

DAVID MICHALIK

A Model-based Optimal Control Approach for CERN's
AWAKE Electron Line Trajectory Correction Problem

MASTER'S THESIS IN CONTROL AND AUTOMATION

AALBORG UNIVERSITY

2021



AALBORG UNIVERSITY
STUDENT REPORT

**The Technical Faculty
of IT and Design
Control & Automation**
Fredrik Bajers Vej 7C
9000 Aalborg
www.es.aau.dk

Title:

A Model-based Optimal Control Approach
for CERN's AWAKE Electron Line Trajectory
Correction Problem

Project:

Master's Thesis Work at the European
Organization for Nuclear Research (CERN)
and Aalborg University (AAU)

Period of Project:

May 1st 2021 - August 31th 2021

Author:

David Michalik

Supervisors at AAU:

John Leth

jjl@es.aau.dk

Jan Dimon Bendtsen

dimon@es.aau.dk

ECTS points: 30

Printed exemplary: 0

Page number: 81

Ended: August 31, 2021

Abstract


In this project, an attempt is made to use model based optimal control for CERN's AWAKE electron beam line trajectory correction problem. CERN is constantly searching for advanced control methods for the new accelerators and this model based approach using an Linear Quadratic Regulator (LQR) in this manner has not been attempted before. The report therefore, begins with a description of CERN and the main accelerators and its experiments. The AWAKE experiment is described in detail, with focus on the electron beam line. Then, a thorough charged particle modelling chapter is presented with derivations through magnetic effects on the system all the way to linear equations of motion. Based on this, extensive state of the art research was done to analyse the control methods currently applied at CERN. Afterwards, the requirements and the control objective is defined. The system is then modelled in a simple approach to prove that model based control approaches can work on ultra-relativistic time-like systems. The model is then put into state space form and discretized. This yielded a bilinear system which is then controlled with an LQR controller. Simulations are then created which are based on a Reinforcement Learning simulation environment, which is modified to use the encoded physical properties of the system elements for the LQR simulation. In the end, the results show that the LQR feedback controller managed to drive the beam trajectory towards the reference. However, due to lack of access to physical hardware, it is difficult to assess real world performance.

Preface

The author would like to thank Jan Dimon Bendtsen and John Leth at Aalborg University for their supervision during this Master's Thesis. Their invaluable help and assistance under this challenging year of 2021 guided the work of this thesis project to the very end. Furthermore, the author would also like to thank Verena Kain at CERN for her contributions in outlining the topic of this project.

Aalborg University, 2021, August 31st.

David Michalik



David Michalik

dmicha16@student.aau.dk

Table of Contents

1	Introduction to the CERN complex	3
1.1	The Accelerator Complex	4
2	The AWAKE Collaboration	9
2.1	AWAKE History	9
2.2	Principles of Plasma Wakefield Acceleration	10
2.3	Layout and Structure of the Experiment	12
3	Modeling of Particle Motion	15
3.1	Transverse Dynamics	15
3.2	Magnets	20
3.3	Transverse equations of motion	26
4	Current Trajectory Control Methods for the Electron Beam Line	35
4.1	Trajectory Correction Control Problem	35
4.2	Multi-objective Hidden Cost Function Minimization	37
4.3	Singular Value Decomposition (SVD) Based Approach	39
4.4	Deep Reinforcement Learning (deep-RL)	41
4.5	Iterative LQR (iLQR)	43
4.6	Conclusion	44
5	Requirements, Limitations and Control Objective	45
5.1	Requirements	45
5.2	Limitations	46
5.3	Control Objective	46
6	System Model and Discretization	47
6.1	Proposed AWAKE Electron Beam Line System Model	47
6.2	EOMs for Individual Drift Spaces, Dipoles and Quadrupoles	48
6.3	State Space Model	50
6.4	Model Discretization	52
6.5	State Measurement via BPMs	55
7	Controller Design	57
7.1	Control of Bilinear Systems	57
7.2	Linear Quadratic Regulator (LQR)	58
7.3	Controller Model	59

8	Electron Beam Line Simulation and Controller Testing	63
8.1	Methodical Accelerator Design (MAD)	63
8.2	Base deep-RL Simulation Environment	64
8.3	Simulation Modifications for LQR Implementation	65
8.4	Controller Implementation	65
8.5	Controller Implementation Testing	67
8.6	Controller Comparison	69
8.7	Conclusion on Controller Testing	70
9	Global Conclusion	71
9.1	Future Work	72
	Bibliography	73

Abbreviations

Abbreviation	Meaning
LHC	Large Hadron Collider
SPS	Super Proton Synchrotron
PS	Proton Synchrotron
PSB	Proton Synchrotron Booster
AD	Anti-Proton Decelerator
CLIC	Compact Linear Collider
FCC	Future Circular Collider
LINAC	Linear Accelerator
LEIR	Low Energy Ion Ring
LIC	LHC Injection Chain
AWAKE	Advanced Proton Driven Plasma Wakefield Acceleration Experiment
RF	Radiofrequency
CGSN	CERN Neutrinos to Gran Sasso
CCC	CERN Control Center
LS2	Long Shutdown 2
BPMs	Beam Position Monitors
LQR	Linear Quadratic Regulator
iLQR	iterative Linear Quadratic Regulator
RL	Reinforcement Learning
ANN	Artificial Neural Network
SVD	Singular Value Decomposition
LWFA	Laser Driven Plasma Wakefields
PWFA	Proton Driven Plasma Wakefields
ORM	Orbit Response Matrix
MAD	Methodical Accelerator Design
DARE	Discrete Algebraic Riccati Equation

Mathematical Notation

Vector notation is in bold, smaller case	v
Upper case vector notation sometime utilized, and explicitly stated	R
Matrices are in capital	M
Time iterations are noted in parenthesis	$\mathbf{x}(t)$
Derivatives relative to time are noted with dot	$\dot{\mathbf{x}}(t)$
Derivatives relative to displacement are noted with prime	$\mathbf{x}'(s)$

Chapter 1

Introduction to the CERN complex

The European Organization for Nuclear Research (CERN) was established in 1954 with the goal of returning high-end physics nuclear research to Europe after the 2nd World War. It is based in Meyrin, Switzerland in the Greater Geneva Area. Over the previous decades, CERN has grown to be one of the largest nuclear research institutions in the world. Today, there are approximately 16 000 members of the personnel on site every day. It has 12 accelerators, among them the Large Hadron Collider (LHC) and the Super Proton Synchrotron (SPS), a decelerator, and a very wide variety of experiments which probe the fundamental structure of both matter and anti-matter. CERN also researches possible new particle acceleration techniques and paradigms, and it is currently in the exploratory phase of the construction of the Future Circular Collider (FCC) and the Compact Linear Collider (CLIC), both of which are planned to substantially increase the energy of particle beam acceleration and collision. [1]

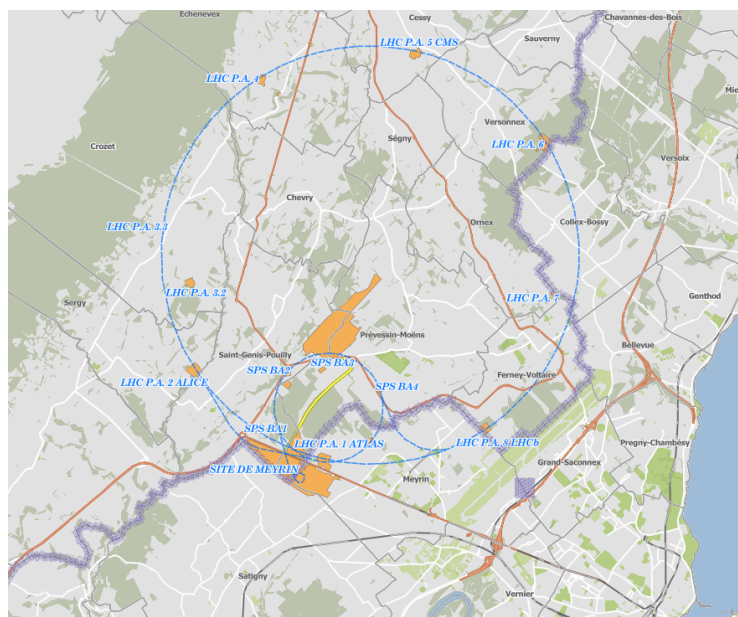


Figure 1.1: Map of CERN and its sites in France and Switzerland (in orange) and the major accelerators (in blue) [2].

1.1 The Accelerator Complex

The entire accelerator complex of CERN can be seen on *Figure 1.2*. This image shows all 12 accelerators, as well as all their corresponding experiments.

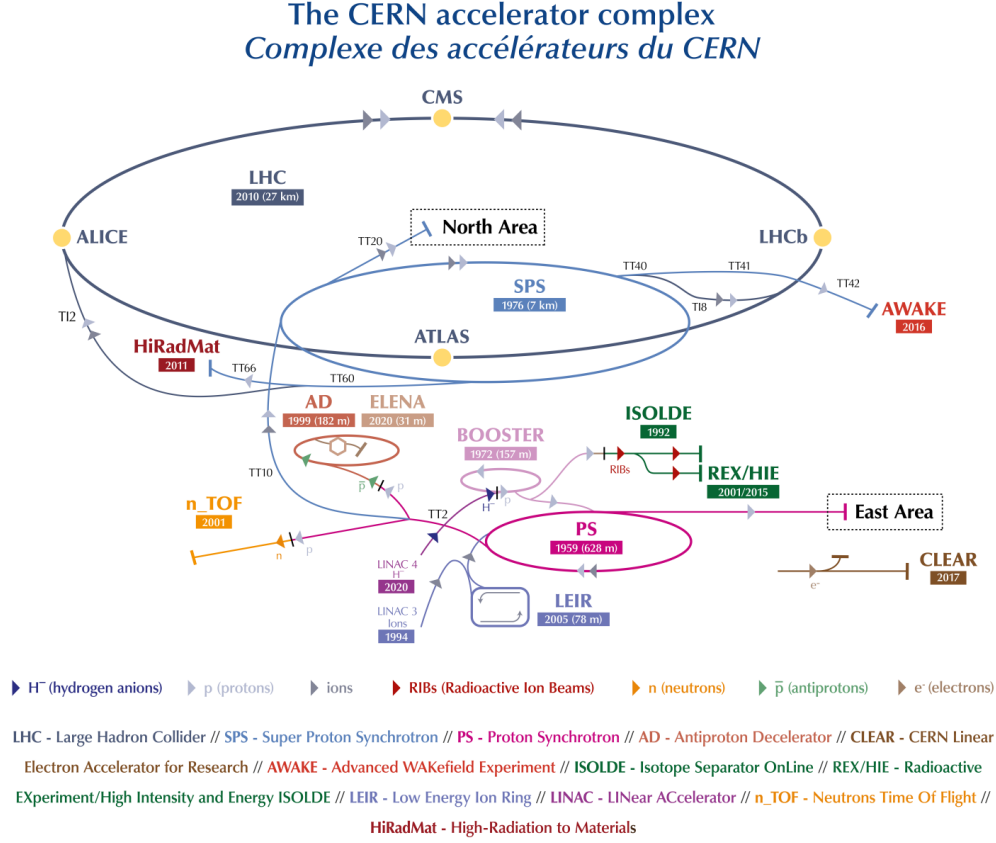


Figure 1.2: CERN Accelerator Complex [3]

As the main focus of CERN is the LHC and its experiments, the major accelerators are coupled into a chain to help the acceleration of particles to higher energies. This chain is called the LHC injection chain (LIC). It includes the brand new Linear Accelerator 4 (LINAC4), the LINAC3, the Low Energy Ion Ring (LEIR), Proton Synchrotron Booster (PSB), Proton Synchrotron (PS) the SPS and then finally the LHC. LINAC3 and LINAC4 linear accelerators are the first steps in the acceleration processes. There are two sources of particles, one for negative hydrogen ions from LINAC3, and one for protons from LINAC4. LIC is capable of accelerating both particle beams. [3]

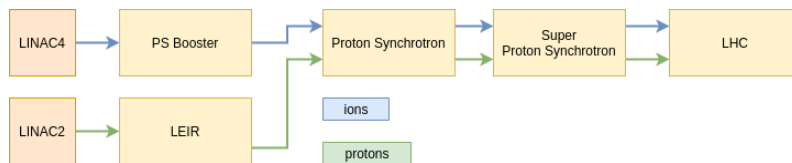


Figure 1.3: LHC Injection Chain (LIC) with LINACs in red and circular accelerators in yellow

1.1.1 Energy and Method of Acceleration

The energy of an acceleration is usually represented in terms of electron volts (eV). An electron volt is defined as the kinetic energy of a single electron particle being accelerated from standstill through vacuum with a potential difference of one volt. The concept of electron volts was originally developed for particle accelerators, and it is still widely used today. As an example, the LHC accelerates protons from 400 Giga eV (GeV) to 6.8 Tera eV (TeV), which results in collisions above 12 TeV, enough to detect the Higgs-boson. [4]

Currently at the CERN the acceleration of particles is done with Radiofrequency Cavities (RF cavities). They are present in most accelerators. Superconducting radiofrequency cavities are metallic chambers designed to accelerate or decelerate particle bunches in particle accelerators. The chambers contain an electromagnetic field, where the oscillations are known as radiofrequency, see *Figure 1.4*. In the LHC, these are designed to have a frequency of 400 MHz.

Due to this field, charged particles entering the chamber from one end get an electrical impulse and in turn get accelerated (or decelerated depending on the current energy of the particle). The timing of the injection for a particle into the chamber is extremely important, since if the particle enters with a non-target energy, either too early or too late, it will get decelerated or accelerated. Particles with the correct energy are not affected. With this process, the particles are eventually "broken" up into particle bunches as they traverse the various RF cavities around the accelerator.

As an example, in the LHC tunnels there are 16 RF cavities, each of them placed into cryomodules to enable superconducting. With these 16 RF cavities, the LHC can accelerate the particle bunches up to 1.6 TeV from 450 GeV injection energy. This is almost a 15x increase in energy. This process takes approximately 20 minutes.

RF cavities has been widely utilized around the world. This includes CERN too, where almost all the accelerators, including the LHC, SPS, PS and PSB, as well as the LINACs, use RF cavities.

However, there are several major drawbacks. One of them is the hard limit on the maximum attainable acceleration gradient. This limit essentially means that the acceleration process cannot achieve more than 100MV/m. Above this, the materials in the cavities break down

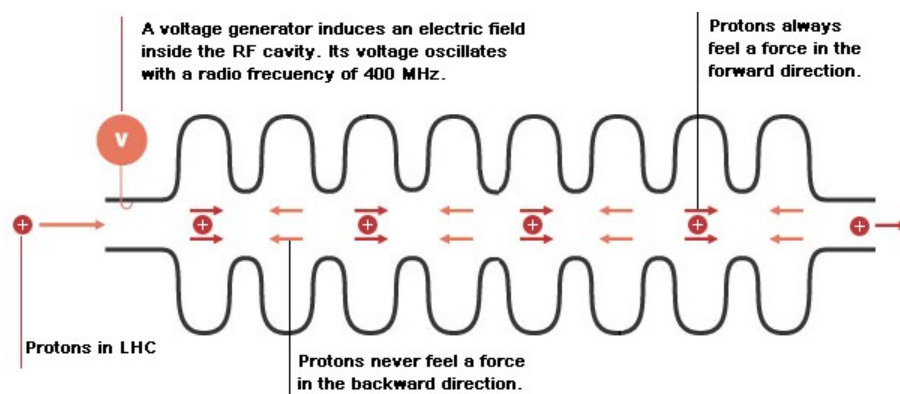


Figure 1.4: Side profile of RF cavities [5]

and stop being superconductive (quench).

Consequently, to increase the energy of the accelerator, a bigger machine is needed which can contain more RF cavities (such as the FCC or CLIC). It is of course possible to instead create circular accelerators where the particles orbit multiple times as they gain energy before entering the experiments. However, bending particles to keep to a target orbit creates a phenomenon called synchrotron radiation. This radiation occurs when relativistic particles are bent with magnetic fields, and it causes a significant energy loss, whilst also increasing the complexity of the understanding of the beam trajectory.

The acceleration energy of the RF cavities over a unit distance is called the acceleration gradient. It is defined as the energy added to the particles in a beam over a set distance. Where, the higher the gradient, the higher the acceleration is per fixed unit of distance. In the CERN context it is often written as 100MV/m, which is 100 MegaVolt/meter.

1.1.2 Large Hadron Collider (LHC)

The Large Hadron Collider is the flagship accelerator of the CERN complex. It is a 27 km long synchrotron. It has several major experiments like ATLAS, ALICE, CMS and LHCb. The LHC receives the particle bunches from the SPS at an energy of 400 GeV and then accelerates them up to 6.5 TeV. It was built between 1998 and 2008. First beam circulated in 2008. In 2012, the ATLAS experiment confirmed the existence of the Higgs-boson and with that fulfilling the original design purpose of the accelerator. Nowadays, the LHC is being prepared to increase the luminosity by a factor of 10 and to allow collision energies up to 14 TeV. These upgrades are part of the High Luminosity LHC upgrades (HL-LHC).

1.1.3 Super Proton Synchrotron (SPS)

The SPS is the 2nd largest accelerator at CERN. It was commissioned in 1979 and it has a 6.9 km ring. During its history it accelerated several particle types, such as: protons, electrons, antiprotons and positrons. Currently, it is part of the LHC and accelerates ions and protons. These particles arrive to the SPS from the Proton Synchrotron (PS) and

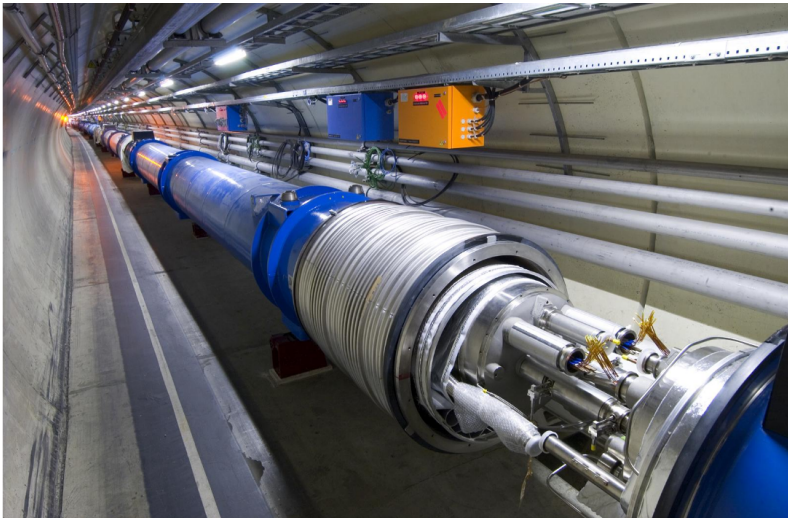


Figure 1.5: LHC vacuum tube in the tunnels [6]

are accelerated to 400 GeV from 26 GeV. Just like the LHC, it also has experiments, such as the AWAKE, North Area and HiRadMat. In the context of this project, the most important experiment is AWAKE. It is described in detail in the next chapter, *Chapter 2 The AWAKE Collaboration*.

1.1.4 LINACs

There are multiple linear accelerators at CERN; LINAC3 LINAC4 and AWAKE. The LINAC3 and LINAC4 are part of the LIC and AWAKE is an experimental accelerator. As mentioned above, LINAC3 and LINAC4 are designed to be the sources for the particles in the entire complex. LINACs have a strong history at CERN, as one of the very first accelerators ever built in 1959 was a linear accelerator. As the name suggests, there were already two other LINACs, LINAC2 and LINAC1. LINAC1 was decommissioned when LINAC3 came online for the first time, and LINAC2 was decommissioned the most recently, on 2020. LINAC4 took its place and today it is integrated into the LIC. During the writing of the thesis the first beams in LINAC4 took place. [7]

Another LINAC at CERN is the AWAKE experiment. It is built around a completely different accelerator paradigm, which instead of the above mentioned RF cavities, it uses plasma wakefields to accelerate particles. AWAKE is introduced in following chapter.

