power supplies (PS)** provide a given power to the corrector magnets.

Process description After having received the authorization to run from the cBox, the mBox process starts. It first does its full initialization, which only happens upon start (the response matrix, the reference orbit, the PID parameters, etc. cannot change while the mBox process is running), and starts the ADC. From this time, the ADC will write new orbit values to the RFM at a rate of 150 Hz.

The mBox waits for the ADC interruption to read the new orbit data from the RFM. The correction is then calculated (in Amperes) and the values are converted in the DAC input format (basically unsigned integers). This data is written back to RFM and an interruption is sent to inform the other elements that the new values are available. This is read by the IOCs that relay to the power supplies powering the corrector magnets.

This defines a real time process, repeated at the frequency of 150 Hz.

The duration of each step is as following:

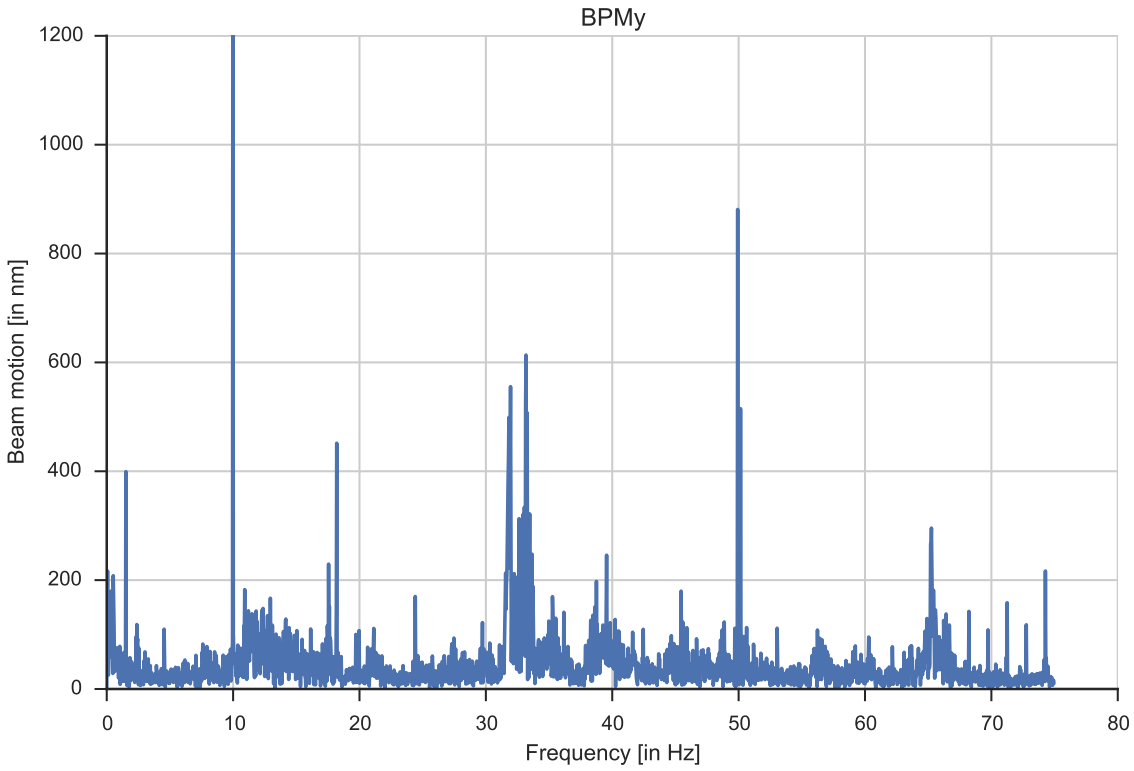
- orbit value acquisition (BPMs): 0.3 to 1.7 ms
- correction computation: 0.5 to 1 ms
- transmission of the correction date to the power supplies: 0.5 to 1 ms

Note — One of the side works done during this thesis was rewriting the whole mBox in C++ in a very object oriented way to allow a smaller computation time (the original one was written in MATLAB), easier improvement and feature additions (like a dynamic correction, easier communication with external scripts, production/experimental modes, etc.). This is fully presented in appendix B.

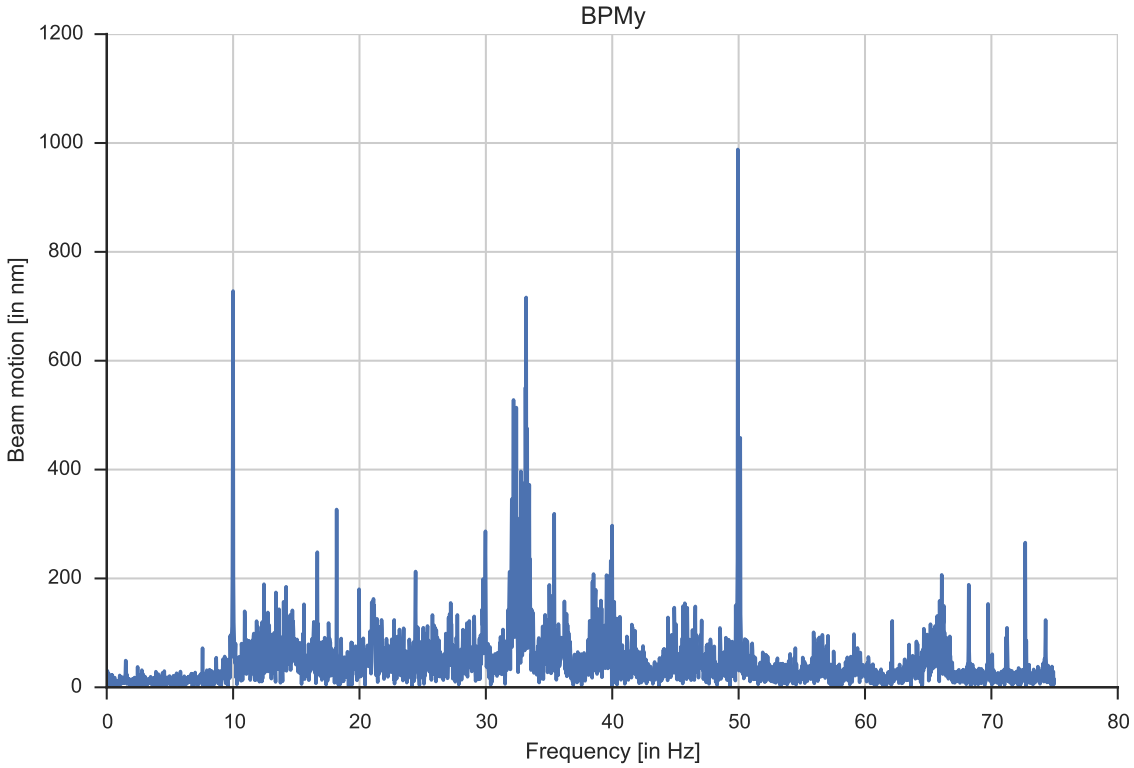
3.4.3 Overview and needed improvements

The results of this correction can be seen in figure 3.5: the low frequencies and most of the peaks are reduced, which is a real gain for the end users of the accelerator. These correction results where reached again after having rewritten the mBox in C++, which ensured that no dramatic regressions were introduced and allowed going forward.

It must be noted that correction gain provides only a correction up to ~ 12 Hz, and increasing it would let the system become unstable. The presented correction is therefore not always sufficient and some specific frequencies (like for example 10, 17.5 or 32 Hz) must be corrected with a separate method dedicated to them, which would need to be added to the full correction process.



(a) No correction



(b) FOFB correction

Figure 3.5: Orbit motion spectrum: with and without FOFB (vertical, current: 300 mA)

3.5 Improvement of the correction for harmonic perturbations

The correction presented until now, which already existed at BESSY II, only provides a correction for the continuous component. It can be seen in figure 3.5, that the correction drastically reduces the perturbations in the lower frequencies (below 10 Hz) but other frequencies still have high components. This section will provide a method implemented during this thesis for a dedicated correction of harmonic perturbations.

3.5.1 General method

A simple extension of the previous method is to analyze a given harmonic of the orbit, and consider the Fourier coefficient of each monitor as the complex amplitude of the harmonic orbit of interest

$$\forall f \in \mathbb{R}, \qquad X_f = \frac{2}{T}\int_0^T x(t)e^{-j2\pi ft}dt \tag{3.15}$$

or, in the digital domain,

$$\forall f \in \left[0, \frac{Fs}{2}\right], \qquad X_f = \frac{2}{N}\sum_{k=0}^{N-1} x(k)e^{-j2\pi f\frac{k}{Fs}} \tag{3.16}$$

where F_s is the sampling frequency, $f = \frac{1}{T}$ the frequency of interest and N the number of samples.

This method is based on the decomposition of the signal in orthogonal functions. With the digitalization, the harmonic function are orthogonal only when the frequency f is a multiple of $\frac{F_s}{N}$. This can be a problem here, as the frequency to analyze might be exactly 10 Hz (see section 3.5.2) and not 9.975 Hz if $N = 2000$, $F_s = 150$ Hz. The input signal should therefore have as many samples as possible so that the frequency of interest can be close to the precedent condition.

This approach provides a complex differential orbit $\Delta\underline{X}_f$ for the harmonic f. From there, the correction values can be calculated with the traditional method

$$\Delta\underline{\Theta}_f = \underline{\underline{S}}^* \, \Delta\underline{X}_f \in \mathbb{C}^n. \tag{3.17}$$

To apply the correction, an oscillation must finally be generated with the obtained parameters. This is, for the ith corrector

$$\forall t \in \mathbb{R}, \qquad y_i(t) = |\Delta\Theta_{f,i}| \cdot \cos\left(2\pi ft - \arg\left(\Delta\Theta_{f,i}\right)\right). \tag{3.18}$$

The main problem with the method as presented here is that the generated signal phase must be synchronized with the time at which the correction parameters were measured. This is possible only for perturbations for which the source is already known and could be used as reference signal.

3.5.2 Example of the 10 Hz perturbation

The 10 Hz perturbation has a well known source – the booster power supply (pre-accelerator). This perturbation has a great impact on the orbit spectrum, and could be used as a validation case.

A coil was used to measure the field generated by a magnet powered by the same source as the booster. It provides a 10 Hz reference, which is synchronized with the perturbation.

The orbit and the reference signal were measured during the exact same time. Both were analyzed with the method presented above. Instead of using equation (3.18), the reference signal was used and its phase and amplitude where changed for each corrector with the help of a finite impulse response (FIR) filter. Because the sampling frequency is 150 Hz, a FIR with $N_{\text{taps}} = 15$ for the ith corrector can be designed as follow

$$h_i[n] = \frac{2}{N_{\text{taps}}} \cdot \frac{|\Delta\Theta_{f,i}|}{|A_{10}|} \cdot \cos\left(2\pi \cdot \frac{f_{10}}{F_s} n - [\arg(\Delta\Theta_{f,i}) - \phi_{10}]\right), \quad n \in 0..N_{\text{taps}} - 1 \tag{3.19}$$

where A_{10} and ϕ_{10} are respectively the amplitude and the phase of the reference signal, $F_s = 150\,\text{Hz}$, $f_{10} = 10\,\text{Hz}$.