Chapter 2

The AWAKE Collaboration

This chapter is about the AWAKE collaboration at CERN. In the first sections, the experiment's history and its goals are presented. Afterwards, the method of plasma wakefield acceleration is explained. The experiment structure is then presented, with focus on the electron beam line. This contains the physical description of the system as well.

2.1 AWAKE History

The Advanced Proton Driven Plasma Wakefield Acceleration Experiment (AWAKE) is a research platform for new linear particle acceleration techniques. It was approved for construction by the CERN Council in 2013 and the first beam passed through the accelerator in 2016. [8]

The goal of this collaboration is to build an experiment that could accelerate particles using plasma wakefields as the main driver, instead of the conventional radiofrequency cavities (see *Subsection 1.1.1 Energy and Method of Acceleration*). Since the discovery

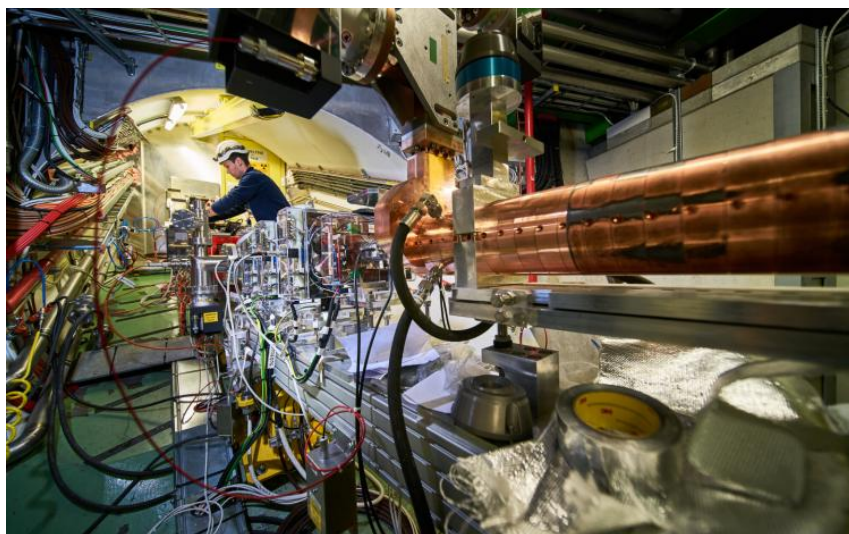


Figure 2.1: AWAKE Experiment beam line in the underground tunnels of the old CGSN experiment, close to the SPS transfer lines. [9]

of this technique in 1979 by scientists at Berkeley, many research institutions around the world are actively developing plasma wakefield based accelerators. As an example, consider the FACET-II at Stanford's SLAC laboratory [10], or Berkeley's BELLA [11].

The novelty of plasma wakefields comes from the the potential to allow acceleration of particles, other than protons, to very high energies to more easily study collisions. As mentioned in 1.1.1, with RF cavities, it is very difficult to accelerate electrons without losing a substantial amount of the input energy to synchrotron radiation. However, by using plasma wakefields in linear accelerators, one could theoretically achieve energies comparable to current circular accelerators in much shorter distances. This is important because often with heavier particles like protons the collisions create very complicated outcomes which are extremely difficult to decipher. Lighter particles like electrons would be ideal for collisions as their outcome is much easier to model. However, due to synchrotron radiation, electrons at this point cannot reach such high energies. Therefore, new accelerator techniques like proton-driven plasma wakefields are crucial for particle acceleration of the future. [12] [13]

In addition to the uniqueness of a plasma wakefield driver accelerator design, the AWAKE collaboration has decided to use *proton-driven* plasma wakefields for the acceleration mechanism. This is a world first at the time of construction. Therefore, besides the plasma wakefields, this CERN experiment also investigates the viability of using proton bunches as the plasma wakefield drivers. [13]

The specific goals of the experiment can be summarized in four points:

1. Development of a long, scalable and uniform proton driven plasma cell for future applications in accelerators.
2. Development and understanding of plasma wakefields with externally injected electrons.
3. Research of injection dynamics and the possible production of high GeV electron bunches in the accelerator.
4. Research the physics behind self-modulation of long proton bunches in plasma.

Since its first beam in 2016, AWAKE has successfully managed to accelerate externally injected electrons from 19 MeV up to 2 GeV in only 10 m. After the success of 2018, the entire CERN complex went into the Long Shutdown 2 (LS2) period for major upgrades to all of its systems. This included AWAKE, where it is being upgraded to increase the acceleration energy of the electrons from the initial 19 MeV to 10 GeV. The LS2 period is expected to end in the second half of 2021 and first beams are already being injected into the complex at the time of writing of this report. [14] [9] [13]

2.2 Principles of Plasma Wakefield Acceleration

Plasma wakefield acceleration is a novel particle acceleration technique from 1979. It was discovered at UCLA by Toshiki Tajima and John M. Dawson [15]. Subsequently, a prototype accelerator was built by Chandrashekhar J. Joshi, also at UCLA. [16] The key principle in plasma wakefield acceleration is that the particles are accelerated in a wave in a wakefield created by an outside energy source. This could either be a strong laser or a

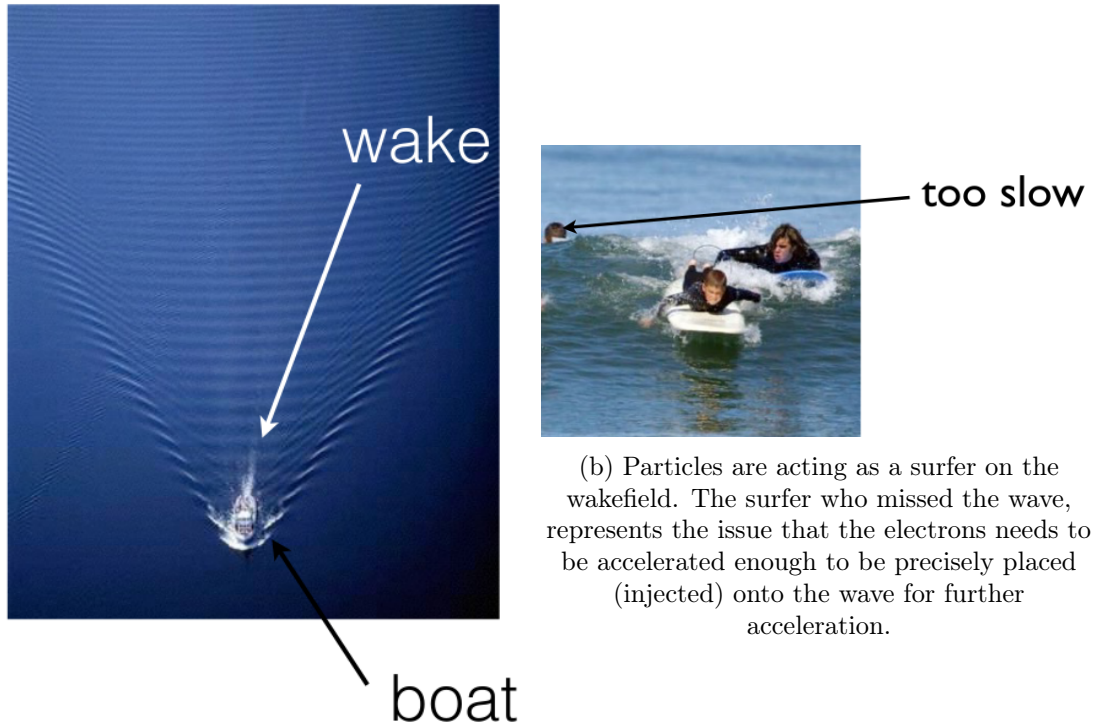
high energy particle bunch. By definition, a wake is a wave-pattern that is produced by an object moving through a fluid. This phenomenon can be well illustrated with boats travelling through water, see *Figure 2.2a*.

The wave created by the boat is called the driver wave. The particles which are to be accelerated are effectively analogous to surfers on the wave, where they travel faster than the wave as they gain the energy, see *Figure 2.2b*. Subsequently, the higher the energy of the driver, the higher the energy of the acceleration. The particles which are to be accelerated are called witness particles. The underlying formula of the acceleration gradient for a plasma wave is:

$$E = c \cdot \sqrt{\frac{m_e \cdot n_e}{\epsilon_0}} \quad (2.1)$$

where, E is the electric field, c the speed of light, m_e the mass of an electron, ϵ_0 the permittivity of free space and n_e is the plasma electron density. The energy of the gradient increases if the energy of the plasma electron density increases, where the permittivity ϵ_0 is a constant.

There are two mainstream implementations of wakefield generation. Laser driven and the above mentioned proton driven wakefields. The main difference lies in where the energy for the driver wave comes from. In the laser driven wakefields, the energy into the system is input with very high energy lasers (up to petawatt strength), whereas in proton driven wakefields the driver wave is created by high energy proton waves. In the AWAKE case these protons arrive from the SPS. [18] [13]



(a) Wakefield produced by a boat on the surface of a body of water.

(b) Particles are acting as a surfer on the wakefield. The surfer who missed the wave, represents the issue that the electrons need to be accelerated enough to be precisely placed (injected) onto the wave for further acceleration.

Figure 2.2: Analogous representation of the wakefield and witness particles. [17]

2.3 Layout and Structure of the Experiment

The AWAKE experiment is built in the old CERN Neutrinos to Gran Sasso (CNGS) facility approximately 120 m underground, see *Figure 2.3*. This site was chosen since there is an already established transfer connection to the SPS, and the since the CNGS facility has reached its end of life, this space was available for new experiments with high energy proton beams. Additionally, sufficient radiation protection was also already integrated to the tunnels in this area. [18]

The main elements of the experiment are:

- Proton transfer line from the SPS.
- RF gun electron source.
- Electron transfer line from the source to the;
- Electron-proton merging point.
- Accelerating plasma vapor cell.

The accelerating plasma vapor cell is approximately 10 m long. The electrons arrive through the 15 m long electron beam line, which transfers the particles from an RF gun to the vapor source to be injected for acceleration. The driver proton bunches arrive from the SPS through the 800 m transfer line. Since no sufficient access was available to the LHC for the proton beams, the SPS was chosen instead. The particles arrive at 400 GeV, which makes the AWAKE experiment the highest energy particle driven plasma accelerator. This enables the acceleration from the 18.8 MeV of the electron gun to the 2 GeV mentioned above. The layout of the experiment is shown in *Figure 2.4*. [18]

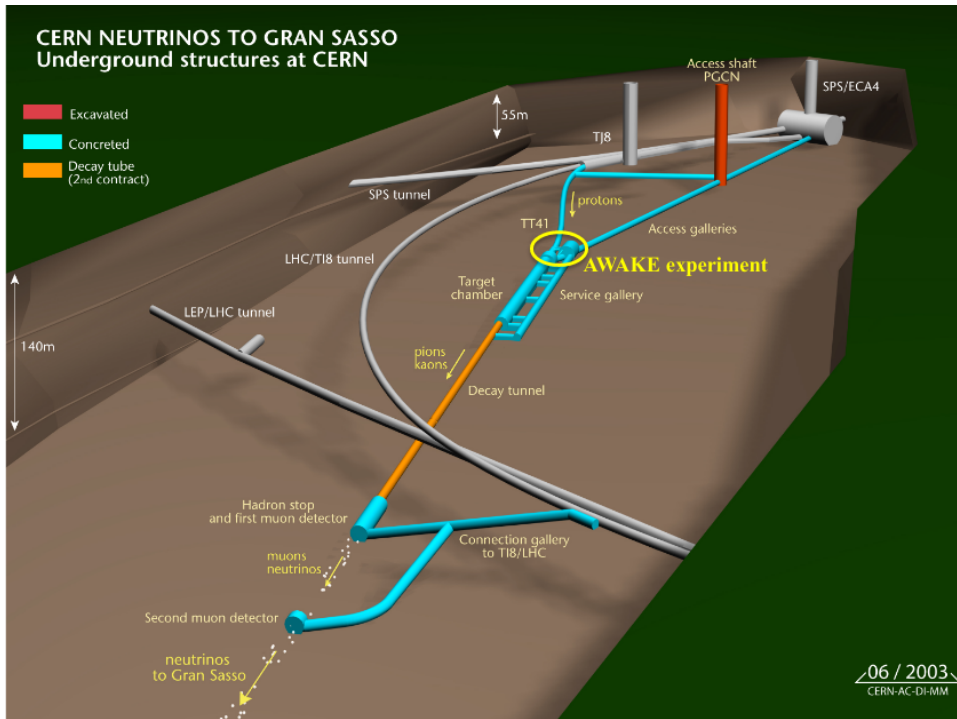


Figure 2.3: AWAKE Experiment placement in the old CNGS facility close to the SPS beamline. There is sufficiently easy access to the 400 GeV proton beams from the SPS.

[19]

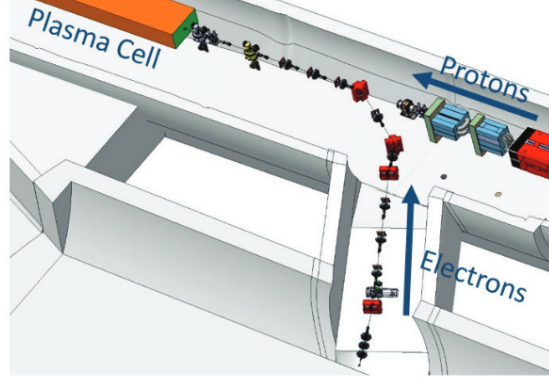


Figure 2.4: AWAKE Experiment Overview. This image shows that the electron beam line is connected to the experiment via the electron merging point, where the witness electrons are injected into the plasma driver wave. [19]

2.3.1 Electron RF gun and beam line

The witness electron bunches are created independently by a 5 MeV RF gun before injection into the plasma cell. The model of this beam line is shown in *Figure 2.5*. The current design of the plasma chamber expects the electrons to be injected with 18.8 MeV and then to be accelerated to 2 GeV over the 10 m long plasma cell. [18]

Parameter	Dipole	Quad	Corr
Number of units	4	11	11
Min. field [Tm]	0.0096		0
Max. field [Tm]	0.0456		$4.34 \cdot 10^{-4}$
Min. gradient [T]		0.01	
Max. gradient [T]		0.18	
Magn. length [mm]	177.5	70.8	110–
Total length [mm]	290	114	34
Free aperture [mm]	≥ 70	≥ 70	≥ 70
radius (x/y)[mm]	$\pm 30/20$	20	13

(2.2)

The RF gun's first 1.5 m is designed to be a short booster LINAC which accelerates the particles to the target energy (10-19 MeV). The rest of the beam line is composed of five dipoles and nine quadrupoles (the description and dynamics of dipoles and quadrupoles can be found in *Section 3.1 Transverse Dynamics*) and the technical specifics of the devices can be seen in Table 2.2. Additionally, in the beam line there are three bends, a 1 m vertical step, a 60° bend to position the beam parallel to the plasma chamber and a final bend to inject the electrons orthogonal. In total there are 11 dipoles available for trajectory correcting control. There are also 11 Beam Position Monitors (BPMs), which measure the position of the beam.

There are various approaches to controlling AWAKE's electron beam line trajectory. They are described in *Chapter 4 Current Trajectory Control Methods for the Electron Beam Line*. Due to the low energy, and low chance of damage to the surrounding system, the electron

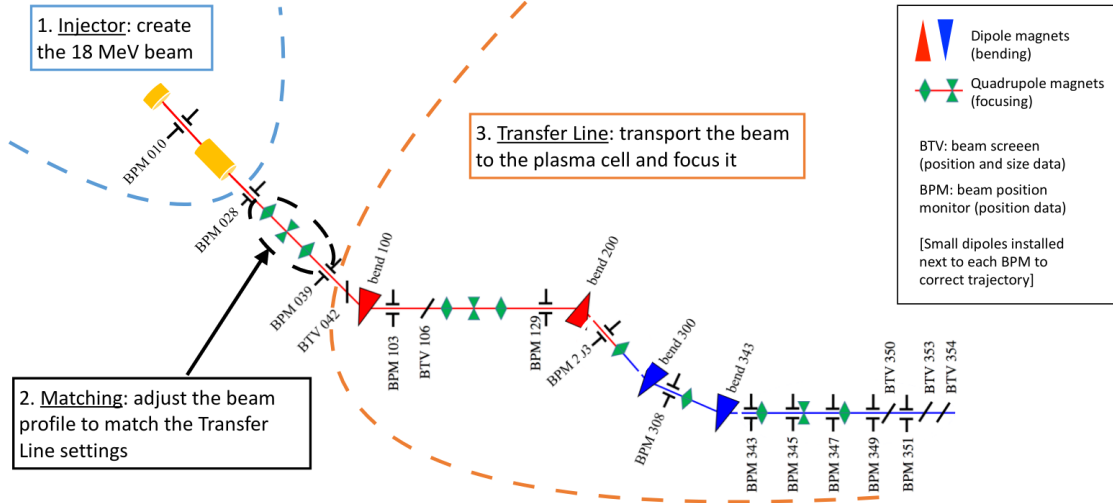


Figure 2.5: AWAKE Electron Beam Line top down overview. [20]

beam line had become a sort of test platform for orbit correction controller algorithms. This thesis builds on top of the work done previously. State of the art controllers for the beam line are described later in the report, see *Chapter 4 Current Trajectory Control Methods for the Electron Beam Line*.

Chapter 3

Modeling of Particle Motion

This chapter introduces the modelling of charged particle motion. This is first done with a derivation of the forces affecting the particle. Then, using multipole expansion of a magnetic field, dipole and quadrupole magnets are introduced. Based on the magnet and charged particle interactions, the transverse equations of motions are then derived. In the end the system is then put into a linear form.

3.1 Transverse Dynamics

To model a beam or particle bunch travelling through an accelerator beam transfer pipe, first the movement of a single particle has to be considered. This is done with transverse beam dynamics or beam optics. Transverse dynamics describe the movement and dynamics of a single charged particle travelling through a static magnetic field. The term beam optics also applies, because similar effects are applied to particle movement as to optical bending and optical focusing. Furthermore, very similar phenomena occurs when magnetic fields are used to affect charged particles, such as dispersion, luminosity or chromaticity. [21, p.17]

3.1.1 Transverse Reference Frame

Particle modeling is usually done in the six dimensional phase space with transverse local coordinates (x, y) , their derivatives, also called angular displacement, (x', y') , the energy of the particle (E) and the longitudinal displacement (s) [21, p.6] [22, p.213]:

$$\mathbf{x} = (x, x', y, y', E, s)^T \quad (3.1)$$

Phase space describes all possible states of a system, where each parameter is represented as a separate axis. This setup describes a single unique point for each and every state of the system. The movement of the system along the phase space is called phase space trajectory, see *Figure 3.1*.

Derivative representation in particle accelerator context

Note that generally in the accelerator particle modeling context, the convention for the derivatives representation of (x, y) are in terms of the longitudinal displacement along the

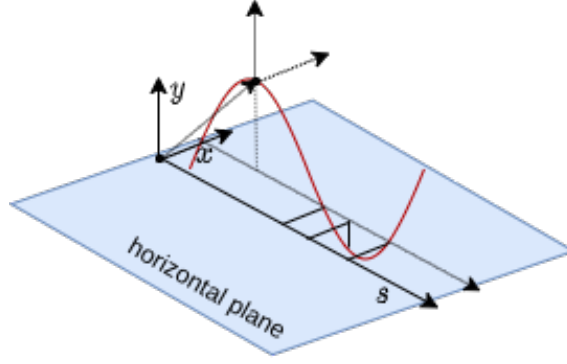


Figure 3.1: Phase space representation of the particle movement along s . The phase is represented as ϕ and the displacement along the x and y axis is shown as well.

accelerator, s , not in terms of time. When the time derivative is used it is represented with a dot instead of a mark [23, p.16]:

$$x' := \frac{dx}{ds}, \quad \dot{x} = \frac{dx}{dt} \quad (3.2)$$

In the rest of the chapter the above convention is used unless otherwise explicitly stated.

Orthogonal right handed frame

Within the phase space of the particle movement, the transverse equations of motion are derived in an orthogonal right handed frame (x, y, s) . This frame is "attached" to the particle along the orbit as it moves. In accelerating and co-moving frames this approach would require complex Lorentz transformations. Therefore, the frame is fixed to the laboratory instead. It follows the reference orbit to model the local displacement of the particle in the longitudinal position, s , see *Figure 3.2* and *Figure 3.1*. [23, p.16]

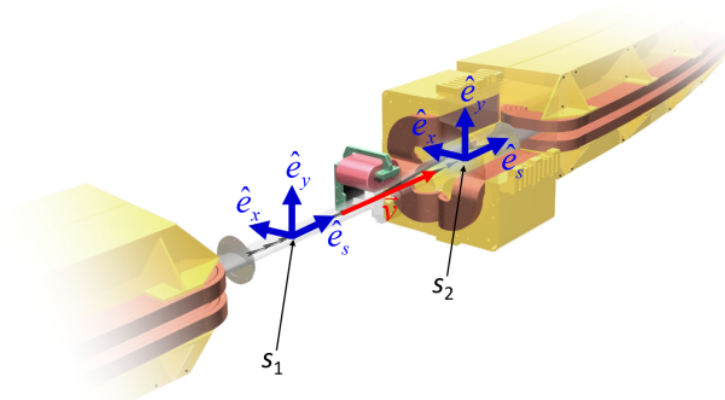


Figure 3.2: Particle orbit local reference frame in an accelerator beam tube. The unit vectors are evaluated at longitudinal coordinate s_1 and s_2 . [23, p.16]

Additionally, a set of $(\hat{e}_{x,y,s})$ unit vectors are also defined, where the \hat{e}_s is tangential to the reference orbit and points in the direction of the particle's velocity v . Furthermore, \hat{e}_y is a perpendicular unit vector to the transverse plane, where it points upwards in the vertical plane. Lastly, \hat{e}_x is defined as the cross product of $\hat{e}_s \times \hat{e}_y$, being perpendicular to both. [23, p.16]

3.1.2 Lorentz Force and Magnetic rigidity

In a particle accelerator, bending and focusing forces acting on charged particles are used to control the trajectory of the beam. The forces are applied with electromagnetic fields. This force is called the Lorentz force [23, p.2]:

$$\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad (3.3)$$

where q is the charge of the particle moving with velocity \mathbf{v} , \mathbf{E} is the electric field and \mathbf{B} is the magnetic field. The forces acting longitudinally ($\mathbf{E}_{||}$) increase the energy of the travelling particle, whereas the perpendicular (transverse) forces (\mathbf{B}_{\perp}) only bend the particles. In this project the problem of particle acceleration is not considered. Thus assuming only transverse bending (\mathbf{B}_{\perp}), the Lorentz force equals to the centripetal force (this is in the case of circular accelerators where the circular orbit bending radius is ρ):

$$m \frac{\mathbf{v}^2}{\rho} = q(\mathbf{E}_{\perp} + v\mathbf{B}_{\perp}) \quad (3.4)$$

where, m is the ultra-relativistic mass of the particle (meaning the velocity \mathbf{v} can be approximated to be equal to the speed of light, c [23, p.2]). It is defined as $m = \gamma_r m_0$ with the rest mass m_0 and Lorentz factor γ_r . The Lorentz factor defines how the properties of objects such as length or measurement of time change as they are moving. It is most often used in special relativity and in Lorentz transformations. In this case, it describes the effect on the mass of the particle as it reaches relativistic speeds [24]:

$$\gamma_r = \frac{1}{\sqrt{1 - \frac{\mathbf{v}^2}{c^2}}}. \quad (3.5)$$

From 3.5 it is possible to see that the γ_r rapidly reaches infinity as the particle reaches the speed of light. In practice this means that as the speed of the particle increases, the resistance against acceleration increases as well, and it represents itself as an increase in mass. However, since the rest mass m_0 for a particle like the proton is $1.6726219 \times 10^{-27}$ kg, the computation for the mass keeps to relatively manageable values, even as \mathbf{v} approaches c .

With m defined, the momentum of the particle is introduced as:

$$p = m\mathbf{v} \quad (3.6)$$

which can be substituted into Equation (3.4) by first factoring out \mathbf{v} . To simplify notation, from now on the \perp sign is neglected, therefore: $\mathbf{B} = \mathbf{B}_\perp$ and $\mathbf{E} = \mathbf{E}_\perp$:

$$mv \frac{v}{\rho} = q(\mathbf{E} + \mathbf{vB}) \quad (3.7)$$

and then substituting in the momentum for mv :

$$p \frac{v}{\rho} = q(\mathbf{E} + \mathbf{vB}) \quad (3.8)$$

then, neglecting E since only transverse magnetic fields are considered gives:

$$p \frac{v}{\rho} = q\mathbf{vB}. \quad (3.9)$$

Afterwards, multiplying by ρ on both sides yields:

$$p\mathbf{v} = q\mathbf{vB}\rho \quad (3.10)$$

and finally, dividing both sides by \mathbf{v} gives:

$$p = q\mathbf{B}\rho. \quad (3.11)$$

Where $\mathbf{B}\rho$ is the magnetic rigidity, or magnetic stiffness. If a charged particle is travelling through a bending magnet on a circular orbit with a bending radius of ρ , then the magnetic rigidity defines the rigidity against bending, that is: the higher the particle momentum the higher the resistance against the deflection generated by the magnetic field. [23, p.2]

3.1.3 Multipole expansion of a magnetic field

Multipole expansion is a mathematical method that is used to decompose functions dependent on angles, as a series. This applies to the particle movement context as well, since the displacement generated by the above mentioned \mathbf{B} field are angular. Therefore, it is possible to apply multipole expansion to magnetic fields. This can give an understanding to how magnetic fields generated by magnets in the accelerator affect the particle and beam trajectory. [23, p.3]

It is important to note here that there are two main assumptions regarding the \mathbf{B} field:

1. only transverse displacement is considered in the (x, y) planes (horizontal and vertical) from the reference orbit and;
2. \mathbf{B} does not have a dependence on the longitudinal coordinate s . This is in terms of a single magnetic field generated by a magnet, not when multiple magnets are present in a beamline.

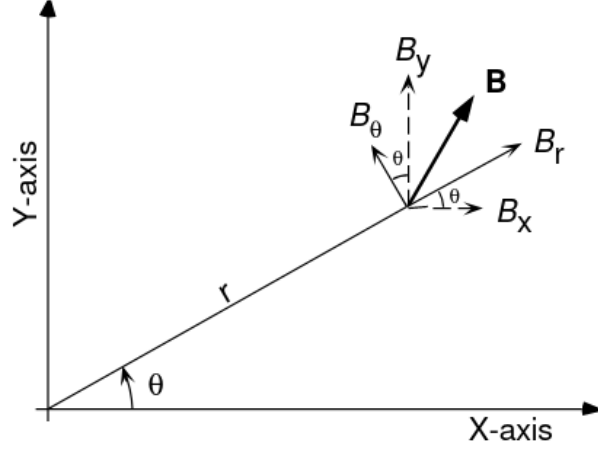


Figure 3.3: The Cartesian coordinates of the transverse components of B , represented as B_x and B_y . [25, p.2]

Therefore, the transverse vector components of \mathbf{B} are denoted as $B_x(x, y)$ and $B_y(x, y)$, see *Figure 3.3*.

Furthermore, since the magnets which are placed in an accelerator are further away than the potential transverse displacement, x, y then Maxwell's equations reduce to:

$$\begin{aligned}\nabla \cdot \mathbf{B} &= 0, \\ \nabla \times \mathbf{B} &= 0.\end{aligned}\tag{3.12}$$

which means that the divergence and the curl will equal to zero. This allows \mathbf{B} to be defined in terms of a scalar potential Φ . Also, the magnetic fields can be generally derived from a vector potential \mathbf{A} and since both \mathbf{A} and Φ only depends on transverse coordinates (as established above) one can write [23, p.3], [21, p.65]:

$$\mathbf{B}(x, y) = -\nabla\Phi(x, y) = \nabla \times \mathbf{A}(x, y)\tag{3.13}$$

Then Equation (3.13) can be evaluated at each of the transverse components, B_x and B_y , which will give:

$$\begin{aligned}B_x(x, y) &= -\frac{\partial\Phi}{\partial x} = +\frac{\partial A_z}{\partial y} \\ B_y(x, y) &= -\frac{\partial\Phi}{\partial y} = -\frac{\partial A_z}{\partial x}\end{aligned}\tag{3.14}$$

where, A_z is the z component of the vector field \mathbf{A} . Note that here, the above derivations are a huge simplification of the mathematics because they lie out of the scope of this project. Further reading can be found at [23, p.3] and [21, p.65], [25] or at [22, p.125].

Based on this, one can then find the Cartesian multipole strengths for the first two orders of the magnetic decomposition:

Order	Name	Magnetic field	Multipole strength
$n = 1$	upright dipole	$\mathbf{B}_1 = \frac{p}{q} \kappa \hat{e}_y \quad (3.15)$	$\kappa = \frac{qB_0}{p} b_1 \quad (3.16)$
$n = 1$	skew dipole	$\mathbf{B}_1 = -\frac{p}{q} \underline{\kappa} \hat{e}_x \quad (3.17)$	$\underline{\kappa} = \frac{qB_0}{p} a_1 \quad (3.18)$
$n = 2$	upright quadrupole	$\mathbf{B}_2 = -\frac{p}{q} k (y \hat{e}_x + x \hat{e}_y) \quad (3.19)$	$k = -\frac{qB_0}{pr_0} b_2 \quad (3.20)$
$n = 2$	skew quadrupole	$\mathbf{B}_2 = \frac{p}{q} \underline{k} (x \hat{e}_x - y \hat{e}_y) \quad (3.21)$	$\underline{k} = -\frac{qB_0}{pr_0} a_2 \quad (3.22)$

Table 3.1: The first two magnetic multipoles for a given \mathbf{B} field. n represents the order of the decomposition. Skew means that the magnet is rotated by 45° relative to the upright position. B_0 is related to Ampere's law and it is derived in the subsequent section *Section 3.2 Magnets* [23, p.6]. Furthermore, b_1, a_1, b_2 and a_2 are dimensionless expansion coefficients used in guiding the derivations, and are not considered later in the project. More can be found in [23, p.4]. Similarly to the expansion coefficients r_0 represents a free variable called reference radius [23, p.4].

As shown in *Table (3.1)*, the first two ($\mathbf{B}_1, \mathbf{B}_2$) multipole expansions of the \mathbf{B} field represent the effects of an upright or a skew dipole magnet, whereas the second represents the effects of an upright and a skew quadrupole. As mentioned above there are additional effects which represents higher orders like quadratic and beyond. However, since the AWAKE electron beam line does not have magnets corresponding to those effects, they are not shown here (see *Subsection 2.3.1 Electron RF gun and beam line*). Additional material on sextupoles and octupoles can be found in [21, p.76] and in [23, p.14].

To state again, the above representation is a considerable simplification of the physics and mathematics, but due to the scope of this project it was not included.

3.2 Magnets

Based on the multipole expansion of the magnetic field, (see *Table (3.1)*) one can see that there are also multiple higher order components to the magnetic field. At $n = 3$ the magnetic field corresponds to sextupoles and at $n = 4$ it corresponds to octupoles, and at $n = 5$ it is called decapoles. Other than the dipole and quadrupole magnets, these higher order fields correct highly non-linear phenomena, such as chromaticity or dispersion, in the beam. They are often utilized in larger, circular accelerators where the beam passes through the magnets many times, which induces these non-linear effects. However, since in the AWAKE electron beam line there are only dipoles and quadrupoles (*Subsection 2.3.1 Electron RF gun and beam line*), the higher order corrector magnets (sextupoles, octupoles and decapoles) are not discussed in this project. So, in the following two sections the models for an upright dipole and quadrupole are considered. [23, p.6] *Figure 3.4* shows the generated field of the different magnets, including higher orders and skew (rotated by 45°) configurations.

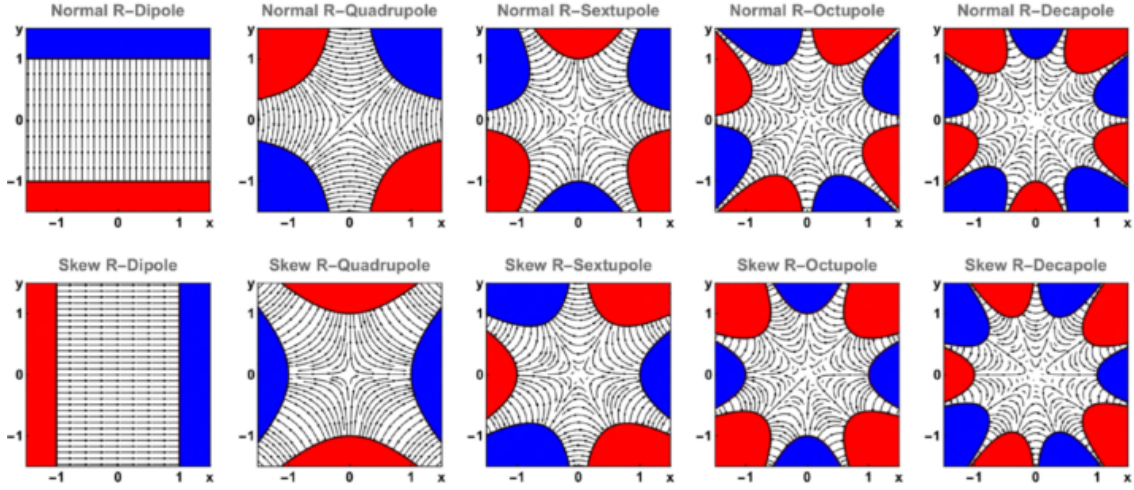


Figure 3.4: Generated fields from dipoles, quadrupoles, sextupoles, octupoles and decapoles. Skew means that the orientation of the magnet is rotated relative to the vertical plane of the moving frame.[23, p. 7]

3.2.1 Dipole magnets

Dipole magnets are two poled bending magnets that are primarily used to deflect the particle beam towards the design orbit. There are two major implementations of dipoles: sector dipoles used in large circular accelerators and corrector dipoles which are meant to correct small transverse displacement in the trajectory of the beam. Both are often used in circular accelerators. However generally, in LINACs only correctors are applied. Effectively for modeling purposes, the only real difference between the sector and the corrector dipoles are: the winding numbers and the maximum current flowing through the coils.

In flat accelerators, the dipole magnetic field is pointing perpendicular to the design orbit (see assumptions in *Subsection 3.1.3 Multipole expansion of a magnetic field*). This is the case for the first multipole strength of a B field, presented in *Table (3.1)* [23, p.9]:

$$\mathbf{B} = \frac{p}{q} \kappa \hat{e}_y \quad (3.23)$$

where, κ is:

$$\kappa = \frac{qB_0}{p} b_1. \quad (3.24)$$

Variable b_1 as mentioned in *Table (3.1)* is a dimensionless expansion coefficient that was used to guide the derivations. It is essentially a free-variable and from now on can be neglected. p stays to be the particle momentum with q charge.

Performing the κ substitution into 3.23, q and p cancel out, yielding:

$$\mathbf{B}(x, y) = B_0 \cdot \hat{e}_y \quad (3.25)$$

To generate a uniform and flat magnetic field, there needs to be two equipotential, Φ , surfaces opposite to each other (see left image of *Figure 3.8*). Then using Ampere's law, it is possible to approximate the generated dipole field [23, p.10]:

$$B_0 = \mu_0 \frac{nI}{h} \quad (3.26)$$

where, μ_0 is the free-space permeability, I the current running through, n the number of turns in the coils and h the distance between the coils. This is also shown on *Figure 3.8*. Note here, that even though I is a single variable for the current, which has no dependence on s , for multiple dipoles in a system, I will produce a dependence on s for the locations of the magnets.

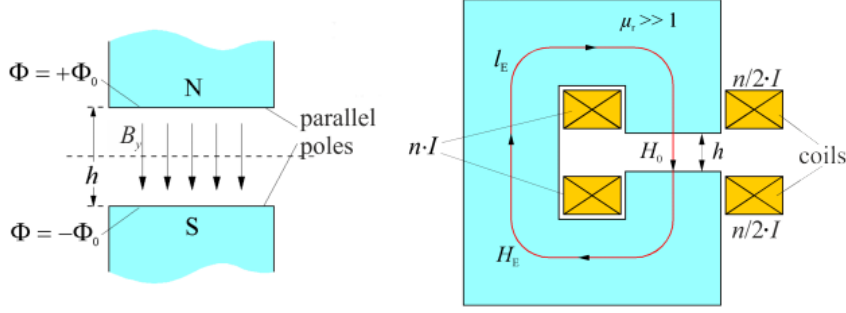


Figure 3.5: Pole-profile (left) and a C-shaped cross section (right) of a bending dipole magnet. The distance between the poles of the magnet is denoted as h , with n number of coils, I current flowing through it. B_y denotes the magnetic field vector component in the y transverse direction and H_0 the magnetic field strength. Φ denotes the material surface potentials, μ_r the material permeability and μ_0 the free-space permeability between the magnet surfaces. [23, p.9]

Dipole bending curvature

The bending curvature of a magnet defines whether the particles is deflected "inwards" or "outwards" relative to the accelerator. It is related the dipole strength, κ (see *Table (3.1)*) by an inverse relation [23, p.9]:

$$\kappa = \frac{q}{p} B_0 = \frac{1}{\rho} \quad (3.27)$$

where, ρ is the bending radius. Substituting 3.26 for B_0 gives:

$$\kappa = \frac{q}{p} B_0 = \frac{q\mu_0}{p} \frac{nI}{h} \quad (3.28)$$

henceforth, the sign of the magnetic field will dictate whether the bending is "inwards" or "outwards". *Figure 3.6* shows that bending dictates the angle of the particle φ and the magnitude of the displacement after the bending, y .

Furthermore, as mentioned above, there are two types of dipoles: sector dipoles and corrector dipoles. In implementation, the sector dipoles have the bending curvature ρ of a circular accelerator, and their purpose is to bend the beam into a circle over the entire period of acceleration. Therefore, the sector dipoles are generally configured up with a constant I , whereas in corrector dipoles the current is variable. The variable current

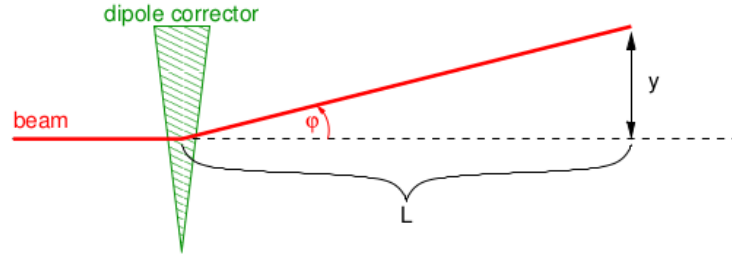


Figure 3.6: Dipole curvature and bending with deflection angle φ and with displacement y . In circular accelerators ρ and φ are equal since the dipole is meant to bend the beam in a circle. These type of magnets are called sector magnets.

is required to correct for transverse displacement. Additionally, corrector dipoles are thinner, with fewer number of coils. An example of a thin corrector dipole is shown in Figure 3.7. [22, p.190]

3.2.2 Quadrupole magnets

Quadrupoles are similar to dipoles, but instead of two magnets they are built using four magnets. In particle accelerators they are used to focus the trajectory of the particle, or particle beam, and they do not bend the beam. This focusing is analogous to the optical focusing. Therefore optics is generally used to model the effect of a quadrupole on a beam [23, p.11].