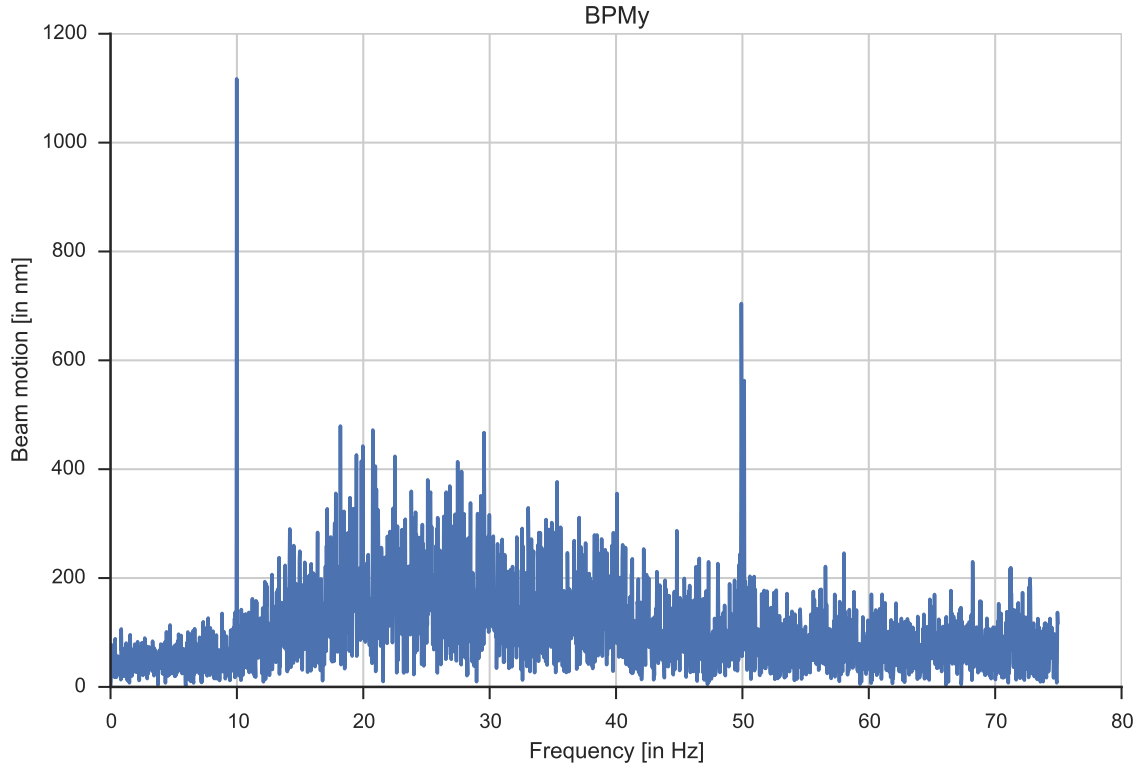
This can then be applied to the reference signal x_{10}: for the ith corrector, the dynamic correction is

$$y[n] = h * x_{10}[n] = \sum_{k=0}^{14} h[k] \cdot x_{10}[n-k] \tag{3.20}$$

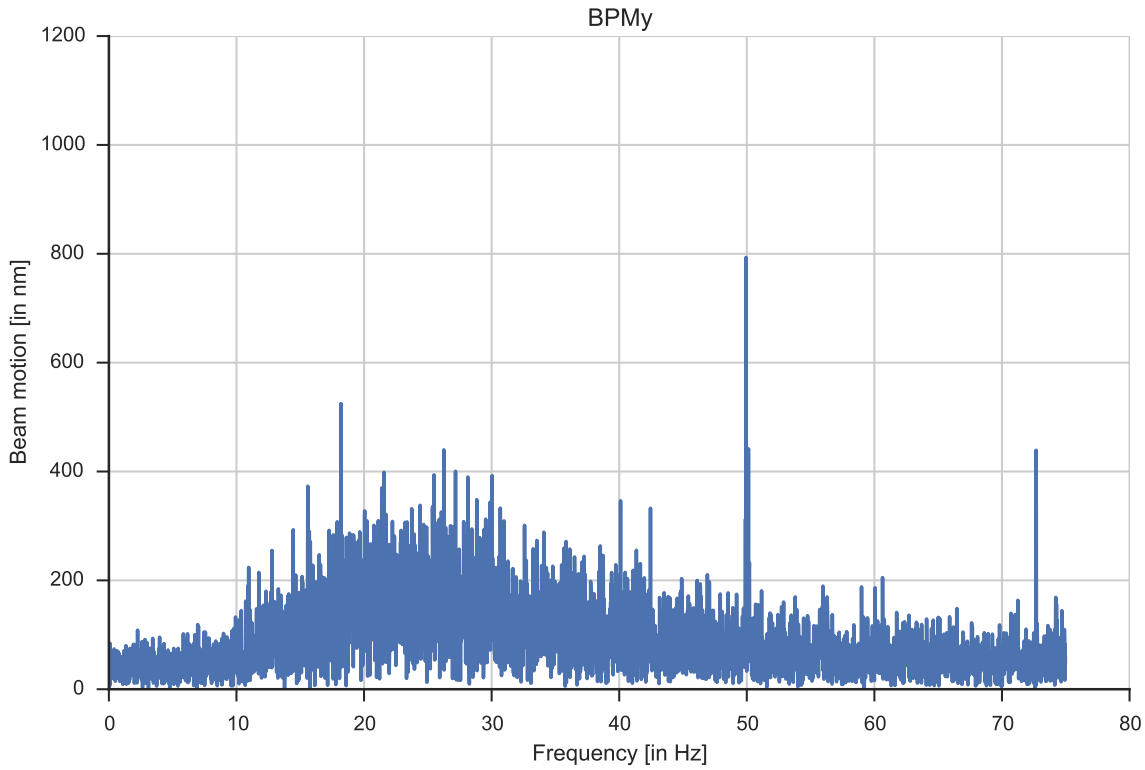
The result of the algorithm is shown in figure 3.6. It can be seen that the perturbation is completely removed. Interestingly enough, the FIR filter needed an additional phase rotation of $-\frac{3}{4}\pi$ rad to obtain a good result. It should be noted that the storage ring was filled with low current only.

3.6 Summary

In order to be sure that the electron beam remains focused and stable, a control process is needed. The concerned hardware and software stack at BESSY II was first studied and the main actor (the mBox) rewritten in a more modern and expendable way. To the original



(a) FOFB only



(b) FOFB + 10 Hz correction

Figure 3.6: Orbit motion spectrum: with and without 10 Hz harmonic correction (vertical, current: 15 mA)

correction was added a dynamic correction, that can correct static harmonic perturbations and almost completely remove them.

Chapter 4

System and correction simulation

One difficulty in improving the current correction process is that machine time is expensive: tests cannot be done very often. A poor correction can indeed destabilize the orbit, or, worse, lose the electrons in the vacuum chamber walls. Thus, it cannot be tested when the storage ring is on service, or other scientists do experiments or measurements for their own research. Furthermore, as presented previously, the process is complex and one must be sure that every actor will play correctly in the improvement proposition. Simulating the whole environment is expected to mitigate this problem, and reduce the online experimental time needed during nights and weekends.

Additionally, based on previous observations, the correction architecture was reexamined. The simulation environment made possible to test it and propose new parameters.

A short theoretical background will be first presented to introduce the mathematical formalism needed to describe control processes. Based on this, the way the system identification was conducted to model the storage ring will be presented. As the study of the correction architecture at BESSY II showed that some improvements in the correction process were possible, a new architecture of the correction, allowing better results, will be described. Finally the simulation results will be discussed.

4.1 Theoretical background

4.1.1 System

To model a practical process, a system is used.

A system is defined as an operator $\mathcal{L}$, changing a set of q inputs $\underline{u}$ into p outputs $\underline{y}$. Generally, the inputs, outputs and operator can vary with time, position, temperature or any

possible variable.

For example if the inputs, outputs and operators are only time dependent:

$$\underline{y}(t) = \mathcal{L}[\underline{u}(t), t] \tag{4.1}$$

4.1.2 Linear time invariant system

A linear time invariant (LTI) system is

- **time invariant**: the operator $\mathcal{L}$ does not depend on the time, meaning that a given input will always generate the same output: $\mathcal{L}[\underline{u}, t] = \mathcal{L}[\underline{u}]$. This can also being described as

$$\underline{y}(t) = \mathcal{L}[\underline{u}(t)] \implies \forall \tau, \quad \underline{y}(t-\tau) = \mathcal{L}[\underline{u}(t-\tau)] \tag{4.2}$$

- **linear**: the operator $\mathcal{L}$ is linear

$$\begin{cases} \underline{y}_1(t) = \mathcal{L}[\underline{u}_1(t)] \\ \underline{y}_2(t) = \mathcal{L}[\underline{u}_2(t)] \end{cases} \implies \forall (\alpha, \beta) \in \mathbb{R}^2, \quad \alpha \underline{y}_1(t) + \beta \underline{y}_2(t) = \mathcal{L}[\alpha \underline{u}_1(t) + \beta \underline{u}_2(t)] \tag{4.3}$$

  This linearity property allows analyzing any complicated input as a sum of easier ones that will be superposed. The system complexity can thus be more easily handled.

It can be proven that in a LTI system, a harmonic input signal of frequency f_0 will produced an harmonic output signal with the same frequency. Only the amplitude and the phase will be different.

4.1.3 Transfer function

LTI systems can be described with specific operators called transfer functions.

Let a system with a single input and single output (SISO) be described as

$$\sum_{k=0}^{M} a_k \frac{d^k y}{dt^k}(t) = \sum_{j=0}^{N} b_j \frac{d^j u}{dt^j}(t). \tag{4.4}$$

Taking its Laplace transform yields

$$\sum_{k=0}^{N} a_k s^k Y(s) = \sum_{j=0}^{M} b_j s^j U(s). \tag{4.5}$$

where $Y(s) = \mathcal{L}\{y(t)\}$ and $U(s) = \mathcal{L}\{u(t)\}$ are the Laplace transform of its output and input.

The transfer function is defined as

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\sum_{j=0}^{M} b_j s^j}{\sum_{k=0}^{N} a_k s^k}. \tag{4.6}$$

- N is the order of the system (i.e. the order of the pole polynomial)
- $N - M$ is the relative order
- A system $G(s)$ is defined as *proper* if $\lim_{\omega \to \infty} G(j\omega) = D \neq \infty$, which requires $M < N$. It means that it can be physically realized.

4.1.4 State-space description

If transfer functions are a very useful tool (for example to represent the Bode diagram of a function, or easily combine successive systems), computing their outputs is not very practical. Another representation is often used, with the help of a inner state variable $x(t)$:

$$\begin{cases} \frac{d\underline{x}}{dt}(t) &= \underline{\underline{A}}\,\underline{x}(t) + \underline{\underline{B}}\,\underline{u}(t) \\ \underline{y}(t) &= \underline{\underline{C}}\,\underline{x}(t) + \underline{\underline{D}}\,\underline{u}(t) \end{cases}, \qquad \underline{x}(0) = \underline{x}_0 \tag{4.7}$$

where $\underline{\underline{A}} \in \mathbb{R}^{n \times n}$, $\underline{\underline{B}} \in \mathbb{R}^{n \times q}$, $\underline{\underline{C}} \in \mathbb{R}^{p \times n}$ and $\underline{\underline{D}} \in \mathbb{R}^{p \times q}$ can be defined from the transfer function equation (4.6) or directly from the differential equation model equation (4.6). This will not be described here, as Python and MATLAB libraries provide functions to directly calculate the matrices. A more thorough reference can be found in [6].