To achieve beam focusing, one must focus in both the vertical and horizontal plane. This means that a magnetic field is required in both the x and y planes with corresponding fields, B_y and B_x . As established above, to have a transverse magnetic effect, the field must be perpendicular to the current axes to have the desired effect. Thus, the B_y corresponds to the x axis and B_x to the y axis. This is provided by the second order magnetic multipole decomposition (see Table (3.1)):

$$B_y = g \cdot x, \quad B_x = -g \cdot y \quad (3.29)$$

where g is the magnetic field gradient along the axes:

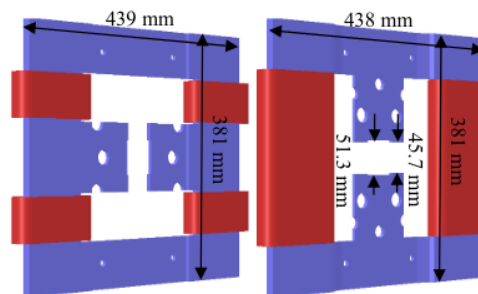


Figure 3.7: An example thin horizontal and vertical steering dipole. Thin correctors are meant to be employed to fix small perturbations in design orbit by quickly changing the current running through. [26, p. 7]

$$g = \frac{\delta B_y}{\delta x} = -\frac{\delta B_y}{\delta y} \quad (3.30)$$

Similar to dipoles, using Ampere's law, it is possible to approximate the required ampere-turns to generate the magnetic gradient g :

$$nI = \frac{1}{\mu_0} \int_0^a g r dr \rightarrow g = \frac{2\mu_0 nI}{a^2} \quad (3.31)$$

where, I is the current, μ_0 is the free-space permeability, a is the aperture and n is the number of turns in the coils. The aperture is defined as the radius of a circle around the beam center in the magnet towards the iron yoke (housing), see *Figure 3.8*.

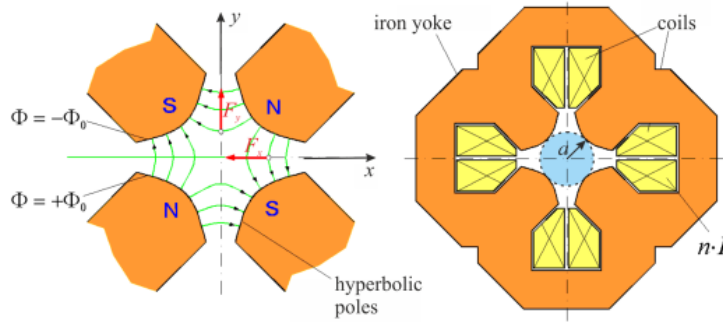


Figure 3.8: Side profiles of a focusing quadrupole magnet. Φ represents the surface potential and a is the aperture. $F(x, y)$ are the restoring forces acting on the particle. Notice how F_y points outwards, meaning it is defocusing, while F_x is pointing inwards, thus focusing. [23, p.11]

Subsequently, using the Lorentz force and Equation (3.29) it is possible to compute the restoring forces acting on the particle:

$$\mathbf{F} = q \cdot (\mathbf{v} \times \mathbf{B}) = q\mathbf{v}g \cdot (x\hat{e}_x - y\hat{e}_y) \quad (3.32)$$

By the varying the sign of the gradient g , it is possible to see that the quadrupole simultaneously focuses in one of the axes while it defocuses in the other. Practically, this means that if the particle beam is focused in the horizontal plane, it is defocused in the vertical plane. In accelerators, it is often corrected by using another quadrupole downstream which is rotated (skewed) to focus in the vertical plane and defocus in the horizontal.

It is therefore now possible to evaluate the upright quadrupole strength k from *Table (3.1)* ($n = 2$), which yields:

$$k = \frac{q}{p}g = \frac{q\mu_0}{p} \frac{nI}{a^2} \quad (3.33)$$

Just as in Equation (3.32) for the gradient g , varying the sign of the k indicates whether the particle beam is focused or defocused. For the case where $k < 0$, the magnet is horizontally focusing and vertically defocusing and for the case $k > 0$, it is the opposite. [23, p.11]

Thin lens approximation

The angle of the quadrupole beam deflection can be computed using a thin lens approximation, where the thickness of the magnet L is assumed to be much smaller than the focal length f , see *Figure 3.9*.

From the figure, one can deduct the following approximation for the deflection angle α :

$$\tan \alpha = \frac{x}{f} \quad (3.34)$$

Due to the thin lens approximation it can be said that:

$$\tan \alpha = \frac{x}{f} = \frac{L}{R} \quad (3.35)$$

where, L is the thickness of the magnet and R is the radius of the bending. In thin lens approximation, L approaches the bending curve, since the bend is assumed to be so small, thus 3.35 holds. Furthermore, substituting the bending radius with $(q/p)B_y$, yields:

$$L \frac{q}{p} B_y = -\frac{q}{p} g x L = -x k L \quad (3.36)$$

where, k is the same quadrupole strength as in 3.33 with g being the magnetic gradient, p the particle momentum and q the particle charge. Equation (3.34) can be rearranged and solved for the vertical (f_y) and horizontal (f_x) focusing lengths:

$$\frac{1}{f_x} = -kL, \quad \frac{1}{f_y} = kL \quad (3.37)$$

which represents the quadrupole acting as an ideal lens in the thin lens approximation.

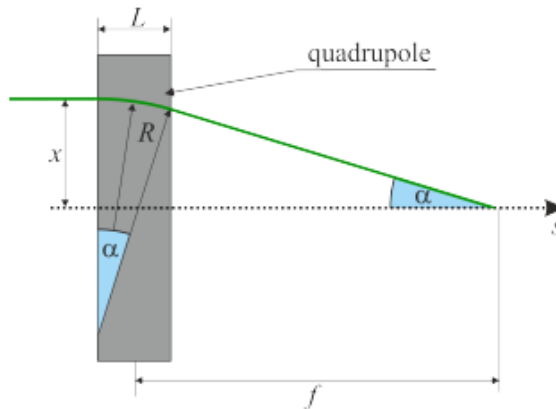


Figure 3.9: Thin lens approximation of a quadrupole. L is the thickness or length of the quadrupole, R the radius of the deflection, x is the displacement before focusing and f is the focal length.

3.3 Transverse equations of motion

To begin with the derivations of the equations of motion, the notion of the reference frame and derivatives definition needs to be expanded first (see *Subsection 3.1.1 Transverse Reference Frame*). Using the definitions of the derivatives in Equation (3.2), it is now possible to define the first derivative of the transverse displacement of the particle from the reference orbit at a longitudinal position, s in both the horizontal and vertical plane [23, p.17]:

$$x' = \frac{dx}{ds}, \quad y' = \frac{dy}{ds} \quad (3.38)$$

The reference orbit, or design orbit, represent an ideal path of the particle along the accelerator. Due to disturbances and misalignments and inter particle interactions in a beam, this reference orbit is never followed precisely, see *Figure 3.10*.

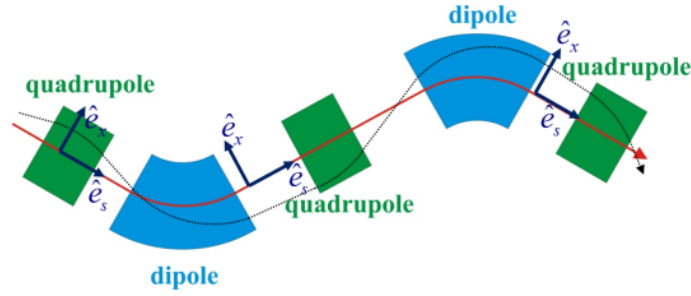


Figure 3.10: Transverse displacement along the reference orbit, at position s along the direction of unit vector \hat{e}_x by a magnitude x . [23, p.17]

The first derivative of the transverse displacement from the reference orbit, x', y' , is often also called angular displacement due to their relation to the angles of the particle trajectory displacement, α_x, α_y , in the horizontal and vertical planes, see *Figure 3.11* [23, p.17].

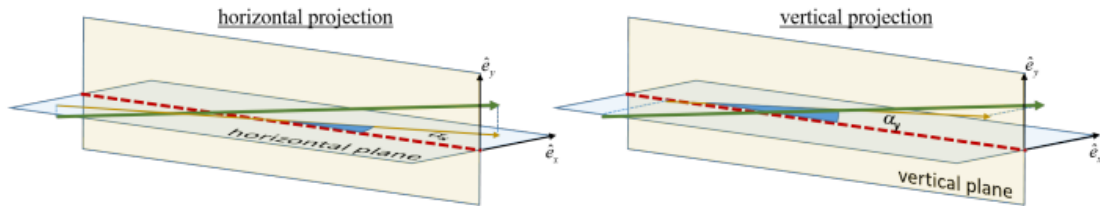


Figure 3.11: Angular displacements x', y' , represented as a projection to the horizontal and vertical planes along the reference orbit, at position s . [23, p.17]

When the transverse displacement x, y are smaller than the bending curvature ρ ($x, y \ll \rho$ and $x', y' \ll 1$), α can be safely approximated to be $\alpha_{x,y} \approx \tan \alpha_{x,y}$. This is called paraxial optics, which is a type of small angle approximation [27], also see *Figure 3.12*. With this approximation, it is now possible to represent each particle by its horizontal and vertical s -dependent transverse and angular displacement:

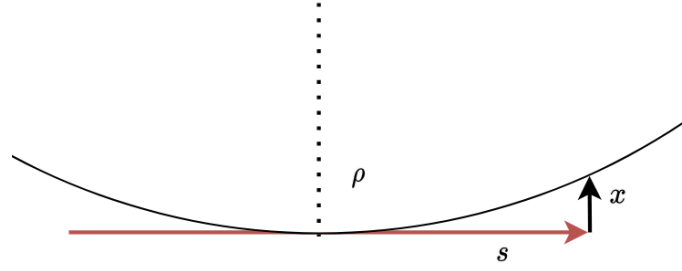


Figure 3.12: If the bending radius ρ is allowed to grow substantially, the transverse displacement x becomes much smaller. Thus it can be assumed that in large, or flat accelerators $x, y \ll \rho$ and the angular displacement to be $x', y' \ll 1$. Only if x', y' are much smaller than 1 can the angle $\alpha_{x,y}$ for the angular displacement can be assumed to be $\alpha_{x,y} \approx \tan \alpha_{x,y}$.

$$\mathbf{x} = \begin{bmatrix} x \\ x' \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y \\ y' \end{bmatrix} \quad (3.39)$$

where, vectors \mathbf{x}, \mathbf{y} are called trace space vectors, and they are defined in the transverse trace spaces of the phase space planes created by (x, x') and (y, y') , see left image of *Figure 3.13*. [23, p.17]

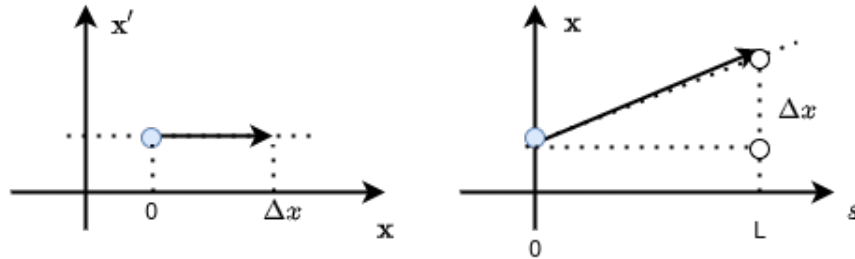


Figure 3.13: Representation of particle motion in a phase space diagram (left) and real space (right). The particle moves from the location 0 to L , while the transverse displacement x increases by Δx , but the angular displacement x' is constant.

3.3.1 Transformations induced by drift spaces, dipoles and quadrupoles

A drift space is a section of the accelerator where no magnetic fields are applied to the moving particle, on *Figure 3.10*, where a drift space is present between each quadrupole and dipole. Consequently, this implies that momentum of the particle ($p \Rightarrow \text{const}$) and the angular displacement ($x' \Rightarrow \text{const}$) is constant, which means that the angular displacement at the entry point of the drift space, x'_0 , stays the same during the entirety of the drift space until it exits with x' [23, p.18]:

$$x' = x'_0, \quad y' = y'_0 \quad (3.40)$$

However, as the particle travels, the transverse displacement x and y does change by the distance traveled in the drift space length L_d and by the angular displacement, see right image of *Figure 3.15*:

$$x = x_0 + L_d x', \quad y = y_0 + L_d y', \quad (3.41)$$

where, Δx can be approximated by $L_d x'$ due to the paraxial optics discussed above in the section.

Dipole magnets deflect each particle regardless of its angular or transverse displacement (see *Subsection 3.2.1 Dipole magnets*) thus, its effect can be approximated to a drift space and similar approximations can be made as in Equation (3.41) and *Figure 3.14*.

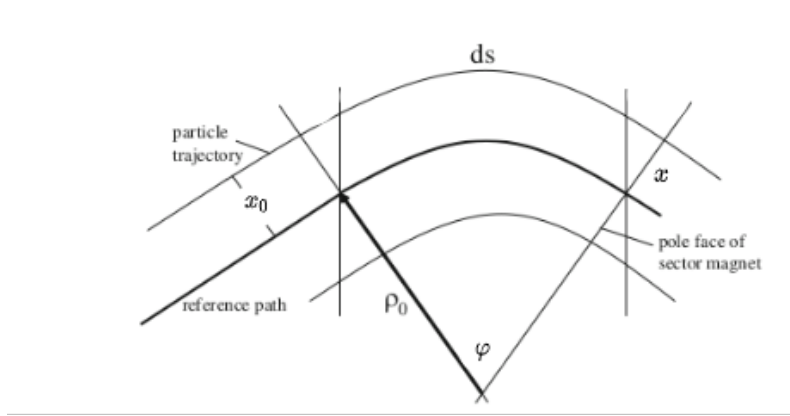


Figure 3.14: Sector dipole magnet bending without effect on the x and x' .

On the other hand, quadrupoles (in a thin lens approximation where $L_{quad} \ll f$), only the angular displacement is affected, see left image of *Figure 3.15*. The angular effect is denoted with $\Delta\alpha = \Delta x'$ and it depends on the focal length $f_{x,y}$ and the position displacement:

$$x' = x'_0 - \frac{x'_0}{f_x}, \quad y' = y'_0 - \frac{y'_0}{f_y}, \quad f_x = -f_y \quad (3.42)$$

where, f_x is the horizontal focal length and f_y is the vertical focal length. Consequently, in quadrupoles the transverse displacement stays unchanged:

$$x = x_0 \quad y = y_0. \quad (3.43)$$

3.3.2 Matrix formalism

Based on the above, it can be assumed that the first two magnetic multipoles can be described by a linear map, as the transverse and angular displacements are only affected linearly [23, p.18]. Therefore, the effect on the trace space vectors, 3.39, can be represented with a matrix multiplication for each plane:

$$\mathbf{x} = \mathbf{M}_x \mathbf{x}_0 \quad \mathbf{y} = \mathbf{M}_y \mathbf{y}_0 \quad (3.44)$$

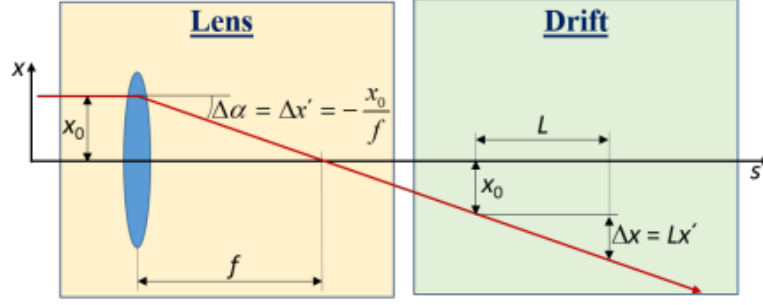


Figure 3.15: Effects of a quadrupole and drift space on the angular and transverse displacements [23, p.18]

This formalism is called the horizontal and vertical transfer matrix ($\mathbf{M}_{x,y}$) of two beam line elements. Which in turn means that it is now possible to represent a transfer matrix for a drift space or a dipole:

$$\mathbf{M}_{drift,x} = \mathbf{M}_{drift,y} = \mathbf{M}_{dipole,x} = \mathbf{M}_{dipole,y} = \begin{bmatrix} 1 & L \\ 0 & 1 \end{bmatrix} \quad (3.45)$$

Subsequently, using Equation (3.42) matrix formalism can be constructed for a quadrupole too (again, assuming thin lens approximation):

$$\mathbf{M}_{quad\ x} = \begin{bmatrix} 1 & 0 \\ n - \frac{1}{f_x} & 1 \end{bmatrix}, \quad \mathbf{M}_{quad\ y} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{f_y} & 1 \end{bmatrix}. \quad (3.46)$$

With the matrix formalism of a drift space, dipole and quadrupole magnet, it is now possible to describe the effects of a sequence of elements in any arbitrary accelerator. This gives the transfer function of the incoming trace space vector \mathbf{x}_0 to any outgoing \mathbf{x} in a beamline. For *Figure 3.10* where there are three quadrupoles, two dipoles and six drift spaces, the matrix formalism is as follows:

$$\mathbf{x} = \mathbf{M}_{drift,6} \cdot \mathbf{M}_{quad,3} \cdot \mathbf{M}_{drift,5} \cdot \mathbf{M}_{dip,2} \cdot \mathbf{M}_{drift,4} \cdot \mathbf{M}_{quad,2} \cdot \mathbf{M}_{drift,3} \cdot \mathbf{M}_{dip,1} \cdot \mathbf{M}_{drift,2} \cdot \mathbf{M}_{quad,2} \cdot \mathbf{M}_{drift,1} \cdot \mathbf{x}_0 \quad (3.47)$$

this can be further reduced to:

$$\mathbf{M}_{acc} = \prod_i \mathbf{M}_i \quad (3.48)$$

where the index i represents the individual elements, like *drift* or *quad*. [23, p.18]

3.3.3 Equations of motion

To be able to derive the final equations of motion, first, a transverse position vector of the moving particle at coordinate s has to be defined:

$$\mathbf{R}(s) = [\rho + x(s)] \cdot \hat{e}_x(s) + y(s) \cdot \hat{e}_y(s) \quad (3.49)$$

where, the (x, y) are the transverse local coordinates, ρ the bending curvature and $(\hat{e}_{x,y}(s))$ are the transverse local coordinate unit vectors (see *Figure 3.16*). The position of the local coordinate frame is also presented in *Figure 3.2*. Recall that, this frame traverses with the particle, thus its orientation depends on the current location, s , around the accelerator. [23, p.19]

Based on *Figure 3.16* it is possible to define the unit vectors \hat{e}_x and \hat{e}_s using polar coordinates in the horizontal plane:

$$\hat{e}_x = \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix}, \quad \hat{e}_\varphi = -\hat{e}_s = \begin{pmatrix} -\sin \varphi \\ \cos \varphi \end{pmatrix} \quad (3.50)$$

Subsequently, if there's in a change in s along the orbit, the local reference frame will rotate by an angle φ as it moves with the particle on the orbit:

$$d\varphi = -\frac{1}{\rho} ds \quad (3.51)$$

thus, the derivative of the unit vectors with bending radius ρ , in terms of longitudinal displacement $\hat{e}_{x,y}(s)$, can also be defined:

$$\frac{d}{ds} \hat{e}_x = \hat{e}'_x = +\frac{1}{\rho} \hat{e}_s, \quad \frac{d}{ds} \hat{e}_s = \hat{e}'_s = -\frac{1}{\rho} \hat{e}_x \quad (3.52)$$

Recall from Equation (3.2), that there are two derivative definition in this context, x' for the longitudinal coordinate and \dot{x} for the time derivative. It is possible to represent one from the other:

$$\frac{d}{dt} = \frac{d}{ds} \frac{ds}{dt} = v_s \cdot \frac{d}{ds} \quad (3.53)$$

where v_s represents the velocity of the particle along the reference orbit (which is approximated to be the speed of light c). In words, this implies that as the particles goes along the orbit with v_s velocity, the longitudinal coordinate s changes with it. Since

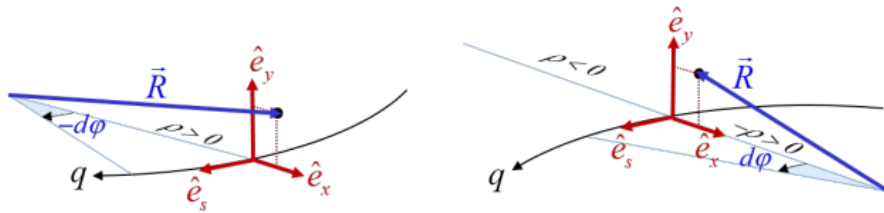


Figure 3.16: Local Coordinates of the transverse movement of a charged particle. With positive curvature: $\kappa = \frac{1}{\rho} > 0$, and with negative curvature: $\kappa = \frac{1}{\rho} < 0$. [23, p.19]

it takes times to travel along the orbit with constant velocity v_s , the displacement along s changes in relation to time elapsed. This effectively implies that that s is a time-like property as it has a strong relation to the distance travelled at speed v_s .