This representation is however a continuous representation, that cannot be used directly on a DSP. It must be first be transformed as follow [6], for a sampling rate $F_s = \frac{1}{T_s}$

$$\underline{\underline{\tilde{A}}} = e^{\underline{\underline{A}} T_s}, \quad \underline{\underline{\tilde{B}}} = \int_0^{T_s} e^{\underline{\underline{A}}(T_s - \tau)} \underline{\underline{B}} d\tau, \quad \underline{\underline{\tilde{C}}} = \underline{\underline{C}}, \quad \underline{\underline{\tilde{D}}} = \underline{\underline{D}} \tag{4.8}$$

and then

$$\begin{cases} \underline{x}[k+1] &= \underline{\underline{\tilde{A}}}\,\underline{x}[k] + \underline{\underline{\tilde{B}}}\,\underline{u}[k] \\ \underline{y}[k] &= \underline{\underline{\tilde{C}}}\,\underline{x}[k] + \underline{\underline{\tilde{D}}}\,\underline{u}[k] \end{cases}, \qquad \underline{x}[0] = \underline{x}_0. \tag{4.9}$$

Equation (4.9) is mostly used in the simulation algorithm.

4.1.5 Transfer function estimation

In some cases, modeling the system with differential equations is either too complex or provides too simplistic results. The system can instead be used as a black box, and it's frequency response measured.

Using Fourier transform The Laplace transform evaluated in $s = j\omega$ results in the Fourier transform. Therefore, as shown in equation (4.6), using an input with enough frequency content should allow for good estimation of the transfer function:

$$\tilde{G}(j\omega) = \frac{Y(j\omega)}{U(j\omega)}. \tag{4.10}$$

However this naive method is not really used in practice, as the measurement noises and artifacts in the output would have strong impact on the estimation.

Spectral density A better method is to use the spectral density and the cross-spectral density.

In the time domain, the output can be written as a function of the input by using a impulse response $g(t) = \mathcal{L}^{-1}\{G(s)\}$

$$y(t) = g * u\,(t) = \int_0^{+\infty} g(\tau)u(t-\tau)d\tau \tag{4.11}$$

The cross-correlation between the input and the output is given by

$$\begin{aligned} R_{uy}(\tau) &= E\left\{u(t)y(t+\tau)\right\} \\ &= E\left\{u(t)\int_0^{+\infty} g(v)u(t+\tau-v)dv\right\} \\ &= ... \\ &= \int_0^{+\infty} g(v)R_{uu}(\tau-v)dv \\ &= g * R_{uu}\,(\tau) \end{aligned} \tag{4.12}$$

where R_{uu} is the autocorrelation of the input. Transforming this equation to the frequency domain yields

$$S_{uy}(j\omega) = G(j\omega)\cdot S_{uu}(j\omega) \iff G(j\omega) = \frac{S_{uy}(j\omega)}{S_{uu}(j\omega)} \tag{4.13}$$

where S_{uy} is the cross-spectral density and S_{uu} the spectral density of the input.

This provides better results in practice because only the relevant information of the signal is used: the transfer function estimation is less dependent on the measurement noise and other possible artifacts.

This type of identification provides a good first approximation of the system. In this case, the practical process is described as a black box, since no prior knowledge is used. This can be refined by adding some theoretical information to constrain the solution. This will not be conducted here, as it would require more time on this specific topic.

Correct input It is important to choose a good input signal. It must contain enough harmonic components to provide an estimation of the full spectrum. Literature often propose a pink noise or a sine-sweep.

A pink noise is a signal which power spectrum is decreasing like $1/\omega$. The spectrum is thus fully represented, with high frequencies having less impact and therefore being less likely to harm the system. In a system like the storage ring, using a noise as input is however not a good idea.

A sine sweep is a sine in which the frequency increases with the time, so that all frequency between the up and down limits $f_{\max}$ and $f_{\min}$ are represented. A linear increase can be obtained by setting

$$f(t) = 2f_{\min} \cdot t + \frac{f_{\max} - f_{\min}}{2 \cdot t_{\max}} t^2. \tag{4.14}$$

The input is then

$$u(t) = A \sin(2\pi f(t)). \tag{4.15}$$

4.2 System identification

The first step in the model is to represent the storage ring. This could be done analytically, by writing the magnet equations and deriving the full differential equations that bind the correction to orbit displacement. To do this, the different types of components all around the orbit, with some non-linear, some coupled and complex behaviors, need to be studied and put in equations.

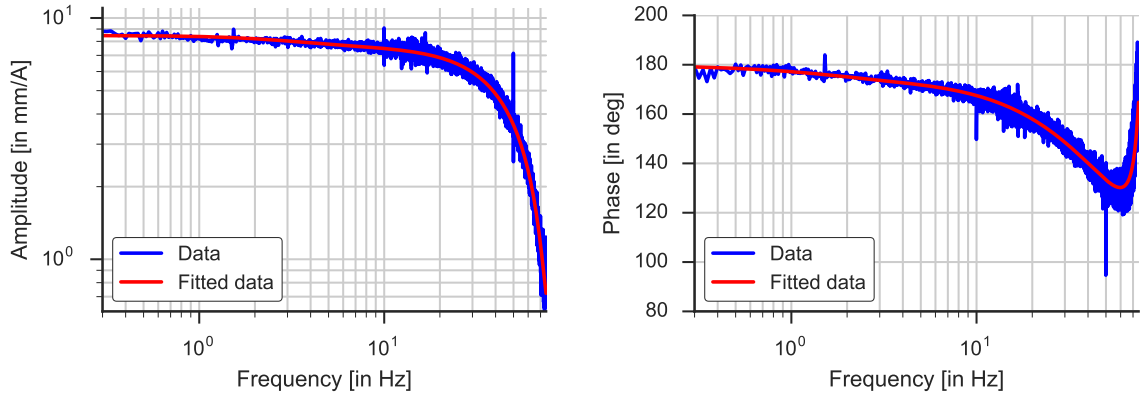
As for the defining the response matrix, it is a difficult problem, with no perfect solution: some instrument optics are unknown and their positioning is never perfectly known.

Instead a system identification is conducted. The system is the storage ring, with the differential orbit as the output, and correction magnet amplitude as input. It is assumed that the system can be modeled in first approximation as a SISO system, generalized as a MIMO one by pre-multiplying it by the response matrix (which is a mere gain):

$$\Delta \underline{X}(t) = \underline{\underline{S}} \cdot [g * \Delta \underline{\theta}(t)]. \tag{4.16}$$

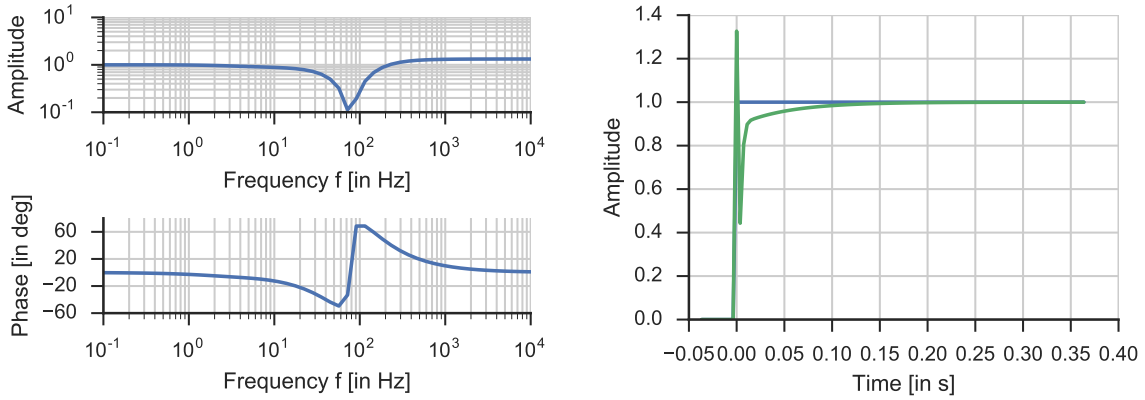
A sine sweep was designed to cover from $f_{\min} = 0$ to $f_{\max} = F_s/2 = 75\,\text{Hz}$ and used as the corrector magnets input (one by one).

By using the (cross-)spectral density of the input and output signal, an experimental estimation of the transfer function is provided. The estimation is then fitted with a MATLAB



(a) Amplitude (b) Phase

Figure 4.1: Results of the vector fitting with vfit3 (blue: raw data, red: fitted transfer function)



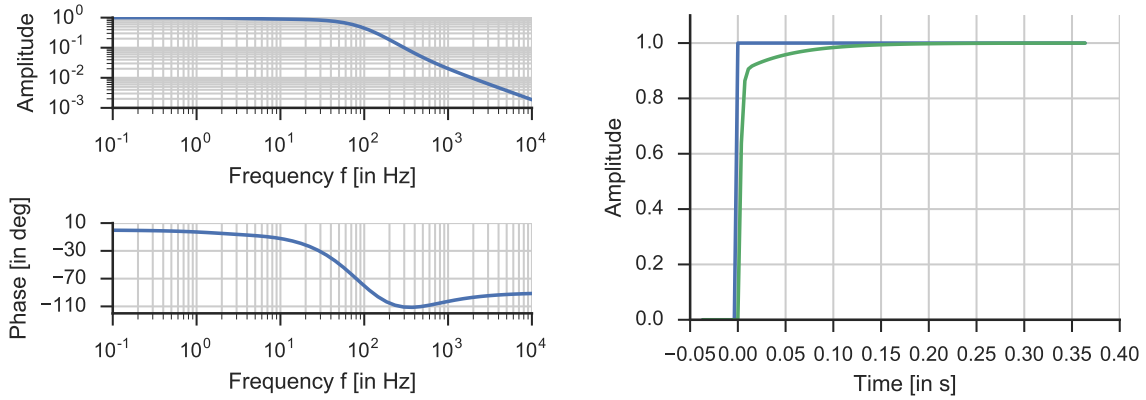
(a) Frequency response (b) Step response

Figure 4.2: Normalized transfer function

routine implementing the Fast Relaxed Vector Fitting [7, 8, 9, 10] and provides the continuous time state-space matrices $\underline{\underline{A}}$, $\underline{\underline{B}}$, $\underline{\underline{C}}$ and $\underline{\underline{D}}$. The result is shown figure 4.1.

It appeared that after converting the system to a transfer function, the numerator was complex. Taking only the real part of each coefficient provided a solution that still fits the experimental data. In order to obtain a transfer function usable with the response matrix, it was normalized so that its ground gain was 1.

Figure 4.1 shows however only the frequency domain below 75 Hz. In the first simulations, it appeared that the step response presented a large spike which was hardly physically possible (see figure 4.2). To correct this, the numerator highest degree coefficient was set to 0, which led to the result given in figure 4.3.



(a) Frequency response

(b) Step response

Figure 4.3: Normalized, zeros-corrected transfer function

Finally the transfer function estimation is

$$G(s) = \frac{117.322203\,s^2 + 319144.577\,s + 6801819.36}{s^3 + 1171.05630\,s^2 + 375569.975\,s + 6801819.36} \tag{4.17}$$

The frequency response given in figure 4.3 corresponds to what was expected: an amplitude that decreases with frequencies, without amplification domain.

This was validated by simulation (see below), where the current situation was reproduced, with similar parameters.

4.3 Simulation model

In order to represent at best the environment, a simulation needs to take into account computation delays and that the correction happens at a given sampled time 150 Hz), whereas other systems perform in continuous time. To achieve that, the simulation is played step by step, at 150 Hz for the correction process and 1.5 kHz for the continuous time.

Figure 4.4 represents the block diagram of the environment, H_{ring} being the identified transfer function of the storage ring, and the low pass filter representing the averaging of orbit values in the BPM recording. An initial correction $\underline{\theta_0}$, which is normally provided by a punctual correction process (the slow orbit feedback or SOFB), is needed to ensure the stability of the orbit.

4.4 Simulation results

Several scenarios were designed to validate the simulation and propose some enhancement to current correction. The delay of the computation is approximated by $\delta t = 3\,\text{ms}$.

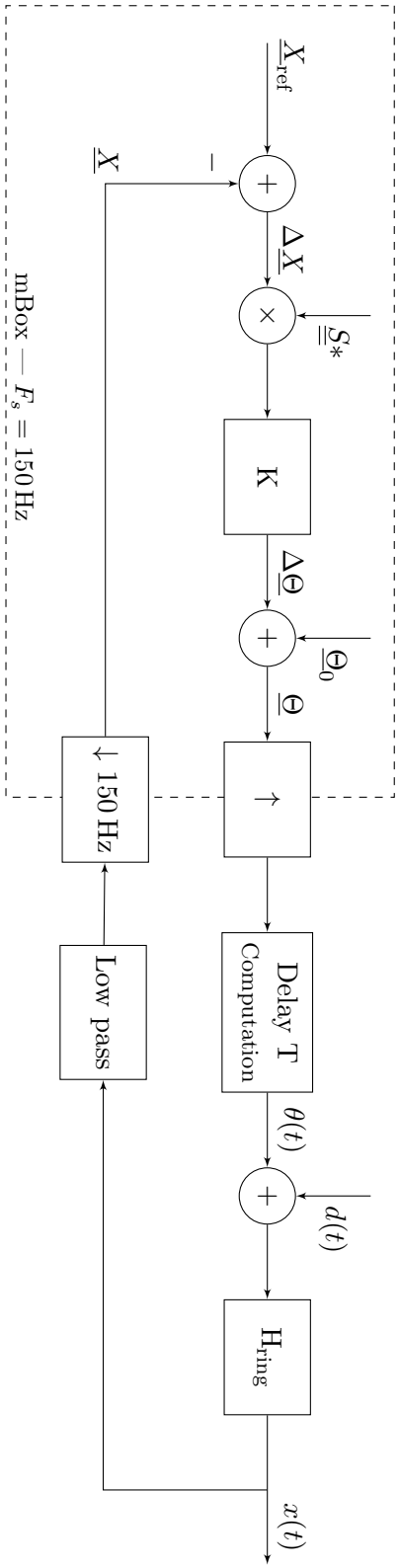
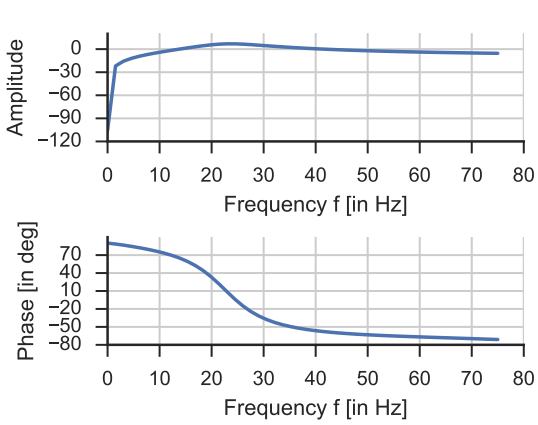
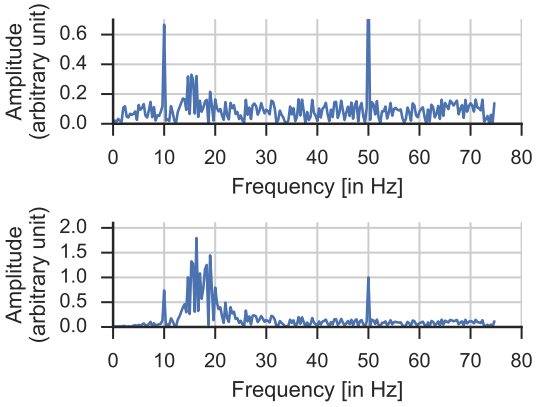


Figure 4.4: The full model, K being the corrector to define

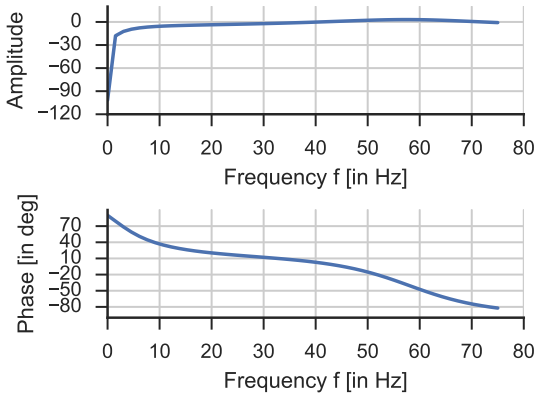


(a) Bode diagram of the closed-loop

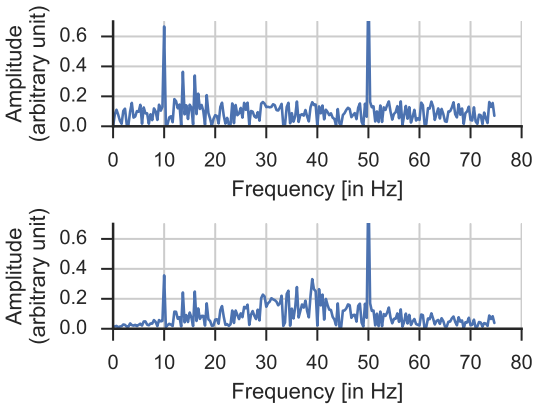


(b) Simulation, top: perturbation d, bottom: output x

Figure 4.5: Frequency response for K = PID(0, $0.8 \cdot F_s$, 0)



(a) Bode diagram of the closed-loop



(b) Simulation, top: perturbation d, bottom: output x

Figure 4.6: Frequency response for a better PID

4.4.1 PID correction

To represent the current configuration in BESSY II, and validate the model, the PID corrector presented in section 3.4 was implemented as a pure integrator of amplitude $0.8 \cdot \mathrm{F_s}$. The closed-loop system frequency response would be as given in figure 4.5a. As in the reality, frequencies below 10 Hz are damped, and are slightly amplified above. Applying the simulation step by step with a plausible perturbation leads to the result presented in figure 4.5b.

The PID can then be slightly enhanced by finding better coefficients, for example with K = PID(0.9, $0.5 \cdot \mathrm{F_s}$, $0.15/\mathrm{F_s}$), which result can be seen in figures 4.6a and 4.6b. It can be seen there that more frequencies are slightly damped and that the amplified area is smaller, with less gain, providing better results.

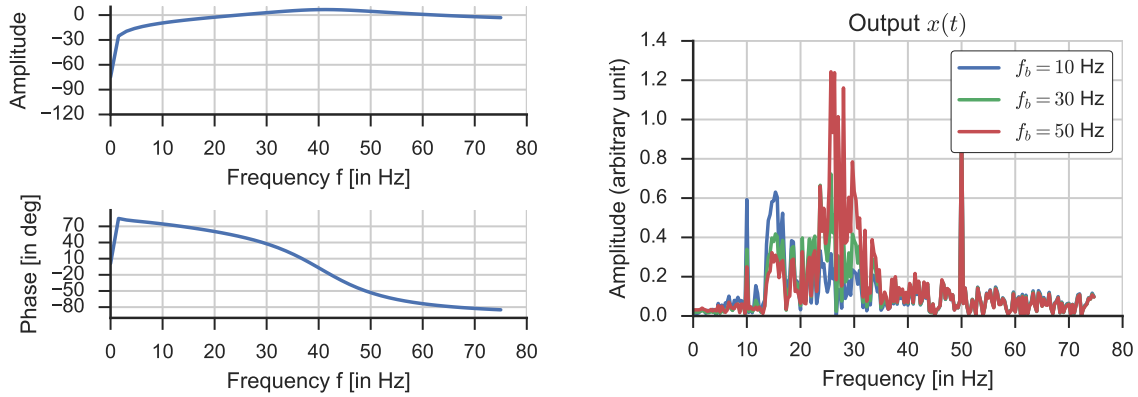
4.4.2 Optimal correction

Following the design rules defined in [11], one can synthesize an $\mathcal{H}_\infty$-corrector, which is an optimal corrector obtained by optimizing the closed and opened loop over the ∞-norm[1] with given parameters. The Robust Control Toolbox from MATLAB was used for this effect. The MATLAB script given below is an adapted copy of what can be found in Skogestad [11], page 60. The computation delay is modeled with the Padé approximation of first order [12] provided by MATLAB and added to the system, so that the corrector takes it into account.

```
% The robust control toolbox is needed.

% Load ring transfert function
[G_num, G_den] = load(G)

% Create delay
delay = 0.003;
[dnum, dden] = pade(delay, 1);

G = nd2sys(conv(G_num,d_num),conv(G_den,d_den),1);

% Define parameters
fb = 30  % Bandwidth
wb = fb*2*pi;
M = 1.001;  % Lim sup (high freq.)
Am = 10^-4;  % Lim sup (low freq.)

% Create weights
Wp = nd2sys([1/M wb], [1 wb*Am]);
Wu = 1;

% Define the system
systemnames = 'G Wp Wu';
inputvar = '[r(1) ; u(1)]';
outputvar = '[Wp; Wu; r-G]';
input_to_G = '[u]';
input_to_Wp = '[r-G]';
input_to_Wu = '[u]';
sysoutname = 'P';
cleanupsysic = 'yes';
sysic;

% Compute corrector
nmeas=1; nu=1; gmn=0.5; gmx=20; tol=0.001;
[khinf,ghinf,gopt] = hinfsyn(P,nmeas,nu,gmn,gmx,tol);
```

This provides a corrector, whose frequency response can be seen in figure 4.7a: it is very efficient in low frequencies, however it has however an amplification area that can be a problem. Correctors with different bandwidth were synthesized. The simulation is given in figure 4.7b. The results are strangely enough always less efficient than the proposed PID (in figure 4.6a).

[1] The ∞-norm is defined for vectors as $||\underline{x}||_\infty = \max_i |x_i|$



(a) Bode diagram of the closed-loop, $f_b = 50\,\text{Hz}$ (b) Simulation (several bandwidths)

Figure 4.7: Frequency response of $\mathcal{H}_\infty$-correctors

4.5 Summary

A simulation environment was built in order to be able to predict the influence of some parameters on the system and to closely study how to correct at best the orbit position.

A first system approximation was identified from the ring. From this was predicted that a more optimized PID corrector is possible, which only needs to be validated experimentally. The optimal correction line of research must be considered in more depth and, with a more ambitious perspective, the design of robust MIMO correctors would certainly provide better and more reliable results.

Chapter 5

Localization of orbit perturbations

A correction, as adapted as it can be, will never be perfect. Instead of dealing with the effects of the perturbation, its sources can first be investigated. When all possible sources are found and removed or isolated, a correction algorithm can be applied to the remaining perturbations, which will hopefully yield better results.

If no source is really obvious (e.g. a non-isolated transformer, the 50 Hz perturbation of the main power), the orbit itself can give some hints to localize it. One local perturbation affects indeed the whole revolution. It results in an oscillation across the orbit which, because of the closed orbit property, will dramatically change its angle at the position of the source (see figure 5.1). This position is termed *kick*.

Two types of perturbations will be described in this section:

- static perturbations, which can be for instance a malfunctioning magnet,
- harmonic perturbations, i.e. a perturbation at a given frequency, which can be for instance the 50 Hz magnetic field of the main power or the 10 Hz field due to a bad isolated power supply.