Assumptions to reduce complexity for the derivation

The above derivations so far concerned a generic approach to circular accelerators. However, it is possible to reduce to complexity in linear accelerators (or very large circular accelerators) with the following assumptions [23, p.20]:

- The change orbit curvature is slow, thus it is possible to neglect ρ' and ρ'' .
- The accelerator is "flat" and the orbit lies within the (x, s) plane only.
- Particles in the accelerator have individual velocities, but the longitudinal velocity are constant $\dot{s} = \mathbf{v} \Rightarrow \text{const}$
- The longitudinal velocity is much faster than the transverse velocity of the particles, $v_s \gg v_{x,z}$
- Paraxial optics takes effect, meaning the transverse displacement x, y is much smaller than the bending radius ρ . Furthermore, transverse motion does not affect longitudinal motion and vise-versa, which means all longitudinal components can be neglected and assume: $\mathbf{R}_s = \mathbf{R}'_s = \mathbf{R}''_s = 0$.

Final derivation of the equations of motion

Based on the above listed assumptions the transverse position, Equation (3.49), vector can be reduced to:

$$\mathbf{R} = r \cdot \hat{e}_x + y \cdot \hat{e}_y \quad (3.54)$$

where, $r \Rightarrow r(s) = \rho + x(s)$ (see *Figure 3.17*). From here on, the s dependence is always assumed, thus it is not explicitly stated anymore. The first derivative of 3.54 can now be defined:

$$\mathbf{R}' = x' \cdot \hat{e}_x + \frac{r}{\rho} \cdot \hat{e}_s + y' \cdot \hat{e}_y \quad (3.55)$$

as well as the second derivative:

$$\mathbf{R}'' = (x'' - \frac{r}{\rho^2}) \cdot \hat{e}_x + y'' \hat{e}_y + 2 \frac{x'}{\rho^2} \cdot \hat{e}_s \quad (3.56)$$

Now, using the assumption that design orbit is flat, the \hat{e}_s term can be safely ignored, since there is no transverse displacement along s if the orbit is flat.

To change the transverse position of a particle, \mathbf{R} , it needs to be affected by the Lorentz force, Equation (3.3):

$$\frac{d\mathbf{p}}{dt} = \gamma_r m_0 \frac{d^2 \mathbf{R}}{dt^2} = q(\mathbf{v} \times \mathbf{B}) \quad (3.57)$$

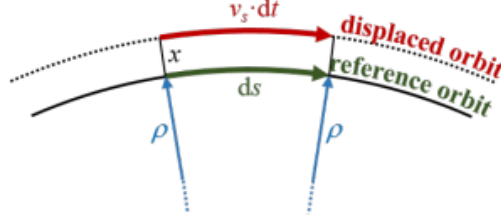


Figure 3.17: Orbit arc with radius $r = \rho + x$, where a particle is moving with constant v_s velocity along the offset-orbit with a horizontal offset x . [23, p.20]

Notice, that the second derivative of the position vector shows up with respect to time. Taking 3.53 and generalizing it based on geometric assumptions about the orbit, see Figure 3.17, it is possible to redefine ds for a single particle:

$$ds = v_s dt \left(\frac{\rho}{\rho + x} \right) = \left(\frac{\rho}{r} \right) v_s dt \implies \frac{d}{dt} = v_s \left(\frac{\rho}{r} \right) \frac{d}{ds} \quad (3.58)$$

which can be plugged into 3.56, to obtain:

$$\ddot{\mathbf{R}} = v_s \left(\frac{\rho}{r} \right)^2 \cdot \mathbf{R}'' \quad (3.59)$$

where:

$$\mathbf{R}'' = \left(x'' - \frac{r}{\rho^2} \right) \cdot \hat{e}_x + y'' \cdot \hat{e}_y \quad (3.60)$$

Again, based on the above assumptions where $v_x \ll v_s$ and $v_y \ll v_s$, the Lorentz force can be expanded by solving the cross product $\mathbf{v} \times \mathbf{B}$:

$$q(\mathbf{v} \times \mathbf{B}) = q \left[-v_s B_y \hat{e}_x + v_s B_x \hat{e}_y + (B_y v_x - B_x v_y) \hat{e}_s \right] \quad (3.61)$$

which can be further simplified because of the assumption that $\hat{e}_s = 0$ and by factoring v_s out, into:

$$q(\mathbf{v} \times \mathbf{B}) \approx q v_s (-B_y \hat{e}_x + B_x \hat{e}_y). \quad (3.62)$$

Now, substituting 3.59 into 3.57 one gets:

$$\gamma_r m_0 \ddot{\mathbf{R}} = q(\mathbf{v} \times \mathbf{B}) \quad (3.63)$$

then, expanding on the right side, by plugging 3.62 in:

$$\gamma_r m_0 \ddot{\mathbf{R}} = q v_s (-B_y \hat{e}_x + B_x \hat{e}_y) \quad (3.64)$$

Substituting, 3.59 into the above on the left side gives:

$$\gamma_r m_0 v_s \left(\frac{\rho}{r} \right)^2 \cdot \mathbf{R}'' = q v_s (-B_y \hat{e}_x + B_x \hat{e}_y) \quad (3.65)$$

Now the \mathbf{R}'' shows up, which was defined in 3.60, so further substituting yields:

$$\gamma_r m_0 v_s \left(\frac{\rho}{r} \right)^2 \cdot \left(x'' - \frac{r}{\rho^2} \right) \cdot \hat{e}_x + y'' \cdot \hat{e}_y = q v_s (-B_y \hat{e}_x + B_x \hat{e}_y) \quad (3.66)$$

At this stage assuming that $p = \gamma_r m v_s$ (see *Subsection 3.1.2 Lorentz Force and Magnetic rigidity*) yields:

$$p \left(\frac{\rho}{r} \right)^2 \cdot \left(x'' - \frac{r}{\rho^2} \right) \cdot \hat{e}_x + y'' \cdot \hat{e}_y = q v_s (-B_y \hat{e}_x + B_x \hat{e}_y), \quad (3.67)$$

and using *Figure 3.17*, where $r = \rho + x$ the left hand side denominator r becomes:

$$p \left(\frac{\rho}{\rho + x} \right)^2 \cdot \left(x'' - \frac{r}{\rho^2} \right) \cdot \hat{e}_x + y'' \cdot \hat{e}_y = q v_s (-B_y \hat{e}_x + B_x \hat{e}_y), \quad (3.68)$$

Then, dividing both sides by p :

$$\left(\frac{\rho}{\rho + x} \right)^2 \cdot \left(x'' - \frac{r}{\rho^2} \right) \cdot \hat{e}_x + y'' \cdot \hat{e}_y = \frac{q}{p} v_s (-B_y \hat{e}_x + B_x \hat{e}_y), \quad (3.69)$$

and again dividing both sides by $\left(\frac{\rho}{\rho + x} \right)^2$ gives the equation:

$$\left(x'' - \frac{x + \rho}{\rho^2} \right) \cdot \hat{e}_x + y'' \cdot \hat{e}_y = \frac{q}{p} \left(\frac{x + \rho}{\rho} \right)^2 (-B_y \hat{e}_x + B_x \hat{e}_y). \quad (3.70)$$

Subsequently, using only the first two magnetic multipoles:

$$B_x = -\frac{p_0}{q} k y, \quad B_y = \frac{p_0}{q} \left(\frac{1}{\rho} - k x \right) \quad (3.71)$$

where, $1/\rho$ and k are normalized to the reference momentum of the reference particle, p_0 . Inserting 3.71 into 3.70 then yields:

$$\left(x'' - \frac{x + \rho}{\rho^2} \right) \cdot \hat{e}_x + y'' \cdot \hat{e}_y = \frac{q}{p} \left(\frac{x + \rho}{\rho} \right)^2 \left[- \left(\frac{p_0}{q} \left(\frac{1}{\rho} - k x \right) \right) \hat{e}_x + \left(-\frac{p_0}{q} k y \right) \hat{e}_y \right]. \quad (3.72)$$

However, since in 3.70, p actually represents a non-reference particle, it is needed to approximate the difference between the reference and non-reference particle: $\frac{p_0}{p}$. This can only be done if the momentum difference is assumed to be smaller, such as $\Delta p = p - p_0$. This momentum difference is explained in relation to the focusing angle of a quadrupole, see *Subsection 3.2.2 Quadrupole magnets*. [23, p.20]

Furthermore, it is needed to also approximate the difference in the reciprocal value $\frac{1}{p}$ as well. This can be done with a first order Taylor-polynomial approximation evaluated at $p = p_0$:

$$\frac{1}{p} \approx \frac{1}{p_0} + \Delta p \frac{\partial(1/p)}{\partial p} \Big|_{p=p_0} = \frac{1}{p_0} \left(1 - \frac{\Delta p}{p_0} \right) \quad (3.73)$$

Plugging the above into 3.72 and solving for x'' and y'' then gives:

$$\begin{aligned} x'' &= \frac{x}{\rho^2} + \frac{1}{\rho} - \left(1 - \frac{\Delta p}{p_0} \right) \left(1 + \frac{x}{\rho} \right)^2 \left(\frac{1}{\rho} - kx \right) \\ y'' &= - \left(1 - \frac{\Delta p}{p_0} \right) \left(1 + \frac{x}{\rho} \right)^2 ky \end{aligned} \quad (3.74)$$

Then, eliminating all non-linear terms, only using the lowest order elements and reordering, gives the linearized equations of motion along the longitudinal coordinate s :

$$\begin{aligned} x''(s) + \left(\frac{1}{\rho^2(s)} - k(s) \right) \cdot x(s) &= \frac{1}{\rho} \frac{\Delta p}{p_0} \\ y''(s) + k(s) \cdot y(s) &= 0 \end{aligned} \quad (3.75)$$

To reiterate the meaning of the terms, they are listed here once again:

1. x and y : Transverse displacement along the unit vectors \hat{e}_x, \hat{e}_y . These were defined in *Subsection 3.2.2 Quadrupole magnets*.
2. x'' and y'' : Second derivative of the transverse displacement. These terms are first introduced when the transverse position vector \mathbf{R} is differentiated, see *Subsection 3.3.3 Equations of motion*.
3. ρ : Bending radius of a dipole magnet. This was first mentioned in *Subsection 3.2.1 Dipole bending curvature*. The bending radius ρ for a circular accelerator is the same as the bending radius for a dipole magnet.
4. k : Quadrupole multipole strength. This variable defines the effect of a generated magnetic field in a magnet onto the moving particle. This was first introduced in *Subsection 3.1.3 Multipole expansion of a magnetic field*.

$$k = -\frac{qB_0}{p} \quad (3.76)$$

5. p : Momentum of the particle. This first showed up in *Subsection 3.1.2 Lorentz Force and Magnetic rigidity*.

$$p = mv \quad (3.77)$$

Chapter 4

Current Trajectory Control Methods for the Electron Beam Line

The AWAKE electron line has been built to be a robust and configurable transfer line. It allows for the testing of new controller designs and is often used as a testbed for new control algorithms which are ultimately destined to be used at different accelerators at the CERN complex. To provide a foundation for a new controller design, in this chapter, four different state of the art controllers are going to be explained. First, a method which uses Singular Value Decomposition (SVD) is introduced, then, a hidden cost function minimization approach, a more recent deep reinforcement learning (deep-RL) controller, and finally the newest concept, an iterative version of the Linear Quadratic Regulator (LQR). But before that, the chapter begins with a description of the control problem and how the performance of the systems can be measured and compared.

4.1 Trajectory Correction Control Problem

The control problem being tackled in this project is to control the trajectory of a particle beam inside the AWAKE electron beamline. Physical description of the beamline was introduced in *Subsection 2.3.1 Electron RF gun and beam line* and now the corresponding control problem is introduced. In particle accelerators the trajectory control can be represented as a multi-objective minimization problem (in order of importance):

1. Minimization of the transverse beam size.
2. Minimization of the beam emittance.
3. Minimization of the strength of the corrector magnets.

Note, that transverse dynamics was derived in *Chapter 3 Modeling of Particle Motion* and from this point on, the concepts introduced in that chapter are going to be required for understanding.

In the following paragraph, each of the objectives is described in more detail.

1) Minimization of the transverse electron beam size: The reduction of the transverse beam size can be represented as a minimization of the transverse displacement x and y . This minimization can be broken down into two sub-objectives:

1. Steer the beam towards the desired orbit and;
2. Focus the beam to minimize the beam spread.

Sub-objective 1) can be achieved by using corrector dipoles, whereas the beam spread is reduced with quadrupole magnets instead. An example of beam size minimization is shown in *Figure 4.1*.

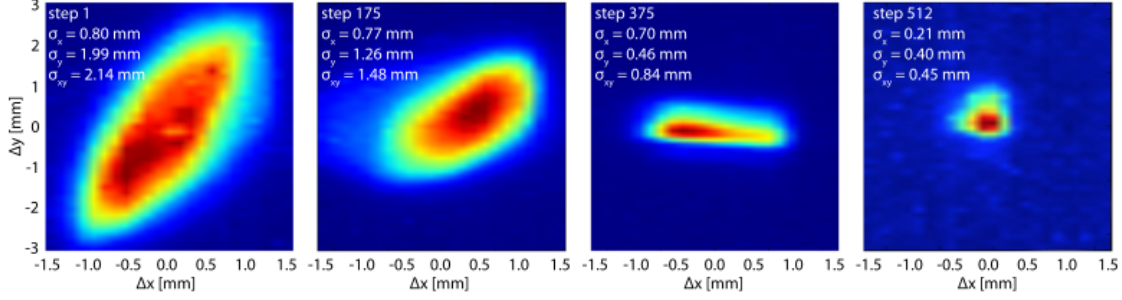


Figure 4.1: Beam size reduction over subsequent minimization steps. The $x - y$ axis represents the transverse $x - y$ displacement. [28]

2) Minimization of the beam emittance: Beam emittance is, in a sense, a function of the transverse displacement. It describes the distribution of the particles inside the beam in the phase space. The higher the beam emittance, the higher the displacement of either x or y or both. It is a collection of parameters which form an ellipse in phase space. An example is presented in *Figure 4.2*.

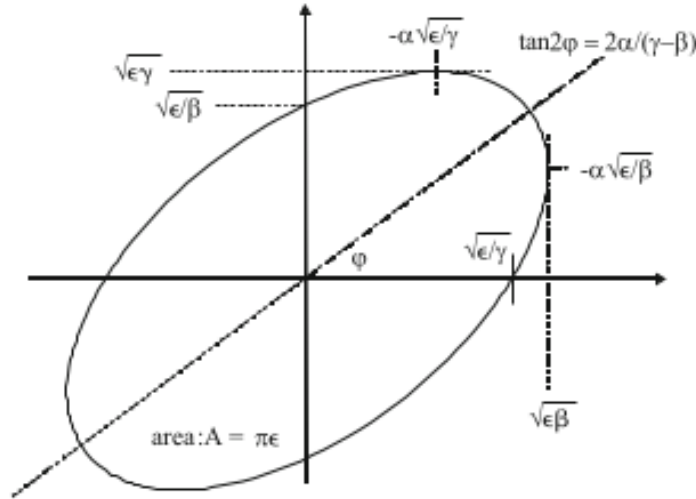


Figure 4.2: Phase space emittance. It is a collection of parameters that form an ellipse with area $A = \pi\epsilon$. [22, p.219]

3) Minimization of the strength of the corrector magnets: Albeit not as important as the other two minimization goals, reducing the required energy to correct the beam is always needed. Generally, this is required to create a system where the beam size and emittance is minimized with magnets which do not "compete" against each other with higher and higher currents.

4.1.1 Orbit displacement comparison

If sufficient measurement devices are accessible in an accelerator, the x and y transverse displacement against a reference orbit can be measured directly. This is done by:

$$x_{error} = x_{measured} - x_{reference} \quad (4.1)$$

where, the goal is to minimize the x_{error} as fast as possible.

In the AWAKE electron beam line this does in fact provide a sufficient comparisons basis between controller performances.

4.1.2 Root-Mean-Square-Error (RMSE) value of trajectory following performance

Another common way of measuring the performance of control algorithms is to define the RMSE difference between the design trajectory and the measured deviation. This can apply for both of the planes, horizontal (X) and vertical (Y), respectively:

$$\begin{aligned} X_{\text{RMSE}}(n) &= \sqrt{\frac{1}{m} \sum_{i=1}^m (\text{BPM}_i(n) - \text{BPM}_{i,o})^2} \\ \text{or} \\ Y_{\text{RMSE}}(n) &= \sqrt{\frac{1}{m} \sum_{i=1}^m (\text{BPM}_i(n) - \text{BPM}_{i,o})^2} \end{aligned} \quad (4.2)$$

where, n is the current iteration, BPM_i is the Beam Position Measurement at the i -th measurement device, and m being the number of elements to taken the average over. In the following sections, this formalism is used to define the performance of the control algorithms.

4.2 Multi-objective Hidden Cost Function Minimization

In [28], a multi-objective hidden cost function based feedback algorithm is introduced to control the trajectory of the beam via BPMs while at the same time reducing beam emittance. First, the beam trajectory minimization is done with a hidden cost function approach, where an iterative model-independent feedback algorithm is applied. This algorithm was designed for highly noisy systems where an analytical approach to a model is either very difficult to derive or simply not possible achieve. The proposed algorithm is as follows. Using a very noisy measurement \hat{C} of the unknown cost function, C , one can create a set of parameters p_i which can be adjusted to minimize C :

$$p_i(n+1) = p_i(n) + \Delta_t \sqrt{\alpha \omega_i} \cos(\omega_i n \Delta_t + k \hat{C}(n)) \quad (4.3)$$

where, ω is a unique dithering frequency, α is the dither size gain. k is the feedback gain. Then by the setting the parameter Δ_t , 4.3 is a difference approximation of:

$$\frac{dp_i(t)}{dt} = \sqrt{\alpha\omega_i} \cos\left(\omega_i t + k\hat{C}(\mathbf{p}, t)\right). \quad (4.4)$$

This, then in turn results in a minimization of the initial cost function C . This approach is dependent on the assumptions that the system follows an average dynamics:

$$\frac{d\bar{\mathbf{p}}}{dt} = -\frac{k\alpha}{2} \nabla_{\mathbf{p}} C(\mathbf{p}, t). \quad (4.5)$$

The dual minimization approach is visualized on the model of the electron beam line, see *Figure 4.3*. This approach showed promising results as the algorithm managed to achieve convergence within 30 iterations. Results are shown in *Figure 4.4*.

Then, applying the same algorithm to three quadrupoles and two solenoids to control the beam size as it comes out of the electron source, 4.3 was set the cost to the beam size:

$$\sigma_{xy} = \sqrt{x^2 + y^2} \quad (4.6)$$

Then combining the two, the authors solve the multi-objective optimization problem of:

$$\begin{aligned} \min_{\mathbf{p} \in \mathbb{P}} \{f_1(\mathbf{p}, t), f_2(\mathbf{p}, t)\}, \\ f_1(\mathbf{p}, t) = w_1 X_{\text{RMSE}}(\mathbf{p}, t), \quad f_2(\mathbf{p}, t) = w_2 \sigma_{xy}(\mathbf{p}, t) \end{aligned} \quad (4.7)$$

where, \mathbf{p} is the total 15 parameters adjusted by the algorithms.