5.1 Static perturbation

5.1.1 Theoretical setting of the problem

The problem is described with the betatron phase variable

$$\Psi = \int_0^s \frac{d\sigma}{\beta(\sigma)}. \tag{5.1}$$

The spatial variable s is only used to have a connection between the result and the actual ring. The explanation will be led in the horizontal plane (with the x variable), but is also

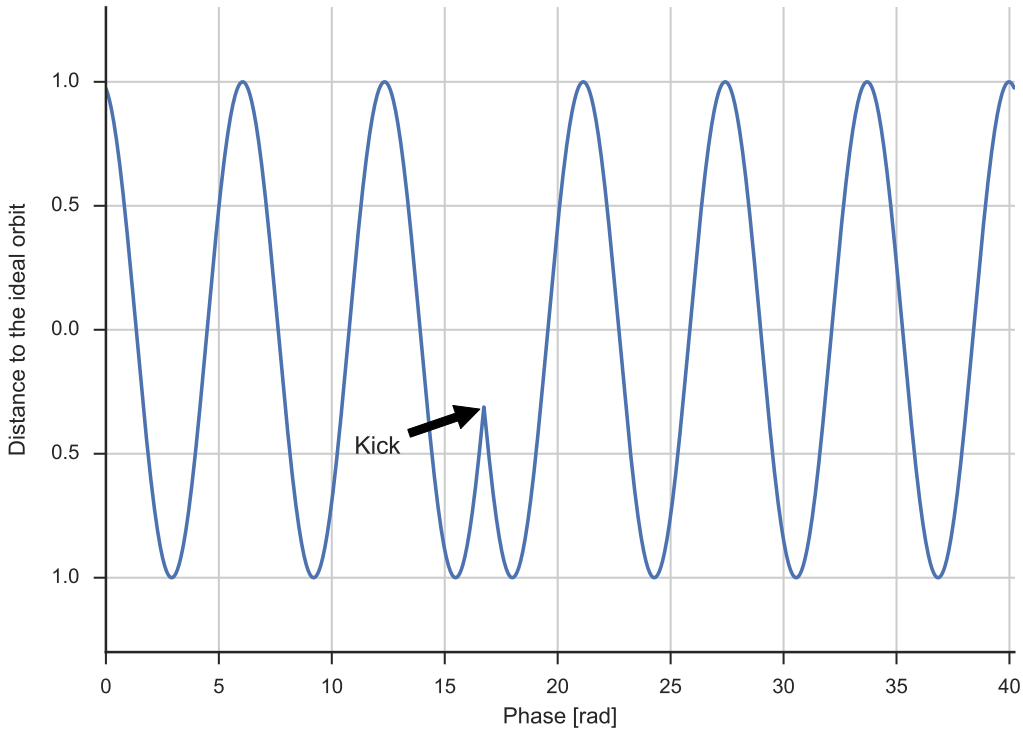


Figure 5.1: Example of kick in the orbit

valid in the vertical plane (y).

Let one kick be at a given phase $\Psi = \hat{\Psi}$. The orbit is modified and oscillates with a constant period of 2π rad. According to the closed orbit condition, because there is *only* one kick, the oscillation after the kick will be stable for one revolution. Furthermore, the orbit must be continuous on all points, thus at the kick position too. This is illustrated in figure 5.1.

Two revolutions are considered, in order to be sure to find one full revolution without a kick. Let $\Psi^{\text{ext}} \in [0, 4\pi Q]$ be this new phase (*ext* for extended). The phase $\hat{\Psi}$ where the kick happens is the one so that

$$\exists (b, c) \in \mathbb{R}^2 : \forall \Psi \in [\hat{\Psi}, \hat{\Psi} + 2\pi Q], \quad x(\Psi) = b \cos(\Psi + c) \tag{5.2}$$

where Q is the tune of the orbit (see equation (2.29)).

This problem has 3 unknowns which should be determined: $\hat{\Psi}, b, c$.

5.1.2 Practical setting

In order to know the position of the orbit, BPMs are used.

Only m BPMs are distributed around the orbit. The previous variable can thus be described in a vectorial form

$$\begin{cases} \underline{\Psi} = [\Psi_0, \Psi_1, ..., \Psi_{m-1}] \\ \underline{x} = [x_0, x_1, ..., x_{m-1}] \end{cases} \quad \text{and} \quad \begin{cases} \underline{\Psi}^{\text{ext}} = [\underline{\Psi}, \underline{\Psi} + 2\pi Q] \\ \underline{x}^{\text{ext}} = [\underline{x}, \underline{x}] \end{cases} \tag{5.3}$$

5.1.3 Solving the problem

The problem is solved in two steps: first, the sine that fits at best the orbit is determined, and second, the position of the kick verifying the closed orbit condition (or continuity condition) is found.

An algorithm is designed to find a sine over a revolution, beginning at each BPM and keeping the one that fits at best:

$$\forall\, k \in [0, m-1], \quad \begin{cases} \underline{\Psi}^k = \left[\Psi_k^{\text{ext}}, \Psi_{k+1}^{\text{ext}}, \cdots, \Psi_{k+m-1}^{\text{ext}}\right] \\ \underline{x}^k = \left[x_k^{\text{ext}}, x_{k+1}^{\text{ext}}, \cdots, x_{k+m-1}^{\text{ext}}\right] \\ \underline{\tilde{x}} = \texttt{fit\_sine}\left(\underline{x}^k, \underline{\Psi}^k\right) \end{cases} \tag{5.4}$$

It is then defined

$$k_0 = \underset{k\in[0,m-1]}{\text{argmin}} \{||\underline{\tilde{x}} - \underline{x}^k||_2\} \tag{5.5}$$

If there where no noise in the signal, and if the number m of BPMs was infinite, then the kick would be exactly at Ψ_{k_0}. However in the real case (with noise), it can only be said that the kick is around Ψ_{k_0}, and the closest sine is $\tilde{x}(\Psi) = b\sin(\Psi + c)$.

To find the exact position of the kick, the property of closed orbit is used: the orbit must be continuous also at the kick phase, which means that $\hat{\Psi}$ is the solution of

$$b\cos(\Psi + c) = b\cos(\Psi + c + 2\pi Q), \quad \text{with } \Psi \in [\Psi_{k_0} - A, \Psi_{k_0} + A], A > 0 \tag{5.6}$$

$$\begin{cases} \Psi + c & \equiv \Psi + c + 2\pi Q \pmod{2\pi} \\ \Psi + c & \equiv -(\Psi + c + 2\pi Q) \pmod{2\pi} \end{cases}$$

$$\iff \begin{cases} 2\pi Q & \equiv 0 \pmod{\pi} \quad \textit{(Never true)} \\ \Psi & \equiv -c - \pi Q \pmod{\pi} \end{cases}$$

The only possible solutions have the form

$$\Psi \equiv -c - \pi Q \pmod{\pi}. \tag{5.7}$$

As the kick is the closest solution to Ψ_{k_0}, a constant K is searched so that

$$\begin{aligned} -c - \pi Q + K\pi \leq \Psi_{k_0} \leq -c - \pi Q + (K+1)\pi \\ \iff \frac{\Psi_{k_0} + c}{\pi} + Q \leq K \leq \frac{\Psi_{k_0} + c}{\pi} + Q + 1 \\ \iff K = \left\lfloor \frac{\Psi_{k_0} + c}{\pi} + Q \right\rfloor \end{aligned} \tag{5.8}$$

and the two remaining possible values are

$$\{-c - \pi Q + K\pi, \quad -c - \pi Q + (K+1)\pi\} \tag{5.9}$$

the kick being chosen as the closest one from Ψ_{k_0}.

5.1.4 Finding the good sinusoidal

Several methods are possible to find the best matching sine, for example by using:

- a pseudo-inversion
- a scalar-product with a sine (resp. a cosine)

Pseudo-inversion The problem can be set as a linear equation problem as follow.

$$\begin{aligned} &\forall k \in [0, m-1], \tilde{x}(\Psi_k) = a_1 \cos(\Psi_k) + a_2 \sin(\Psi_k) + a \\ \implies &\begin{pmatrix} 1 & \cos(\Psi_0) & \sin(\Psi_0) \\ 1 & \cos(\Psi_1) & \sin(\Psi_1) \\ \vdots & \vdots & \vdots \\ 1 & \cos(\Psi_{m-1}) & \sin(\Psi_{m-1}) \end{pmatrix} \begin{pmatrix} a \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} x_0 \\ x_2 \\ \vdots \\ x_{m-1} \end{pmatrix} \\ \implies &\begin{pmatrix} a \\ a_1 \\ a_2 \end{pmatrix} = \text{pseudo\_inv} \begin{pmatrix} 1 & \cos(\Psi_0) & \sin(\Psi_0) \\ 1 & \cos(\Psi_1) & \sin(\Psi_1) \\ \vdots & \vdots & \vdots \\ 1 & \cos(\Psi_{m-1}) & \sin(\Psi_{m-1}) \end{pmatrix} \begin{pmatrix} x_0 \\ x_2 \\ \vdots \\ x_{m-1} \end{pmatrix} \end{aligned} \tag{5.10}$$

The pseudo inverse is calculated in `Matlab` with

```
a = M\x
```

and in `Python` with the least-squares method

```
a = numpy.linalg.lstsq(M, x).
```

Projection on cosine/sine planes Since the orbit is expected to be written as

$$x(\Psi) = a + a_1 \cos(\Psi) + a_2 \sin(\Psi)$$

it can also be described as

$$x(\Psi) = \langle x\,, 1\rangle + \langle x\,, \cos\rangle \cos(\Psi) + \langle x\,, \sin\rangle \sin(\Psi) \tag{5.11}$$

with $\langle f\,, g\rangle$ being the scalar product for real functions: $\int_T f(t)g(t)dt$.

In the numerical case, the scalar product is approximated by its vectorial counterpart by

$$\begin{aligned} \langle \underline{f}\,, \underline{g}\rangle : \quad & \mathcal{R}^n \times \mathcal{R}^n \longrightarrow \mathcal{R} \\ & (\underline{f}, \underline{g}) \quad \longmapsto \quad \frac{1}{n}\sum_{k=0}^{n-1} f_i g_i \end{aligned}$$

This however provides less good results, as $\sin(\Psi)$ and $\cos(\Psi)$ are numerically orthogonal only if the frequency is a multiple of the sampling frequency divided by the number of samples (F_s/N) which cannot be always verified.

Coefficient format By defining $b = \sqrt{a_1^2 + a_2^2}$ and $c = -\arg(a_1 + ja_2)$ the previous formulas can be written

$$\tilde{x}(\Psi) = a + a_1 \cos(\Psi) + a_2 \sin(\Psi) = a + b \sin(\Psi + c).$$

5.2 Localizing multiple sources

A step further would be to localize several sources. To do this, one must be able to describe the orbit as an arbitrary piecewise sinusoid function, which is equivalent to the following requirements:

1. Find the number of perturbation sources (or piece junctions)
2. Find the position of theses perturbation sources (or where the junctions are)

The only work in this direction resulted in a brute-force algorithm which would test every single possibility and finally keep the one with the lowest RMS. Further mathematical and algorithmic literature must be studied before deepening this idea.

5.3 Harmonic perturbations

Some perturbations can be purely harmonic. The example of the 50 Hz field generated by the main power is an obvious one that cannot be easily isolated.

As shown in figure 3.5, the correction leaves some spikes in the spectrum, which is to be avoided. Hopefully some of them might be localized and removed or isolated, dealing again with the cause of the perturbation instead of its symptom.

Because the storage ring is considered in a first approximation as a linear system, the source of a harmonic perturbation can be localized as a harmonic disturbance with the same frequency. The very same algorithm can thus be used on a pseudo-orbit extracted from the harmonic of interest.

The harmonic orbit is calculated the same way as in section 3.5, equation (3.15), providing a complex vector $\Delta\underline{X}_f$.

If the perturbation is unique, then it is expected to be writable as $d(t) = A_d \cos(2\pi f t + \phi)$. Because the phase shift is the same for all BPMs, the orbit time signal for the ith BPM can be written as

$$x_i(t) = \hat{X}_{f,i} \cos(2\pi f t), \qquad \hat{X}_{f,i} \in \mathbb{R}. \tag{5.12}$$

All complex amplitude must thus exactly describe the same sinusoid of frequency f and phase α_0, only with the amplitude being different at each BPM. The complex vector $\Delta\underline{X}_f$ can thus be fully described by their amplitudes.

However, since the signal also contains noise and since that the perturbation *might* not be unique, it is rotated until the vector of sine (or imaginary part) is minimal, or zero in an ideal case.

$$\begin{cases} \alpha_0 = \underset{\alpha \in [0,2\pi[}{\operatorname{argmin}} \{\mathcal{I}m\left(\Delta\underline{X}_f \cdot e^{-j\alpha}\right)\} \\ \underline{\hat{X}}_f = \mathcal{R}e\left(\Delta\underline{X}_f \cdot e^{-j\alpha_0}\right) \end{cases} \tag{5.13}$$

This way only one perturbation is considered: the one which contributes the most to the signal.

The new signal $\underline{\hat{X}}_f$ can be used as a differential orbit signal. The kick position is then calculated with the method described in the static case, section 5.1.

Note — To achieve the phase optimization given in equation (5.13), the Karhunen–Loève transform (or principal component analysis) can be used [13]. A description of the algorithm is given in appendix A.2. If the results are exactly the same, this allows the problem to be solved within a broader theoretical setting. The goal is not anymore to see the sine part vanish, but to deal with the perturbation space in which the distortion is the largest. In theory, dealing with each principal component would allow to find the position of each perturbation source.

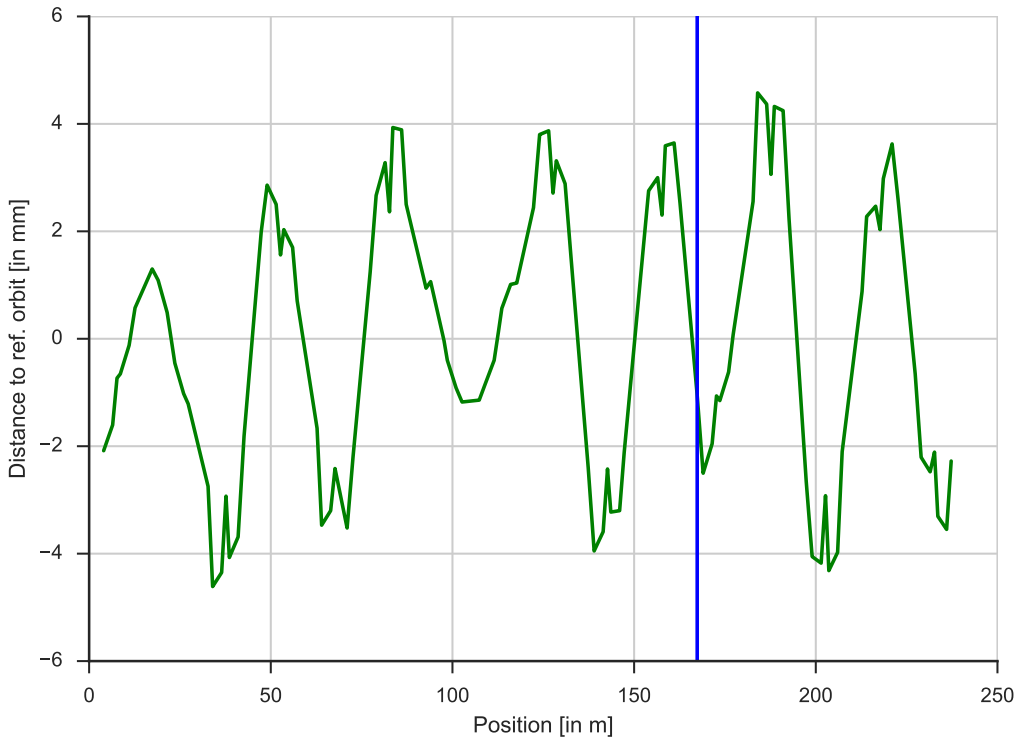


Figure 5.2: Orbit with unique static perturbation

5.4 Experimental results

5.4.1 Artificial static perturbation

The localization algorithm described above is used on an artificial experiment. All correctors are set to 0, except one which is randomly chosen and given an arbitrary amplitude. The orbit is generated by using the response matrix (see section 3.3.3) and is represented by the green curve on figure 5.2 (the final localization is in blue).

The localization algorithm iterations are shown in figure 5.3. The top graph presents in blue the RMS errors between the orbit and a generated sine while setting the kick at each BPM. In green and red are respectively the best fitting amplitudes and phases for each iteration. The convergence is visible and verifies that there is very little risk of being far from the position of the perturbation. However the region where the RMS reaches its smallest value is very flat and thus not very precise.

What is worth remarking is that the amplitude and phase vary quite continually, meaning that choosing the wrong position to set the sinus parameters is not too harmful. This validates the idea of first setting the sinusoid parameters and then finding the exact position of the

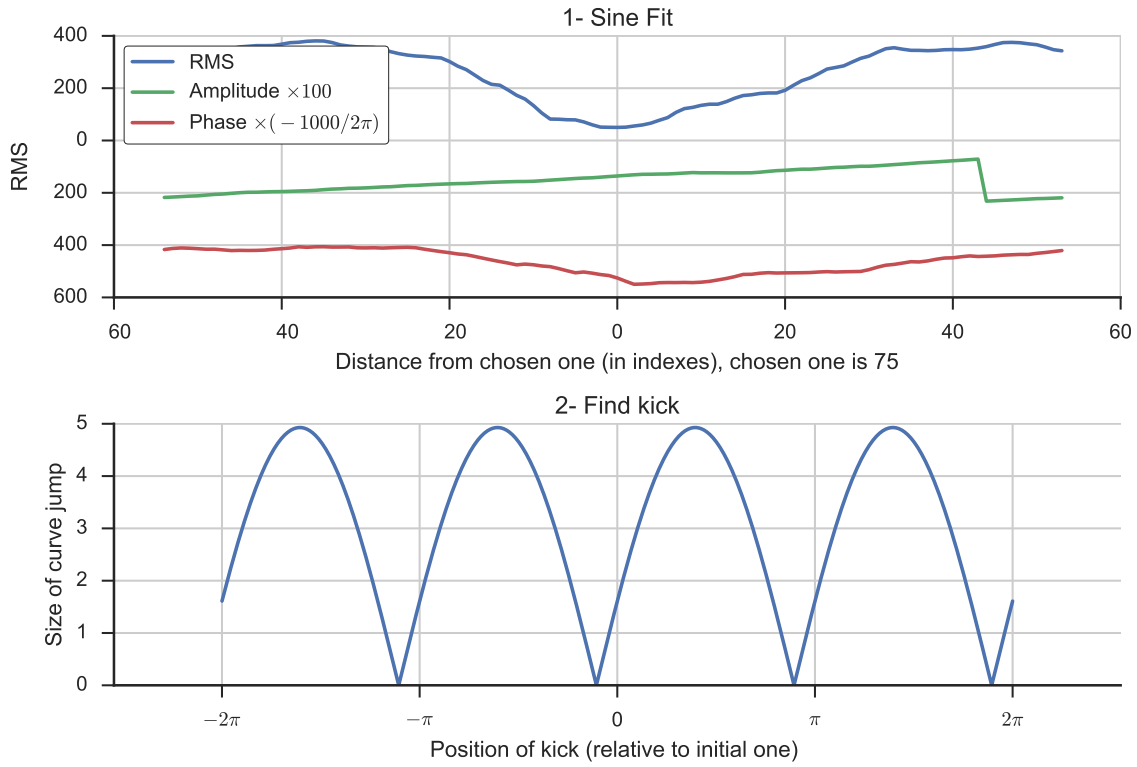


Figure 5.3: Error (RMS) and parameters curve of the localization algorithm

kick.

On the second graph (figure 5.3, below) is represented the jump done by the curve left and right from the BPM chosen to set the sinusoid parameters. In the algorithm, the first minimum of the curve is defined as the kick position.

Finally the orbit and the sinusoid constructed by the algorithm can be seen on figure 5.4. They are quite overlapping, and the closed orbit condition is verified.

The same artificial experiment was conducted by setting each corrector separately to an arbitrary value and trying to localize it back. Figure 5.5 shows the localization errors. It appears that the localization error in such an environment is quite precise in the vertical direction ($d \approx 2\,\mathrm{m}$ on a ring of 240 m), except around the 11th corrector. In this area, during the measurement of the response matrix used to generate the artificial orbit, a 7 Tesla wiggler (a device for producing intense X-rays from the electron beam) was activated; this means that it has a significant impact on the orbit optics and is therefore likely to degrade the localization process. A real test, directly on the storage ring, would certainly provide better results at this position, but reduce the overall precision because of measurement noises. In the horizontal direction, the results are less accurate because the number of available points by oscillation is lower. The horizontal tune is approximatively 17.85 whereas the vertical one