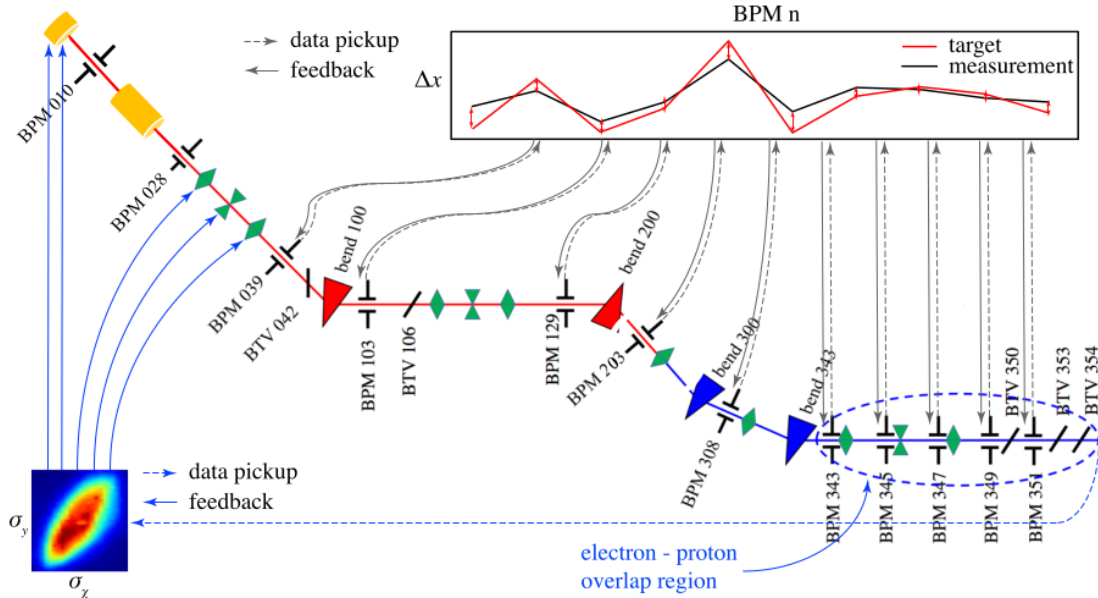


Figure 4.3: Multi-objective optimization of the beam orbit and emittance. In this approach there are two control loops acting on magnets. One is tasked with reducing the beam size via the quadrupoles and solenoid magnets at the RF gun injection, whereas the other one is tasked with keeping to the design orbit by change corrector magnet currents. [28]

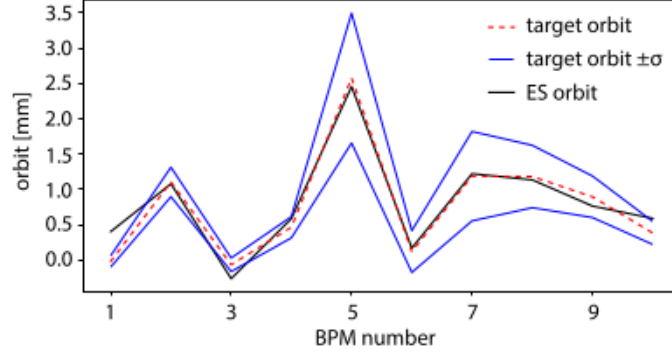


Figure 4.4: Beam orbit deviation at the i -th BPM. In this paper, the orbit deviation is measured in micro meters [mm]. [28]

4.3 Singular Value Decomposition (SVD) Based Approach

The SVD based approach to beam alignment proposes a very fast and efficient algorithm in minimizing the RMS of the beam at each BPM. If an orbit response matrix (ORM) \mathbf{A} , is known about the accelerator one can solve the equation:

$$\Delta \mathbf{x} = \mathbf{A} \cdot \theta \quad (4.8)$$

to compute the required m -th corrector strength, θ at the m -th BPM measurement $\Delta \mathbf{x}$, and the orbit response matrix, \mathbf{A} is defined as [21, p.87] [29]:

$$\mathbf{A}_{i,k} = \frac{\sqrt{\beta_m \beta_k}}{2 \sin(\mu/2)} \cos(|\phi_m - \phi_k| - \mu/2). \quad (4.9)$$

where, β, μ, ϕ are all Twiss parameters of the accelerator elements. At this stage in the report, Twiss parameters are not described, because first particle modelling needs to be introduced. This is done in *Chapter 3 Modeling of Particle Motion* and subsequently, Twiss parameters are described in more detail in *Chapter 8 Electron Beam Line Simulation and Controller Testing*. At this stage it is sufficient to understand that Twiss parameters are properties of magnets in the accelerator and are used to define and measure the ORM.

The ORM describes the relation between the corrector currents applied and the BPMs measurements. This matrix can be measured by individually applying currents to the magnets and measuring their effect on the orbit displacement. At HERA, this was done with more than 300 000 measurements [29]. In theory this method provides a deterministic response of the accelerator to any beam with a well defined minimum and maximum corrector currents. However, in practice, due to BPMs measurement issues and with outside disturbances the ORM often does not give a precise representation. [29]

In cases where there are low number of correctors and BPMs devices, the ORM based approach can be a very powerful, fast and robust way of correcting the beam orbit. Therefore, (4.8), can be rearranged and solved for θ by inverting the ORM:

$$\Delta \mathbf{x} \mathbf{A}^{-1} = \theta \quad (4.10)$$

This matrix inversion can be computed with well defined algorithms such as singular value decomposition, where the ORM \mathbf{A} can be decomposed into three matrices:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* \quad (4.11)$$

where, \mathbf{U} is an $m \times m$ complex unitary matrix, $\mathbf{\Sigma}$ is a $m \times m$ rectangular, diagonal matrix and \mathbf{V} is an $n \times n$ complex unitary matrix. The operator $*$ represents complex conjugate, however, in this case transpose is sufficient, because the \mathbf{A} matrix is not represented in the complex plane. Therefore, if $\mathbf{\Sigma}$ is now shown as a diagonal matrix, one gets:

$$\mathbf{A} = \mathbf{U} \cdot \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ & \dots & \dots & \\ 0 & 0 & \dots & \sigma_n \end{pmatrix} \cdot \mathbf{V}^t \quad (4.12)$$

where, the σ entries are the singular values. Therefore, the solution 4.10 can be expanded to:

$$\theta = \mathbf{A}^{-1} \cdot \Delta \mathbf{x} = \mathbf{V} \cdot \begin{pmatrix} 1/\sigma_1 & 0 & \dots & 0 \\ 0 & 1/\sigma_2 & \dots & 0 \\ 0 & \dots & \dots & \\ 0 & 0 & \dots & 1/\sigma_n \end{pmatrix} \cdot \mathbf{U}^t \cdot \Delta \mathbf{x} \quad (4.13)$$

which is the unique solution to the required corrector currents θ for a given observation of BPMs readings $\Delta \mathbf{x}$. Additionally, if in 4.12 one or more of the singular values are zero, the decomposition might not have a solution. However, the general practice is to still solve the SVD analytically and then to replace the corresponding $1/\sigma$ with a zero in 4.13. This way a solution can be still found. This essentially means, that if the problem is represented from the least-square sense, the following residual, r is minimized:

$$r = |\mathbf{A} \cdot \theta - \Delta \mathbf{x}| \quad (4.14)$$

from which the solution to θ can be obtained. This also gives the minima for $|\theta^2|$, which is the minima for the RMS strength of the correctors.

4.3.1 Feedforward controller

When examining 4.10 one can notice that the solution takes the form of a feedforward controller:

$$\mathbf{y} = \mathbf{D}\mathbf{u} \quad (4.15)$$

where, \mathbf{y} is the output of the system, which is the measurement $\Delta \mathbf{x}$, the feedforward matrix \mathbf{D} is the ORM of the accelerator \mathbf{A} and the control input \mathbf{u} is the corrector currents θ .

In practice, SVD is a good approach for fast and reliable computation for the corrector currents, however, since the ORM is a deterministic approach to the model of the accelerator, meaning the ORM always responds with the same output of the displacement to a set current in any point of time. This implies that over longer periods of time, the ORM based approach might not account for drift in magnets and outside disturbances which did not exist at the time of the measurement.

Nonetheless, results show that the SVD approach can be used and on the AWAKE electron beam line it can produce convergence without 3 iterations, see *Figure 4.5*.

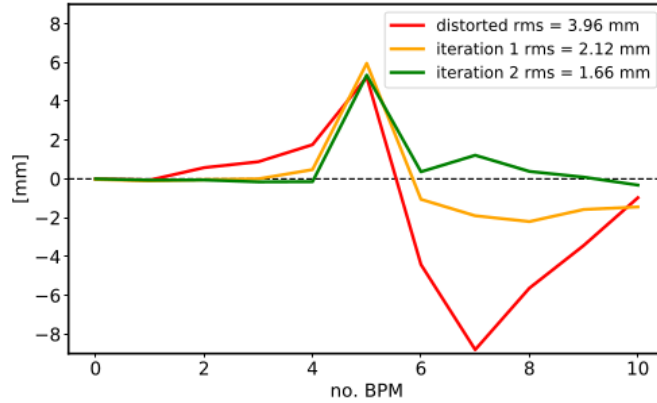


Figure 4.5: Results of an SVD algorithm RMS correction on the AWAKE electron beamline, where the algorithm corrects the RMS of the beamline to below $2\mu\text{m}$ in three iterations. [30]

4.4 Deep Reinforcement Learning (deep-RL)

Another proposed approach to the trajectory correction problem is to use Reinforcement Learning (RL) with a deep neural network to estimate a non-linear system model. In other words, the RL agent learns an optimal policy for the control problem. Reinforcement Learning with an optimal control policy is analogous to classic optimal control, such as LQR or MPC. [30]

The authors in [30], trained a sample efficient model-free RL model on the AWAKE electron beam line to control the trajectory of the electron beam. The proposed setup of the feedback controller is presented in *Figure 4.6*.

The expected reward, or Q -policy is calculated as such:

$$Q^\pi(s, a | \theta^Q) = \mathbb{E}_\pi \sum_{k=0}^N \gamma^k r_{i+k+1} | s_i = s, \quad a_i = a \quad (4.16)$$

where, agent's action is denoted as a assuming that it is performed in state s , the current episode is denoted as π , N is the number of states from $s_i = s$ until the very last state γ . A discount factor is also included, r . Furthermore, θ^Q is the network parameters. A corresponding value function is also defined:

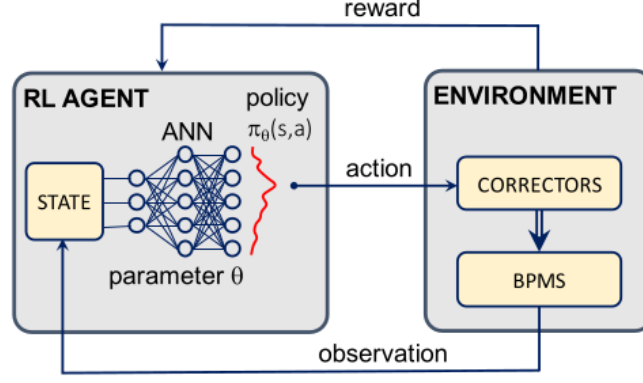


Figure 4.6: deep-RL feedback loop for the AWAKE trajectory correction problem. A measured state is mapped to an action with an Artificial Neural Network (ANN). [30]

$$V^\pi(s | \theta^v) = \mathbb{E}_\pi \sum_{k=0}^N \gamma^k r_{i+k+1} | s_i = s \quad (4.17)$$

The value function measures the overall expected reward and it closely relates to the Q -function.

In this paper the RL implementation uses a PER-NAF network show in *Figure 4.7*. It is a neural network with the shown architecture, where the input layer on the left acquires the state s followed by two hidden dense layers with 32 nodes, each.

The outputs of the system are:

$$\mu(s, a | \theta^\mu), \quad P(s, a | \theta^P), \quad V(s | \theta^V) \quad (4.18)$$

The further details of mathematics behind the RL agent is not described anymore in this project. The paper can be found in [30].

4.4.1 Simulation Environment

The RL agent applied in this paper is ran in OpenAI Gym. [31] The simulation is written in python and uses the OpenAI RL trainer to train the model and to test it. The AWAKE

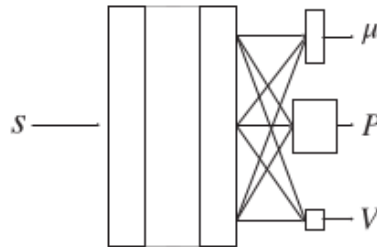


Figure 4.7: The NAF architecture used in this paper. [30]

parameters are parsed in from MAD-X which is a CERN specific simulation encoding where all the accelerators are described numerically. MAD-X is further described in *Section 8.1 Methodical Accelerator Design (MAD)*. This simulation was provided to this project as a basis for the controller implementation. Therefore, a more in-dept explanation can be found in *Chapter 8 Electron Beam Line Simulation and Controller Testing*.

4.4.2 Deployment to AWAKE and LINAC4

This RL agent in this paper was both deployed to AWAKE and to LINAC4 during the LS2 period to test out the performance of the algorithm. The results can be seen in *Figure 4.8*.

The model-free RL agent in the end proved to be a fast and reliable algorithm in correction the trajectory of the beam in the AWAKE electron beamline. Further iterations of this approach are expected to applied at CERN in the future for other accelerator control problems.

4.5 Iterative LQR (iLQR)

The most recent state of the art controller continues the work of the deep-RL system. In this approach the authors estimated the model of the system with an ANN from Tensorflow2 and applied an iterative version of the LQR controller onto the ANN produced non-linear plant. The proposed algorithm was then applied to the AWAKE electron beamline. Unfortunately, at this point this cannot be sourced here, as the work is still CERN internal and the material is hosted on internal networks only. It is foreseen that in the future this research is formalized into a scientific paper.

The iterative LQR technique in this paper, is a method to apply the LQR infinite horizon approach to a non-linear system. At each time step the algorithm would compute a new feedback law and apply it around a linearized point. This approach is not widely document in the control theory world, thus only source in this thesis is regarding biological movement [32].

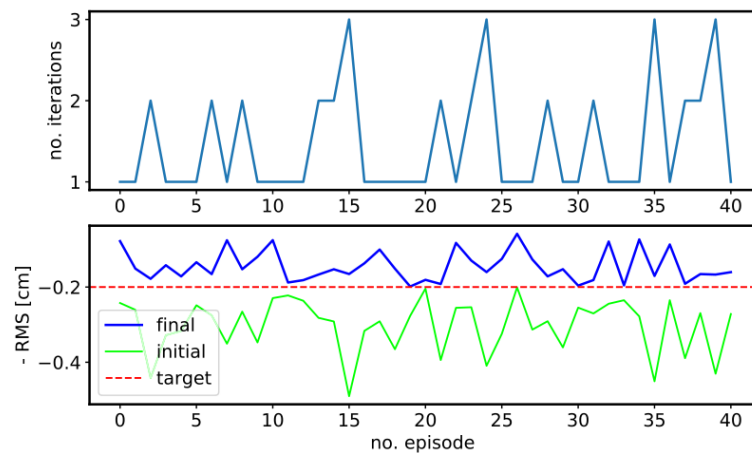


Figure 4.8: Results of a 35 iteration trained agent to correct the RMS of the beam to less than $2\text{ }\mu\text{m}$ [30]

4.6 Conclusion

In this chapter different controller designs have been presented. There are two approaches which all of the controllers above follow:

1. the controllers are model-free
2. the controllers are feedforward controller without feedback.

Thus it can be concluded that a model-based feedback approach could be an option for simpler transfer lines and LINACs.

Chapter 5

Requirements, Limitations and Control Objective

The state of the art analysis in *Chapter 4 Current Trajectory Control Methods for the Electron Beam Line* showed that the current controller designs are either model-free, or estimated with a deep neural network, or estimated to be using a generic system dynamic model to minimize a hidden cost function. It has been widely stated in [30] and [28] that model based approaches are difficult to achieve due to the complexity of the modeling. However, at the time of the writing of this report, for the AWAKE electron beam line, there were no clear attempts have yet been made with modern approaches, such as optimal control.

Therefore, this project will attempt to derive a model for the AWAKE electron beam line and design a sufficient controller which achieves bending correction for the beam trajectory within the beam line. For this, a set of requirements are now stated along with the limitations of this project. The specific control problem is also stated in words at the end

5.1 Requirements

- Model a sample system of the AWAKE electron beam line.
 - For a model based approach a model is going to created of the system based on the particle modeling outlined in *Chapter 3 Modeling of Particle Motion*.
- Create a model based optimal controller with the objectives:
 - Minimize the trajectory displacement to within less than $2\mu\text{m}$ and;
 - the beam size.
- Simulate the AWAKE electron beam line.
- Compare the controller performances to the results stated in the *Chapter 4 Current Trajectory Control Methods for the Electron Beam Line*:
 - SVD approach;
 - deep-RL;

5.2 Limitations

There are a variety of limitations to this project. These are now listed in the following subsections.

5.2.1 Lack of access to physical hardware

At the time of the writing of this thesis, the CERN complex had begun emerging from LS2, and beams are circulating in the systems again. This means that, the testing phase has already finished and no new control system applications are implemented directly onto accelerator hardware. Therefore, for this project only a simulation based approach is viable.

5.2.2 Simulation was designed around the deep-RL paper

The simulation that was provided for this thesis was built specifically for the deep-RL network, which was built inside an neural network testing framework. This means that there is a need for a simulation rewrite. This issue is further discussed in *Chapter 8 Electron Beam Line Simulation and Controller Testing*.

5.3 Control Objective

The objective of this thesis is to show that model-based control methods are capable at controlling a simplified model of the AWAKE particle transfer line with corrector dipoles for a given trajectory correction problem. This can be verified on the simulation provided from the deep-RL paper, since it describes the beam optics of the transfer line. This is to prove viability for further iterations for more complex model based approaches.

Chapter 6

System Model and Discretization

In this chapter, a proposed system model for the electron beam line is introduced. Assumptions about the system behaviour are also listed. Then the EOMs for the individual components in the model are described. Afterwards, the system is setup into a state space model which is subsequently, discretized. In the end the chapter proposes a controller setup for the system, which is the modelled in the subsequent chapter.

6.1 Proposed AWAKE Electron Beam Line System Model

The AWAKE electron beam line was introduced in *Subsection 2.3.1 Electron RF gun and beam line*. Now a simplified model is presented. As mentioned in *Section 5.3 Control Objective*, the control objective is not right away control the entire AWAKE electron beam line, but rather to prove that modern control techniques are a viable approach in the search of advanced control paradigms for linear accelerators. Thus, a model is now proposed for which the subsequent state space model is derived for, see *Figure 6.1*.

The flow of the system is approximately as follows:

1. The electrons exit from the RF gun into a vacuum tube, with initial conditions of the displacement $x(0)$ and $x''(0)$.
2. The system is corrected with the first corrector dipole at $s = j$.
3. BPMs measurement is performed, returning the measured state at $s = j$.
4. Particles then traverse a drift space where no actuation is present, that is no magnetic fields are acting on the particle.
5. Another correction after the drift space is performed at $s = j + \Delta s$.
6. That is subsequently measured again as the output of the particle displacement at $s = j + \Delta s$.
7. Steps 4-5-6 are repeated again.

Where, j is defined to be an initial location along s , and $j + \Delta s$ is the location along s at the subsequent actuation and measurement location, see *Figure 6.2*.