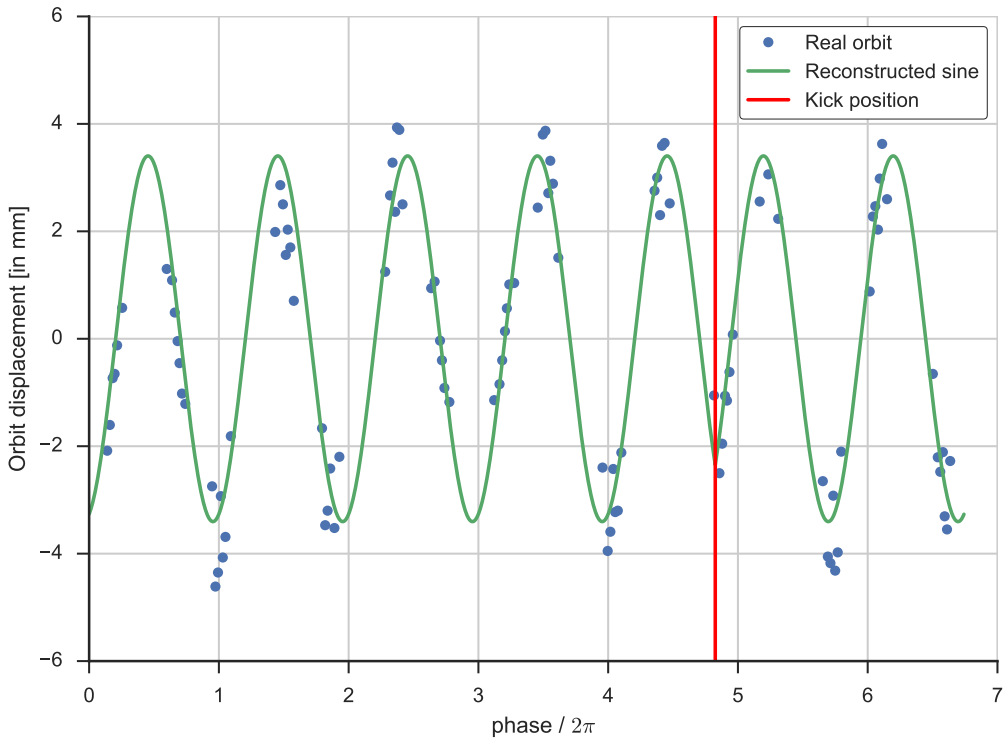


Figure 5.4: Reconstructed sinusoid and orbit

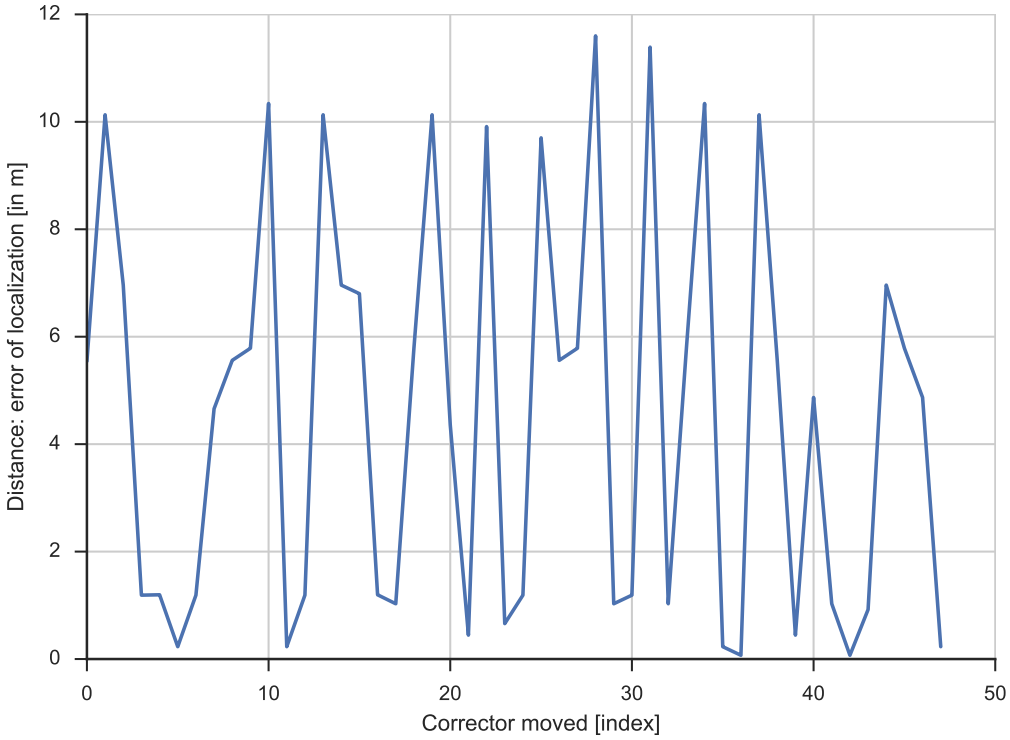
is 6.74, while the number of BPMs is identical in both directions.

Even by taking the worst localization error, it reduces the source position to less than 1/10 of the orbit circumference, which is already a non-negligible gain in the perturbation source removal process.

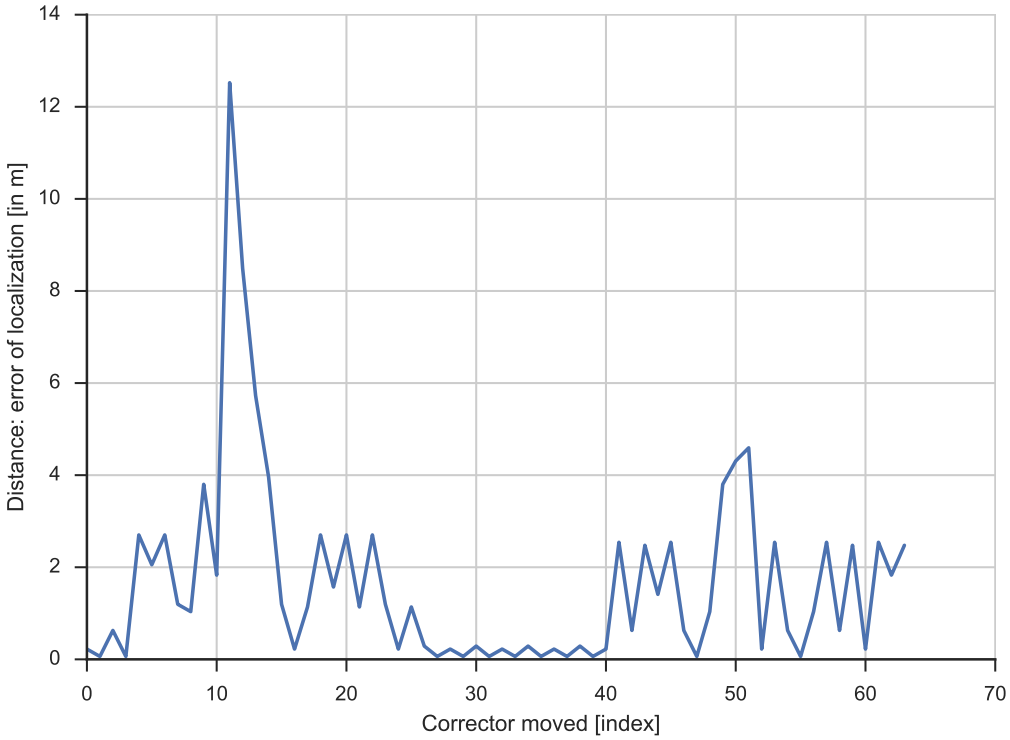
5.4.2 10 Hertz harmonic perturbation

A similar simulation was conducted for the harmonic simulation. Instead of setting the selected corrector magnet to an arbitrary amplitude, it is set to a sine with an arbitrary phase (in this case, random between 0 and 2π rad), sampled at $F_s = 150$ Hz. Every sample vector is then multiplied with the response matrix, which results in an orbit sample. This orbit can then be used as the input of the algorithm presented in section 5.3.

The localization results are exactly the same as in the static case, provided that enough samples are used (for example, 10 s at 150 Hz, for frequencies up to 70 Hz) and that the added noise is not too high.



(a) Horizontal



(b) Vertical

Figure 5.5: Localization error for all correctors

5.5 Summary

Instead of trying to correct some perturbations, isolating or removing their source would be very appealing. With the described algorithms, one can expect to reduce the potential zone to up to 2 m (which is less than 1% of the ring circumference) in perfect conditions, and in the worst case around 10 to 20 m (or around 10% of the circumference). With such a reduced area, it is very conceivable to determine the actual source, either with a mobile magnetometer or simply by observing the suspicious devices.

Chapter 6

Conclusion

6.1 Master thesis summary

To provide BESSY II users a reliable light, the electrons' orbit in the storage ring must be finely controlled. To remove perturbations, a process based on multiple actors implements an inverse problem solving coupled with a PID corrector. After the process was modernized to be easily expandable, a dynamic correction was designed resulting in complete removal of the 10 Hz harmonic perturbation. The implementation of a simulation environment allowed for synthesizing and testing correctors. An improved PID and an optimal corrector were proposed to enhance the current correction process.

As some perturbations cannot or are difficult to correct, algorithms succeededin localizing their source with a precision of around 1% of the circumference if they affect orbit vertically, and around 10 to 20% if it does horizontally.

As all the tools have been developed in a very modular way and precisely documented, they can hopefully be reused and expanded in the future by the accelerator department, which was supervising this thesis.

6.2 Possible extension and future work

The duration of the master thesis being limited, some interesting developments were only slightly considered and not deepened.

- In this thesis, the system identification did not use any theoretical or physical input. To achieve better correction and control results, a more precise system analysis of the storage ring should be conducted, which would allow a more constrained and less noise sensitive identification.

- As the storage ring is a MIMO system (several corrector magnets, influencing the response of several BPMs) with coupling effect also between the horizontal and vertical axes, future work should go in the direction of synthesizing robust MIMO correctors.
- On the localization side, implementing an algorithm able to localize several perturbations seem to be a very challenging mathematical study.

Bibliography

[1] K. Wille, *The physics of particle accelerators: An introduction.* Oxford University Press, 2000.

[2] "Photon source Bessy II." [Online]. Available: https://www.helmholtz-berlin.de/quellen/bessy/index_en.html

[3] G. A. Decker, "Operational aspects of experimental accelerator physics," in *AIP Conf. Proc.*, vol. 249, 1992, p. 637.

[4] E. Plouviez, "Fast positional global feedback for storage rings," in *Proc. DIPAC*, 1999, pp. 7–11.

[5] J. Li, C. Diao, W. Li, G. Liu, Z. Liu, and B. Sun, "Closed orbit correction of HLS storage ring," in *Proc. PAC*, 2001, pp. 1255–1258.

[6] King, *Struktur und Parameteridentifikation*, 2015.

[7] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by Vector Fitting," *IEEE Trans. Power Delivery*, vol. 14, no. 3, pp. 1052–1061, July 1999.

[8] B. Gustavsen, "Improving the pole relocating properties of vector fitting," *IEEE Trans. Power Delivery*, vol. 21, no. 3, pp. 1587–1592, July 2006.

[9] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. D. Zutter, "Macromodeling of multiport systems using a fast implementation of the vector fitting method," *IEEE Microwave and Wireless Components Letters*, vol. 18, no. 6, pp. 383–385, June 2008.

[10] "The vector fitting web site." [Online]. Available: https://www.sintef.no/projectweb/vectfit

[11] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design.* John Wiley & Sons, 2005.

[12] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Johns Hopkins University Press, Baltimore, 1989, pp. 557–558.

[13] R. Wang, *Introduction to orthogonal transforms: with applications in data processing and analysis.* Cambridge University Press, 2012, ch. 8.

Appendices

A

Mathematical appendices

A.1 Brightness and brilliance

In the following section, the θ subscript is either x or y.

The quality of the beam is described by the *brightness* and the *brilliance*.

Let F be the *flux* of photons, normalized to a beam current of 1 A

$$F = \frac{\text{photons}}{\text{s } 0.1\% \text{ BW A}}, \tag{A.1}$$

The brightness describes the angular divergence of the beam (given by $\sigma'_\theta = \sqrt{\frac{\varepsilon_\theta}{\beta_\theta}}$). It is defined as:

$$S = \frac{F}{2\pi\sigma'_x\sigma'_y} = \frac{F\sqrt{\beta_x\beta_y}}{2\pi\sqrt{\varepsilon_x\varepsilon_y}} = \frac{\text{photons}}{\text{s } 0.1\% \text{ BW mrad}^2 \text{ A}} \tag{A.2}$$

while the brilliance includes also the transverse dimensions ($\sigma_\theta = \sqrt{\varepsilon_\theta\beta_\theta}$):

$$B = \frac{F}{4\pi^2\sigma_x\sigma_y\sigma'_x\sigma'_y} = \frac{F}{4\pi^2\varepsilon_x\varepsilon_y} = \frac{\text{photons}}{\text{s } 0.1\% \text{ mm}^2 \text{ BW mrad}^2 \text{ A}} \tag{A.3}$$

These definitions vary in literature. These are taken from the book of K. Wille [1], and apply to Gaussian-shaped electron beams. The invariant idea is that both values are determinated by the beam emittance σ: the design of the accelerators and the correction are aimed at obtaining the smallest emittance ε_θ as possible.

A.2 Principal components analysis – Karhunen-Loève transform (KLT)

Let $\underline{\underline{X}}$ be a signal with n dimensions.

$$\underline{\underline{X}} = \left[\begin{array}{c|c|c|c} \underline{X}_1 & \underline{X}_2 & \dots & \underline{X}_n \end{array}\right] \tag{A.4}$$

The covariance matrix is extracted

$$\underline{\underline{A}} = \text{covar}(X) \tag{A.5}$$

and diagonalized so that the eigenvalues are sorted in a decreasing order (the first one being the most significant). The covariance being symmetric, it is always diagonalizable in $\mathbb{R}$.

$$\underline{\underline{D}} = \underline{\underline{P}}^t \underline{\underline{A}} \underline{\underline{P}} \tag{A.6}$$

The principal components are the columns of

$$\underline{\underline{\hat{X}}} = \underline{\underline{P}}^t \underline{\underline{X}} \tag{A.7}$$

and in the first column of the matrix resides the most significant of theses principal components.