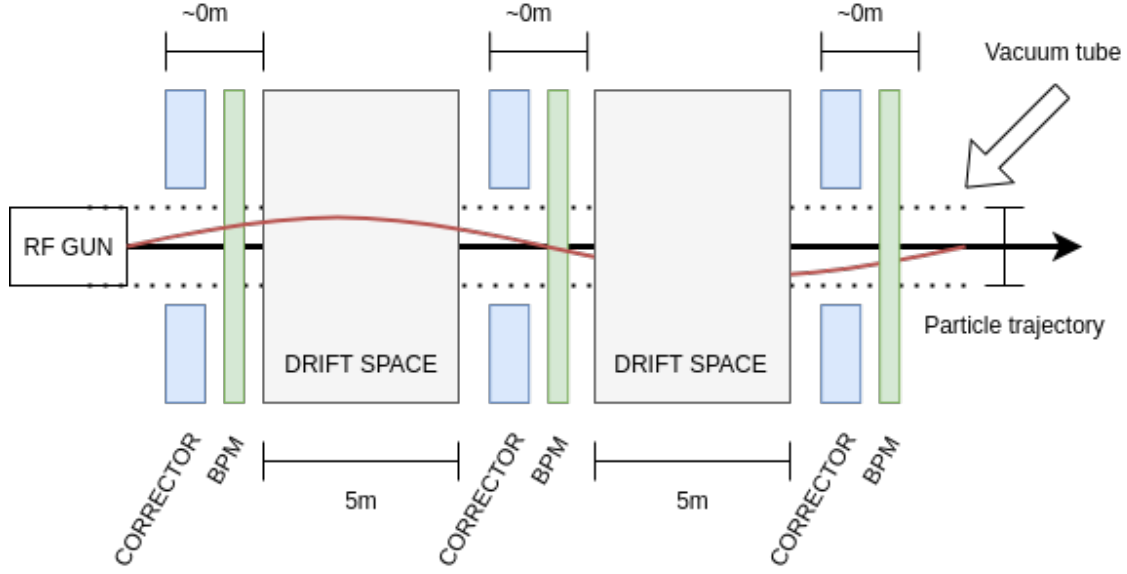


Figure 6.1: A simplified electron beam line, with three corrector dipoles, three BPMs and three drift spaces. Note how the corrector and BPM are signified to be 0 m apart. This is because the measurement is effectively performed at the actuation. This was discussed in *Subsection 2.3.1 Electron RF gun and beam line*

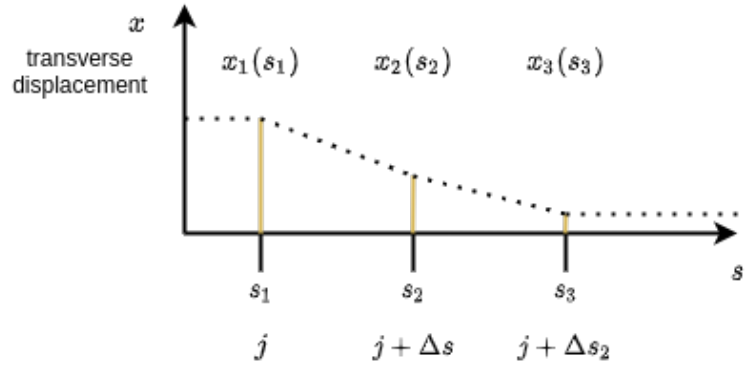


Figure 6.2: Transverse coordinates of the transverse displacement at s_1 , s_2 and s_3 . At each location a magnet and BPM is assumed.

6.2 EOMs for Individual Drift Spaces, Dipoles and Quadrupoles

To be able to model the system, the equations of motions defined in Equation (6.1) needs to be simplified to individual contributing effects of the magnets. The EOMs with the combined effect of dipoles and quadrupoles are presented here again:

$$\begin{aligned} x''(s) + \left(\frac{1}{\rho^2(s)} - k(s) \right) \cdot x(s) &= \frac{1}{\rho} \frac{\Delta p}{p_0} \\ y''(s) + k(s) \cdot y(s) &= 0 \end{aligned} \quad (6.1)$$

As per the definition of the system model (see *Section 6.1 Proposed AWAKE Electron Beam*

Line System Model), only corrector dipoles and drift spaces are considered, therefore, in the following subsection only the drift space and dipole effects are derived for the EOMs.

6.2.1 EOM: Drift space

In a drift space, both the bending from the dipole $\frac{1}{\rho} = 0$ and the $k = 0$ is set to zero which yields:

$$\begin{aligned} x'' &= 0 \\ y'' &= 0 \end{aligned} \tag{6.2}$$

This is the simplest case of the EOM. This is due to the fact, that the a particle passing through drift space does not change its momentum and it is not bent or focused by dipoles and quadrupole magnetic fields. This was further discussed in *Subsection 3.3.1 Transformations induced by drift spaces, dipoles and quadrupoles*.

6.2.2 EOM: Corrector dipole effects

In a system which is only affected by dipoles, the quadrupole strength to zero $k = 0$ must be set to zero. By doing this in 6.1 then the equations of motion reduce to:

$$\begin{aligned} x''(s) + \frac{1}{\rho^2(s)} \cdot x(s) &= \frac{1}{\rho} \frac{\Delta p}{p_0} \\ y''(s) &= 0 \end{aligned} \tag{6.3}$$

However, additionally, since there is no momentum change introduced by the dipoles, the right hand side of the equation, $\frac{\Delta p}{p_0}$, is also set to zero:

$$\begin{aligned} x''(s) + \frac{1}{\rho^2(s)} \cdot x(s) &= 0 \\ y''(s) &= 0 \end{aligned} \tag{6.4}$$

Note that, 6.4 it appears that there are no possibilities for the control input. However, ρ is not a constant, it depends on the magnetic field generated by the dipole magnet. It can be expanded to:

$$\frac{1}{\rho} = \kappa \tag{6.5}$$

where, κ is the dipole strength:

$$\kappa = \frac{q}{p} B_0 \tag{6.6}$$

where, B_0 is the magnetic field, which is excited by the current passing through a constant number of coils:

$$B_0 = \mu_0 \frac{nI}{h} \tag{6.7}$$

Thus, the control input into the dipole affected EOM is the current, I . Furthermore, if there are multiple dipoles in a system, the current depends on the longitudinal displacement $I(s)$. The whole derivation of the dipole strength can be found in *Subsection 3.2.1 Dipole magnets*.

6.3 State Space Model

Now that the EOM for the individual magnet effects have been derived, it is now possible to construct the state space model of a two dipole system, shown in *Section 6.1 Proposed AWAKE Electron Beam Line System Model*. This means that the, the EOM has to in the form:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t)\end{aligned}\tag{6.8}$$

where, $\dot{\mathbf{x}}(t)$ is the next state of system based on the current state \mathbf{x} at time t , the system model is A and the input is defined as $\mathbf{u}(t)$. The B matrix maps the inputs to the correct states. Furthermore $\mathbf{y}(t)$ is the output of the system, where C is the output matrix which maps the states to \mathbf{y} , and D is the feedforward matrix. Feedforward control was already mentioned in *Section 4.3 Singular Value Decomposition (SVD) Based Approach*.

Recall the final equations of motion for a dipole are:

$$\begin{aligned}x''(s) + \frac{1}{\rho^2(s)} \cdot x(s) &= 0 \\ y''(s) &= 0\end{aligned}\tag{6.9}$$

Since 6.9, is a continuous, s dependent, second order ODE, to be able to write it up in a state space form, it has to be converted into a first order ODE.

To being with, 6.9 can be rearranged by moving everything on the right hand side except $x(s)''$:

$$x''(s) = -\frac{1}{\rho^2(s)} \cdot x(s)\tag{6.10}$$

From Equation (6.6), it is known that $1/\rho$ is equal to κ which can be substituted into the above, as such:

$$x''(s) = -\left(\frac{qB_0}{p}\right)^2(s) \cdot x(s)\tag{6.11}$$

since, the momentum p and the charge q are constant and are independent of s , a helper variable can be defined $\alpha = \left(\frac{q}{p}\right)^2$. Substituting α into 6.11:

$$x''(s) = -\alpha B_0^2(s) \cdot x(s)\tag{6.12}$$

Furthermore, it also known that to control the dipole magnet and its angle of deflection, B_0 can modulated with a current input, I . Therefore, the input $u(s)$ can now be defined to be $u(s) = B_0^2(s)$. Plugging $u(s)$ in gives:

$$x''(s) = -\alpha u(s) \cdot x(s) \quad (6.13)$$

At this point, two helper variables has to be defined:

$$\begin{aligned} z_1(s) &= x(s) \\ z_2(s) &= x'(s) \end{aligned} \quad (6.14)$$

which can be represented in a column vector form with its derivative:

$$\mathbf{z}(s) = \begin{bmatrix} z_1(s) \\ z_2(s) \end{bmatrix}, \quad \mathbf{z}'(s) = \begin{bmatrix} z_1'(s) \\ z_2'(s) \end{bmatrix} \quad (6.15)$$

The helper variables z_1, z_2 , can now be then substituted into 6.13:

$$z_2'(s) = -\alpha z_1(s)u(s) \quad (6.16)$$

However, this is not yet in the form of a state space equation because z_2 does not appear on the right hand side of the equation. This can be solved by introducing another equation:

$$z_1'(s) = z_2 \quad (6.17)$$

and then placing it into a system of equation with 6.16:

$$\begin{aligned} z_1'(s) &= z_2 \\ z_2'(s) &= -\alpha z_1(s)u(s) \end{aligned} \quad (6.18)$$

Then, by introducing matrices A and B in the form:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ -\alpha & 0 \end{bmatrix}, \quad (6.19)$$

it is possible to rewrite 6.18 into the state space formalism of 6.8:

$$\begin{bmatrix} z_1'(s) \\ z_2'(s) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_1(s) \\ z_2(s) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\alpha & 0 \end{bmatrix} \begin{bmatrix} z_1(s) \\ z_2(s) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}, \quad (6.20)$$

which can be reduced to the compact state space form, with dependence on s :

$$\mathbf{z}'(s) = \mathbf{A}\mathbf{z}(s) + \mathbf{B}\mathbf{z}(s)\mathbf{u}(s) \quad (6.21)$$

6.3.1 Remark on longitudinal dependence in state space formalism

By definition, state space representation is a form of first order differential equations as a set of relations in terms of input, output and system variables. Generally, the systems that are to be controller using state space representation are dependent on t as noted in Equation (6.8).

However, since beam dynamics usually works with ultra-relativistic particles (see *Subsection 3.1.2 Lorentz Force and Magnetic rigidity*) the time dependence is generally replaced with a longitudinal time-like dependence on s . The relation between the change in t and change in s , are dependent on the velocity of the particle (see *Subsection 3.3.3 Equations of motion*):

$$\frac{d}{dt} = \frac{d}{ds} \frac{ds}{dt} = v_s \cdot \frac{d}{ds} \quad (6.22)$$

Therefore, this allows the system to be represented in the continuous state space form. However, as it is described in *Section 6.4 Model Discretization* this introduces some problems to the system. The issues are described in that section.

6.3.2 System Linearity

It is important to note here that the system derived in 6.21 does not entirely hold the linear state space model form, instead it takes the form of:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + (B\mathbf{x}(t) + B)\mathbf{u}(t) \quad (6.23)$$

which is the definition of a bilinear system. The bilinearity arises due to the interaction between the state vector x and the system input matrix $Bx(t)$ and B . This means that to solve this control problem, linear controllers can only be used if the system is linearized around a chosen operating point; or by non-linear controllers applied to the bilinear problem. The choice of the controller is discussed in *Chapter 7 Controller Design*.

6.4 Model Discretization

Now that the continuous system state space system has been setup, it needs to be discretized. This means that the system now needs to take the form:

$$\begin{aligned} \mathbf{x}(k+1) &= A\mathbf{x}(k) + B\mathbf{u}(k) \\ \mathbf{y}(k) &= C\mathbf{x}(k) + D\mathbf{u}(k) \end{aligned} \quad (6.24)$$

This can be done with a variety of methods, such as the Euler method or the finite element method. Since, the system model is a bilinear ordinary differential equation with a time-like dependence, the Euler method suffices. There are two variations of the Euler method, the forward and the backward, shown respectively:

$$y_{n+1} = y_n + hf(y_n, t_n), \text{ or } y_n = y_{n-1} + hf(y_n, t_n), \quad (6.25)$$

In this case the forward Euler method is chosen as the discretization method. Note that, the forward Euler method is a truncated Taylor series expansion of the first order, which introduces an error at every step. This creates the local truncation error. However, as it was shown before, the model is now in a first order form, therefore the choice of discretization method holds. Then by taking the system of equations from 6.18, the Euler method can be applied to each of the derivatives which gives:

$$\begin{aligned} z_1(k+1) &= z_2(k)h + z_1(k) \\ z_2(k+1) &= -\alpha z_1(k)u(k)h + z_2(k) \end{aligned} \quad (6.26)$$

Rearranging the second equation to:

$$\begin{aligned} z_1(k+1) &= z_2(k)h + z_1(k) \\ z_2(k+1) &= z_2(k) - \alpha h z_1(k)u(k), \end{aligned} \quad (6.27)$$

then can be put into a state space form by introducing an identity matrix and the vector \mathbf{h} it is possible to write up:

$$\begin{bmatrix} z_1(k+1) \\ z_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix} \begin{bmatrix} h \\ h \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\alpha & 0 \end{bmatrix} \begin{bmatrix} h \\ h \end{bmatrix} \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}, \quad (6.28)$$

By factoring out \mathbf{z} on the right hand side of the equations, the above can be rewritten to:

$$\begin{bmatrix} z_1(k+1) \\ z_2(k+1) \end{bmatrix} = \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} h \\ h \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\alpha & 0 \end{bmatrix} \begin{bmatrix} h \\ h \end{bmatrix} \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}, \quad (6.29)$$

The systems of equations can be put into a compact state space form:

$$\mathbf{z}(k+1) = (\mathbf{A}\mathbf{h} + \mathbf{I}) + \mathbf{B}\mathbf{h}\mathbf{z}(k)\mathbf{u}(\mathbf{k}) \quad (6.30)$$

Then, by redefining \mathbf{A} by incorporating \mathbf{h} and \mathbf{I} into the matrix yields:

$$\mathbf{A} = (\mathbf{A}\mathbf{h} + \mathbf{I}) \quad (6.31)$$

Then by plugging the redefined \mathbf{A} into 6.30, and by taking the compact form of the matrices the final discrete state space form is achieved for the dipole EOM:

$$\mathbf{z}(k+1) = \mathbf{A}\mathbf{z}(k) + \mathbf{B}\mathbf{h}\mathbf{z}(k)\mathbf{u}(\mathbf{k}) \quad (6.32)$$

6.4.1 Issues with discretizing in space and time concurrently

So far two things has been assumed about the system for the discretization to work mathematically:

1. The actuation and measurement devices are equally placed from each other. Meaning that $s_2 = s_1 + s_1$.
2. Furthermore, s is a time-like property that also signifies distance, therefore the continuous state space model works, where t is substituted in with s , and the derivative is solved with respect to s .

However, by revisiting and introducing an updated version of *Figure 6.2*, it is possible to highlight the issue of the location, where prediction in discrete form $k + 1$ is situated, see *Figure 6.3*.

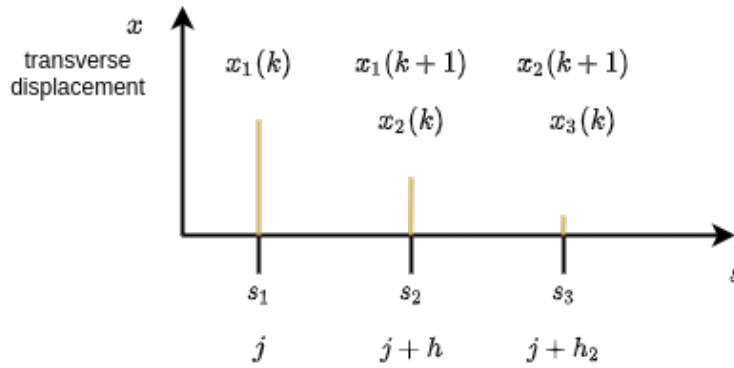


Figure 6.3: Discrete transverse coordinates of the transverse displacement at s_1 , s_2 and s_3 . At each location a magnet and BPM is assumed.

The image shows that the $k + 1$ estimate of the state space model is actually equal to the measured state of the next dipole element. Recall that the physical definition of electron beam line says that the measurement is at the exact place of actuation. In practice this would mean, that to describe the system, z_3 would need to be introduced as such:

$$\begin{aligned} z_1(k+1) &= z_2(k)h + z_1(k) \\ z_2(k+1) &= z_2(k) - \alpha h z_1(k)u(k), \\ z_3(k+1) &= z_2(k+2) - \alpha h z_2(k)u(k) \end{aligned} \tag{6.33}$$

This formalism does not follow the state space formalism described in Equation (6.24), as this system would not be causal. Furthermore, by inspecting the the continuous form of this equation one can notice that $z_3(k)$ and $z_3(k+1)$ would require the introduction of 3rd order derivatives of x and in state space form, eventually a 4th order derivative too:

$$\begin{aligned} z_1(s) &= x(s) \\ z_2(s) &= x'(s) \\ z_3(s) &= x''(s) \end{aligned} \tag{6.34}$$

Eventually this would mean numerical instability as derivatives increase due to the truncating error of the forward Euler method. Due to these issues outlined, another controller approach is presented in the following section.

6.4.2 Proposed new controller design

The proposed new controller design with assume that there is a controller at each magnet, see *Figure 6.4*. This would mean that subsequent predicted states would actually be at the same magnet location, effectively acting on the new particle that is going to arrive by the next iteration. However, this approach would require a set of assumptions:

1. It is assumed that k is simply a time-like property without longitudinal displacement.
2. The subsequent particles in the beam would behave the same way for the magnet actuation as the time ones at k .

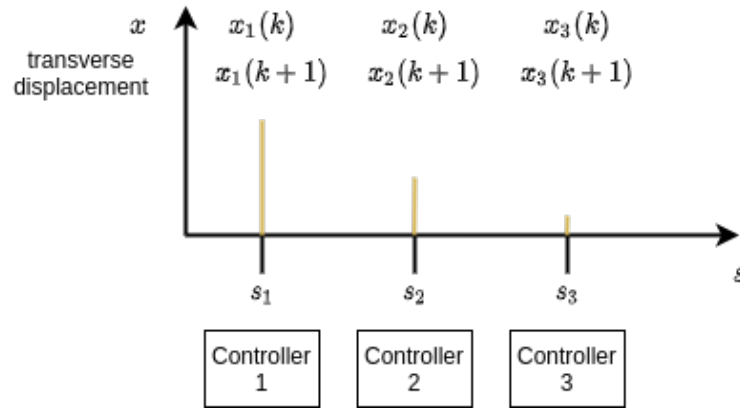


Figure 6.4: The new proposed controller model would assume that there is a controller at each of the magnets. This is possible since the actuation would not effect the current particle, rather it would affect the subsequent particle coming through the vacuum tube..

Furthermore, this approach would allow for the following adjustments for optimization:

1. Different feedback gains, K , at each magnet. Since the magnets downstream of the beamline would never see very high fluctuations and would rather be needed to make very small corrections to the beam to maximize the steady state convergence to the reference.
2. Individual models for the dipole error introduction. Currently, the magnets are assumed to be uniformly the same. By using a different controller, this would allow to introduce local adjustments against dipole bending errors.

The controller implementation is discussed in *Chapter 7 Controller Design*.

6.5 State Measurement via BPMs

Beam Position Monitors (BPMs) are designed to indirectly, and in a non-destructive way, measure the location of the beam in the beam pipe. This is done by measuring the electromagnetic fields produced by the particle passing through, see *Figure 6.5*.

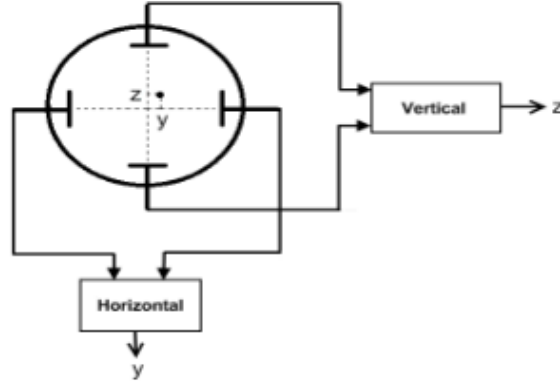


Figure 6.5: An electrode Beam Position Monitor. Electrodes are placed at the sides of the device, which can detect the magnetic fields produced by the magnets in the beam pipe. [33]

The BPMs measurement for the transverse axes, x and y can be calculated in the following way by using the delta-sigma, $\Delta\Sigma$, method:

$$\begin{aligned} X &= \frac{1}{S_x} \times \frac{V_{up} - V_{down}}{V_{up} + V_{down}} \equiv \frac{1}{S_x} \times \frac{\Delta V_2}{\sum V_2} \\ Y &= \frac{1}{S_y} \times \frac{V_{right} - V_{left}}{V_{right} + V_{left}} \equiv \frac{1}{S_y} \times \frac{\Delta V_y}{\sum V_y} \end{aligned} \quad (6.35)$$

where, V is the potential, $S_{x,y}$ are proportional constants called position sensitivity. This project work is not further concerned with BPMs, since the provided simulation systems can produce a numerical scalar and does not require an estimator. Further reading can be found at [33].