B

mBox++

The program that was handling the correction computation (see section 3.4.2) was written in MATLAB. It had 2 main drawbacks:

- it was not very fast. MATLAB being an interpreted language, it produces slower programs than a compiled one such as C or C++.
- it was difficult to extend. The code was written in a very functional way, making it hard to had debugging outputs, timers or other modules. If this can be done in MATLAB, it is however not the easiest programming language to do it.

The program was thus rethought and rewritten from scratch. It can be found at

https://github.com/ochurlaud/MSc_FOFB-mBox,

licensed with the GNU General Public License v2.

B.1 Specifications

B.1.1 Constrains

- mBox++ must work without any change in the current architecture and environment
- mBox++ must be easily maintainable and improved: modularized, encapsulated and namespaced code (e.g. object oriented) is needed (global scope must not be used too much to be able to trace variable evolutions)

B.1.2 Performance

- mBox++ must be able to provide the same results as the MATLAB version, and better in extended modes

- mBox++ must be at least as fast as the MATLAB version
- mBox++ must be robust (e.g. graceful exit when requested, no crash on error/exception)

B.1.3 Features

- mBox++ must have a *read only* mode where it computes the correction but has no influence on the environment
- mBox++ must provide an experimental mode (where the correction is handled by external scripts)
- mBox++ must be able to communicate with external programs and to share its data.
- Tools to debug/profile must be provided

B.2 Technology used

B.2.1 Language and paradigm used

Technology: C++, Standard Template Library (STL)

Rationale: Compiled language, thus faster than MATLAB or Python. It is higher level than C and thus more manageable. It supports and eases object oriented programming, which allows encapsulation and to organize the code in business objects (e.g. one class is called *Logger* and deals with the logs and debug strings, another acquire the data and is called *ADC.*.). Every needed types exist in the STL (vectors, strings, etc.) and it is a widely used library very well documented and time tested. The C++ 11 version is used as it provides several improvements compared to previous ones and is now available on most platforms.

Resources: http://www.cplusplus.com — http://en.cppreference.com

B.2.2 Algebra computation

Technology: Armadillo

Rationale: Open source C++ library whose API is close to MATLAB's and supports multi-thread calculation. It is a long tested technology (currently version 7.x) with a large and active scientific community.

Resources: http://arma.sourceforge.net

B.2.3 External communication

Technology: ZeroMQ[1]

Rationale: Multi-platform, multi-language and open source distributed messaging library. It can be used to share messages between Python, Matlab, C/C++. It is a long tested technology (currently version 4.x), with a large and active community

Resources: http://zeromq.org

B.2.4 Others

- The program should be able to execute interpreted scripts: it uses the official Python library in C to call Python script.
- To ease the localization of dependency and the compilation, the CMake (https://cmake.org) build tool is used.

B.3 The program: architecture and runtime scheme

Figure B.1 represents the UML[2] diagram of the classes that build mBox++.

The class `mBox` is the one from which everything begins. It creates the shared objects and deals with the state machine process. It owns a `Handler` which can be a `CorrectionHandler` if mBox++ runs in normal mode, or a `MeasureHandler` if it runs in experiment mode.

The actual mathematics (computation of the calculation) happens in the processors. If the handler is a `CorrectionHandler`, then the processor can be a `CorrectionProcessor` which does exactly what the MATLAB code did (see section 3.4) or a `Dynamic10HzCorrectionProcessor` which implements what was described in section 3.5.2. Otherwise, if the handler is a `MeasureHandler`, the processor is directly the Python script provided by the user.

A handler delegates the actions to objects it owns:

1. The `ADC` reads the data from the ADC reserved area in the RFM.
2. The handler reads the data from the `ADC` object, does some conversions (mostly from a integer representation to doubles) and provides the data to its processor that calculates the correction.
3. The correction is then converted to integers by the handler and provided to the `DAC` which writes it to the DAC reserved area in the RFM.

[1] ZeroMQ is often shortened as zmq or ZMQ

[2] Unified Modeling Language: modeling language intended to provide a standard way to visualize the design of a system (Wikipedia)

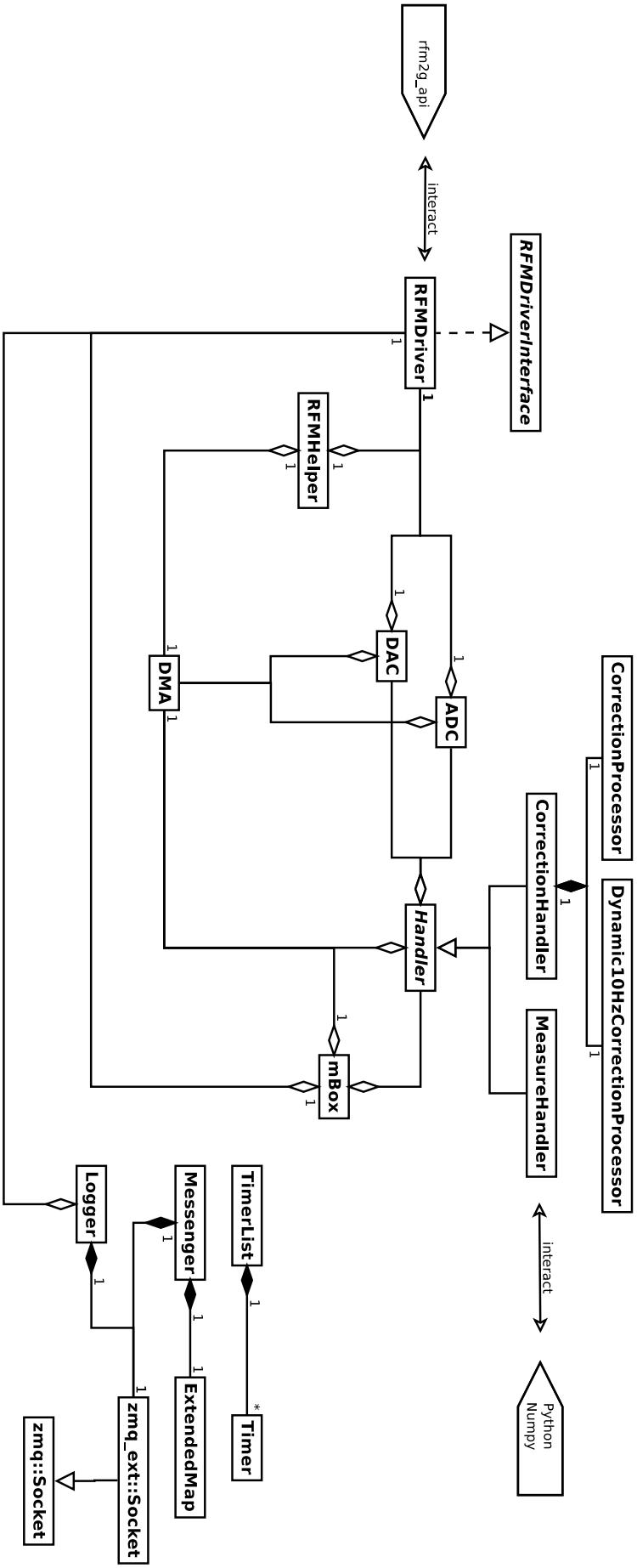


Figure B.1: Class diagram of the mBox++ program

To access the RFM, the C functions provided by the constructor are used. An interface `RFMDriverInterface` is implemented by the `RFMDriver` which provides a more object oriented version of the C API.

Note — In order to be able to test the program without having access to a RFM hardware, a second `RFMDriver` (accessible by compiling with the `-DDUMMY_DRIVER=ON` flag) implements the interface: it reads and writes in a 64 Mb binary file that represents the RFM. This can allow the creation of a simulation of the whole environment and expedient interaction with the program..

Three modules are statically constructed and globally defined (in their own namespaces):

- `TimerList` which is a "clever" list of `Timer`s, used to profile the parts of the code. A new or already existing timer can be started with

  ```
  TimingModule::addTimer("name_of_timer");
  ```

  stopped with

  ```
  TimingModule::timer("name_of_timer").stop();
  ```

  and its values shown (in this case once every 1000 times) with

  ```
  TimingModule::printAll(Timer::Unit::ms, 1000);
  ```

  The `TimimgModule` being global, it can be accessed from everywhere and thus some profiling can be done across objects and functions.
- `Logger` which is a class logging actions and errors. By defaults, errors only are shown to the stdout (standard output stream). Logs and errors are published on a given port (3333 by default) with ZeroMQ and can be read by subscribed services (the logs and errors can thus be output, saved in a file, published on a website stream or whatever crazy idea the user can have). It uses the ZeroMQ Publisher mode. A debug mode is also available (see `--help` command). It publishes incoming and outgoing values that can be used by other scripts or programs for display, scripting, value-checking... The full description of the uses are given in the Doxygen[3] documentation
- `Messenger` which is a class for exchanging information with outside scripts and programs. It is listening on a given port (3334 by default). It is using the ZeroMQ Router mode. Every time it receives a request, it processes it and either returns

[3]Popular documentation generator that extracts specific comments to produce PDF or HTML documentation. See http://www.stack.nl/~dimitri/doxygen/index.html

the value of the requested variable, or sets the given variable to the received value. The difference with the `Logger` is that the logger publishes values even if no one is listening and sends it when asked by the code, whereas the `Messenger` only answers queries. It is notably used in the harmonic correction process described in section 3.5.2: a python script calculates all the phases and amplitudes and sends the outcome, which are then used by `Dynamic10HzCorrectionProcessor`.

B.4 Project organization

The project is organized as followed:

- `/cmake` contains CMake modules to find other libraries and dependencies.
- `/doc` contains resources for the documentation and the generated documentation (which is not synchronized in the version control).
- `/experimentScripts` contains Python scripts that can be used as processor in experiment mode.
- `/python_tools` contains Python scripts and modules that can be used with the program: the helper for the harmonic correction is there (`tenHz.py`), a module to communicate with mBox++ through a binary file in dummy mode (`cbox.py`, `dummy_simul.py`) and various helpers to subscribe to the ZeroMQ streams, save data in a specified way...
- `/src` contains all the C++ code to be compiled. Its root contains the core of the project. The rest of the code is split in `/src/modules`, `/src/handlers` for a better readability.

Finally at the root of the project is `README.md` file giving the dependencies, how to compile, install and use mBox++. A Doxygen configuration file `doxygen.conf` let the documentation be generated by starting

```
$ doxygen doxygen.conf
```

B.5 Use

mBox++ cannot run if it does not receive the order from the cBox: it would otherwise hang in a waiting state. It can be used in normal mode

```
$ mbox --rw
```

in read only

```
$ mbox --ro
```

A full overview of the commands are given by

```
$ mbox --help


=== mbox (2015-2016) ===
Use:
mbox --ro
    Read only version: just reads the RFM and calculates
    the correction, don't write it back.
mbox --rw
    Read-write version: reads the RFM, calculates the
    correction and write it on the RFM.
mbox --experiment <FILENAME>
    Read-write version for experiments: read the file <FILENAME>
    to know which values to create.

Other arguments (to append):
--debug
    Print the logs on the the stderr.
--logport <PORT>
    Which port the log publisher should use.
--queryport <PORT>
    Which port the query messenger should use.
```

C

Search Kick

Search Kick was originally only able to find kicks in the orbit, in order to localize perturbation sources. It is currently closer to a Python orbit toolbox that allows translating dump data between various formats, localizing perturbations, and other small algorithms for inverting matrices or optimizing signals.

The library can be found at

https://github.com/ochurlaud/MSc_SearchKicks

licensed with the GNU General Public License v2.