Chapter 7

Controller Design

In this chapter the control system design is presented. It begins with a short discussion on the possibilities of optimal bilinear and non-linear control, which sets the basis for the choice of controller. Afterwards, the chosen controller is described. Then, the controller model is constructed.

7.1 Control of Bilinear Systems

Bilinear systems are a special class of nonlinear systems. The general definition of a bilinear system takes the form:

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + \sum (B\mathbf{x}(k) + B)\mathbf{x}(k). \quad (7.1)$$

As stated in, *Subsection 6.3.2 System Linearity*, the system becomes bilinear because of the interactions between B and $\mathbf{x}(k)$. There are several methods currently present to control discrete bilinear systems. One of such is based on the Sum of Squares (SOS) and on the quadratic Lyapunov function [34]. In this paper, the authors propose a direct optimal controller for bilinear systems. The objective is to minimize the cost function:

$$J(x, u) = \mathbf{x}(k+1)^T Q \mathbf{x}(k+1) + \mathbf{u}(k)^T R \mathbf{u}(k) \quad (7.2)$$

where, the Q and R matrices are weight on the state error and on the actuator input, respectively. These matrices are further discussed in the chapter, below. The optimal full state feedback law then becomes:

$$\mathbf{u}(k) = - \left[(B(x) + B)^T Q (B_x + B) + R \right]^{-1} (B_x + B)^T Q A \mathbf{x}(k) \quad (7.3)$$

This controller is a promising approach, however, as the authors state in the paper, it does not guarantee global stability. For that, a polynomial ratio of the controller is constructed to maximize the region of convergence. [34]

7.1.1 Non-linear Model Predictive Control

Other attempts were made to apply a non-linear Model Predictive Controller (non-linear MPC) to the bilinear problems by guiding the non-linear parts of the bilinear system towards the linear parts. [35] An MPC approach would also be applicable due to its possibilities with input and output constraints. In the bilinear system of dipole magnets, the input constraints could be characterized by the maximum and minimum current, whereas the output constraints could create a maximum and minimum for the allowed transverse displacement.

Non-linear MPCs have been widely used before, however, in this specific case due the nature of the fast dynamics over this electron beam line, for the trajectory problem the prediction finite horizon approach would not theoretically work.

7.1.2 Choice of controller

Even though solutions have been provided for the bilinear system, in this project, at first an Linear Quadratic Regulator is attempted, this is based on the the assumptions outlined *Chapter 3 Modeling of Particle Motion*, where the $x(s)$ effect on the system is negligible compared to the longitudinal components. As in, transverse displacement is measured in micro-meters against longitudinal meters. Therefore, it can be assumed that the system acts mostly linearly and a linear controller could suffice. This was briefly discussed in *Section 4.5 Iterative LQR (iLQR)*. This choice is also motivated by the approach of having a different controller at each magnet.

Thus, to reduce complexity and to speed up computation a LQR is chosen as the controller. In the following sections the LQR implementation is discussed.

7.2 Linear Quadratic Regulator (LQR)

LQR is a type of optimal control. It is concerned with finding an optimal state feedback law through a quadratic minimization problem. Mainly, LQR works on linear differential equations of motion in the form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (7.4)$$

where, the control input \mathbf{u} is defined as a multiplication between the feedback F and the current state \mathbf{x} :

$$\mathbf{u}(k) = -F\mathbf{x}(k). \quad (7.5)$$

By inputting 7.5 into 7.4 yields the whole dynamic system with full state feedback:

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}F)\mathbf{x} + \mathbf{B}F\mathbf{x}_{ref} \quad (7.6)$$

A quadratic cost function is then setup which introduces two matrices, Q and R :

$$L = \frac{1}{2} \mathbf{x}^T Q \mathbf{x} + \frac{1}{2} \mathbf{u}^T R \mathbf{u}. \quad (7.7)$$

By solving this cost function an optimal solution to the control problem can be found. Additionally, the Q matrix allows manual input over the weight of the state errors, hence the multiplication with \mathbf{x} , whereas the R matrix allows for manual input over the weights on the actuator cost, hence the \mathbf{u} . By tuning these matrices it is possible to find satisfactory performance of the controller. However, there is no set mathematical model which can automatically set the matrices, generally it is done by the current engineer.

The optimal solution can now be found by solving a discrete Algebraic Riccati equation (DARE):

$$PAx + A^T Px + Qx - PBR^{-1}B^T Px + \dot{P} = 0 \quad (7.8)$$

which can be rewritten as:

$$PA + A^T P + Q - PBR^{-1}B^T P = 0 \quad (7.9)$$

then, this can be used to get the optimal infinite-horizon discrete LQR gain F :

$$F = (R + B^T PB)^{-1}(B^T PA + N^T). \quad (7.10)$$

The above equation then gives the full state feedback solution for \mathbf{u} .

The LQR controller implementation can be summarized by:

1. Setup state space model and isolate A and B matrices
2. Define Q and R matrices
3. Solve for F by computing a solution to the DARE
4. Implement full state feedback for $\mathbf{u}(k)$

7.3 Controller Model

As mentioned in *Section 6.4 Model Discretization*, the proposed controller model needs to assume that the next state of the system appears at the exact same place, i.e: the longitudinal displacement s is heavily assumed to be time-like. This allows for the implementation of a controller onto each magnet. A set of further assumptions can be defined to better specify the control problem:

1. Since the predicted states can be measured, a controller at each magnet is appropriate.
2. Since there are no change in dynamics, as at this point only dipoles are actuated and in the drift space the particles do not change momentum and stay at the same velocity.
3. The horizontal and vertical states are decoupled.

4. The bilinear $\mathbf{x}(k)$ has very little effect on the B matrix.

This way, the controller controls the particle displacement for the next iteration at each magnet. This is shown in *Figure 7.1*.

The proposed feedback loop can be summarized in a list:

1. The trajectory reference is set as $r = 0$. This is always zero since the controller goal is to reduce deviation away from the center of the beamline as soon as possible. The steady state error is desired to be zero.
2. The trajectory error is computed.
3. The optimal full state feedback law is applied.
4. The dipole corrector current is set.
5. The trajectory output is observed by a BPM directly after the actuation.

A single magnet controller model can now be introduced, see *Figure 7.2*.

Since it is assumed that the magnets are uniformly identical, the computational algorithm of the controller would resolve to the following steps:

1. Compute full state feedback law $-F$.
2. Iterate magnets from 0 to n number of magnets.
 - Apply feedback law at each iteration.
 - Measure response at each iteration.
3. Repeat until beam in beam pipe. The algorithm must not quit if the steady state reference is reached.

7.3.1 Initial conditions

The initial conditions for the system are the displaced particles arriving from the RF gun. Since it is assumed that the states are decoupled only, the $x(0)$ and $x''(0)$ has to be defined. The implementation of the initial conditions are further discussed in *Chapter 8 Electron Beam Line Simulation and Controller Testing*.

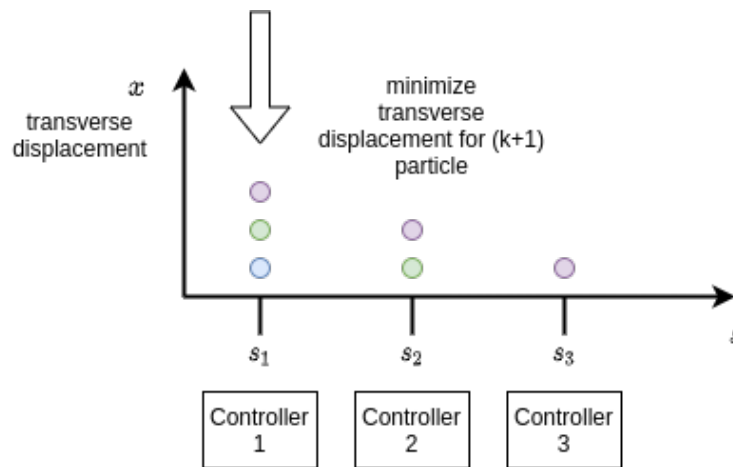
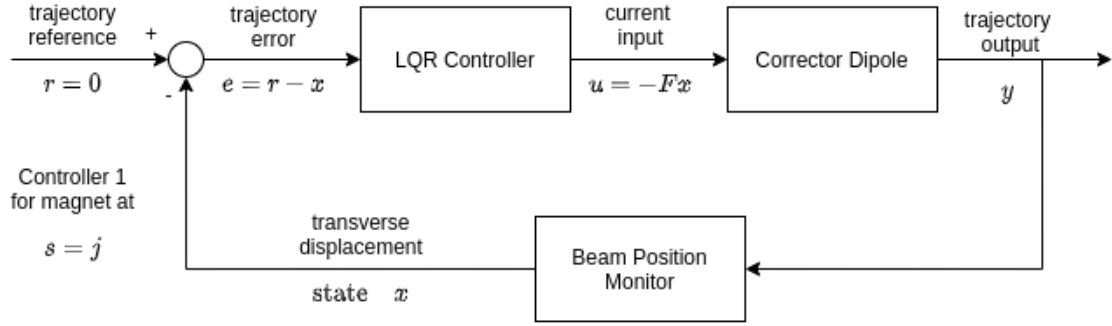


Figure 7.1: Controller Model for controller at magnet placed $s = j$.

Figure 7.2: Controller Model for controller at magnet placed $s = j$.

7.3.2 Constants and final state space form

From *Section 6.4 Model Discretization* the constants can now be set for the controller. *Table (7.1)* shows the values for each of the constants to be put into the model. Notice, that the dipole magnet winding number is set to 1. As discussed in *Subsection 3.2.1 Dipole magnets*. Furthermore, the Euler approximation time between the two states h is set to 1. This is based on the assumptions outlined about the behaviour of k .

The constants then be substituted into α :

$$\alpha = \left[\frac{q}{p} \mu_0 n \frac{1}{h} \right]^2 \approx 2.230025e6 \quad (7.11)$$

and with that the final state space form takes:

$$\begin{bmatrix} z_1(k+1) \\ z_2(k+1) \end{bmatrix} = \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -2.230025e6 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}, \quad (7.12)$$

Constant	Value	Unit
Charge q	$1.602\,176\,62 \times 10^{-19}$	[C]
Vacuum permeability μ_0	$1.256\,637\,062\,12 \times 10^{-6}$	[H/m]
Speed of light c	2.99×10^8	[m/s]
Electron mass m	$9.109\,383\,56 \times 10^{-31}$	[kg]
Dipole magnet winding number n	1	[-]
Dipole magnet distance between poles	0.05	[m]
Particle momentum p	$m \cdot 0.99 \cdot c$	[kg m/s]
Distance between magnets h	1	[m]

Table 7.1: Constants for the model of the system.

Chapter 8

Electron Beam Line Simulation and Controller Testing

In this chapter the beam line simulation is presented. This simulation is written in python and it heavily depends on the deep-RL simulation introduced in *Section 4.4 Deep Reinforcement Learning (deep-RL)*. It interfaces to various CERN environments and CERN specific software. These interfaces and dependencies are explained. Then, the structure of the base simulation is shown. This was changed to allow for an implementation of both the feed forward SVD, LQR and additionally a general P-controller. The specific implementation is then presented. The controller testing is also contained in in this chapter.

8.1 Methodical Accelerator Design (MAD)

Methodical Accelerator Design (MAD) is a CERN built scripting language for accelerator design. The current version MAD-X, is the successor of MAD-8 which was released in 2002. The new version includes specific upgrades exclusively to help the design of the LHC. It is still currently maintained and it is the main tool for charged-particle optics design of medium to very large accelerators. [36]

The AWAKE electron was also encoded in this language. A snippet is shown in *Figure 8.4*.

The file is broken up into columns. Each row in the column represents a different element

1	* NAME	KEYWORD	S	L
2	\$ %s	%s	%le	%le
3	"TT43\$START"	"MARKER"	0.000000000	0.000000000
4	"BEG1.1000"	"MARKER"	0.000000000	0.000000000
5	"DRIFT_0"	"DRIFT"	0.179170000	0.179170000
6	"BPM.430028"	"MONITOR"	0.179170000	0.000000000
7	"DRIFT_1"	"DRIFT"	0.248570000	0.069400000
8	"MCAWA.430029"	"KICKER"	0.289170000	0.040600000

Figure 8.1: MAD-X file scripting language example, with a DRIFT, KICKER and MONITOR elements and their longitudinal displacement s and physical length L . KICKER and MONITOR stands for the corrector dipole and the corresponding BPM.

An extended version of this example is utilized in the python simulation.

in the particle accelerator. This above example models the AWAKE electron beam line outlined *Subsection 2.3.1 Electron RF gun and beam line*. Each column represents a different property of the elements. These properties are called Courant-Snyder properties of the elements such as magnets. They are often called Twiss parameters.

8.1.1 Twiss parameters

For this simulation environment, the Twiss parameters are used to compute the Orbit Response Matrix. Recall the equation from *Section 4.3 Singular Value Decomposition (SVD) Based Approach*:

$$\mathbf{A}_{i,k} = \frac{\sqrt{\beta_m \beta_k}}{2 \sin(\mu/2)} \cos(|\phi_m - \phi_k| - \mu/2). \quad (8.1)$$

The python implementation is shown in *Figure 8.2*. In this example the authors called the ORM R , in other literature it is also customary to use that instead of A . Furthermore, some of the variables are also with different symbols. This is also often the case with newer implementations of older concepts, such as the ORM.

```

1 def calc_response_matrix(self, bpms: t.Sequence[TwissElement],
2                           correctors: t.Sequence[TwissElement],
3                           dtype: t.Optional[type] = None) -> np.ndarray:
4
5     rmatrix = np.zeros((len(bpms), len(correctors)), dtype=dtype)
6     for i, bpm in enumerate(bpms):
7         for j, corrector in enumerate(correctors):
8             beta_prod = bpm.beta * corrector.beta
9             mu_diff = bpm.mu - corrector.mu
10            rmatrix[i, j] = (
11                0.0
12                if mu_diff < 0.0
13                else (np.sqrt(beta_prod) * np.sin(2 * np.pi * mu_diff))
14            )
15     return rmatrix

```

Figure 8.2: Function showing the implementation of the ORM calculation in the python simulation.

8.1.2 Twiss Reader

A Twiss reader class was originally designed to parse the MAD-X file into python. This class provides functionality so that the ORM can be computed. Some of the main functions provide a way to access any user desired MAD-X files, furthermore, in this class also encodes a matrix based trajectory computation. This is the same formula outlined in *Subsection 3.3.2 Matrix formalism*.

8.2 Base deep-RL Simulation Environment

The base that was used to develop the LQR simulation was based on the deep-RL python simulation. This simulation was designed to train the RL agent to find the optimal control

policy to correct the trajectory. The full algorithm was described in *Section 4.4 Deep Reinforcement Learning (deep-RL)*. It contains the Twiss reader module, which provides access to the MAD-X file by parsing the requested Twiss Elements into the `TwissElement` class.

The simulation was written in python and it is using the OpenAI.gym trainer to train the agent and execute a test period on the parsed MAD-X simulation of the electron beam line.

8.3 Simulation Modifications for LQR Implementation

The base simulation was strictly built around the RL agent. It had the form of an OpenAI project, meaning that the algorithm implementation which iterates over magnets was not present. This had to be added to the simulation so that it was possible to interface with the. The following changes have been made:

- Introduction of a new class to handle different controllers
- Extrapolation of the ORM computation functions from the OpenAI simulation into this class
- Introduction of a controller loop which iterates over the magnets.
- Implementation of functions which can input computed corrector currents into the magnets
- The measurement is also computed using the ORM.

Controllers which are based on an iterative loop can now be implemented parallel to the deep-RL algorithm, see *Figure 8.3*.

8.4 Controller Implementation

The LQR controller is implemented into the simulation using Scipy. It is an extensive set of mathematical computation algorithms in a python library form. [37] Among others, scipy provides functions to safely compute and invert matrices. This is of course required to compute the DARE. In the simulation, a function was created to compute the feedback gain F , the DARE X and the eigenvalues of the system. The function implementation can be seen in *Figure 8.4*.

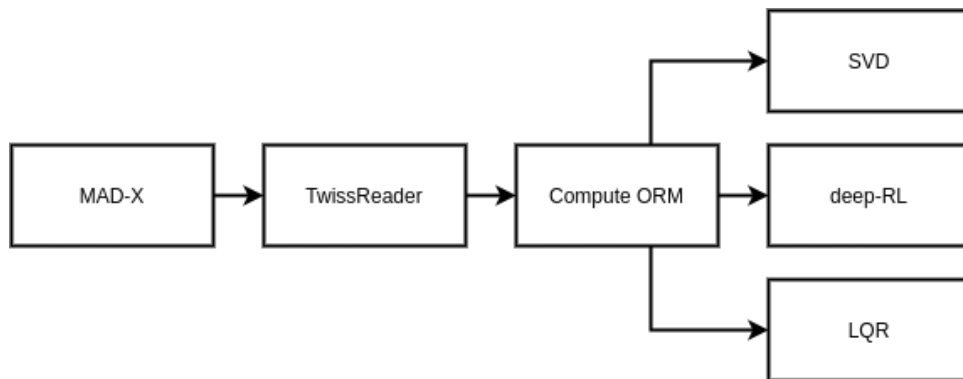


Figure 8.3: Simulation design expanding to use SVD and LQR parallel to deep-RL

```

1 def dlqr(self, A, B, Q, R):
2
3     # first, try to solve the algebraic Ricatti equation
4     X = np.matrix(scipy.linalg.solve_discrete_are(A, B, Q, R))
5
6     # compute the LQR gain
7     K = np.matrix(scipy.linalg.inv(B.T * X * B + R) * (B.T * X * A))
8
9     # also compute the eigen values and vectors to inspect stability
10    eigen_values, eigen_vectors = scipy.linalg.eig(A - B * K)
11
12    return F, X, eigen_values

```

Figure 8.4: Python function to compute the optimal feedback gain, F

The Q and R matrices are also implemented using Scipy and will be manually tuned for best performance. The control algorithm for each of the magnets can now be also implemented. It is done using a for loop, which iterates until 10 steps. The BPM measurements are returned as a vector containing $x(k)$ for every magnet. Then, the error can be computed between the reference of $x = 0$ and the current state $x(k)$. The feedback gain is then multiplied by the error. This implementation can be seen in *Figure 8.5*.

Effectively, this algorithm uses the feedback law to compute the control input for each of the magnet within an iteration.

8.4.1 Initial Conditions

Since the beam is represented as an almost continuous line of particles inside the beam pipe the initial conditions require special implementation. In this case, the initial conditions are set as a semi-random corrector currents into the beam before the algorithm is applied to it. This means, that before the control algorithm is applied, the beam is "disturbed" into a random position. This random position is defined by the `np.random.uniform` function from Numpy, see *Figure 8.6*.

```

1 F, X, eigVals = awake.dlqr(A, B, Q, R)
2 simul_lenght = 10
3 for iteration in range(0, simul_lenght):
4     bpms_measurements = awake.compute_observation(corrector_current=corrector_currents)
5     # z2 = z1'
6     # z2 = (z1_prev - z1_curr) / dt
7     # z[0] = z1_curr
8     # z[1] = z2
9
10    error = reference - bpms_measurements
11    corrector_currents = -F.item((1, 1)) * error

```

Figure 8.5: Control loop implementation

```

1 def get_initial_conditions(self):
2     self.settings = np.asarray(
3         np.random.uniform(-1, 1, self.settings.shape),
4         np.float32,
5     )
6     return self.settings
7
8 # Main
9 # Setup the awake simulation class
10 plane = "H"
11 remove_singular_devices = True
12 awake = AwakeSimul(plane, remove_singular_devices)
13
14 # initial conditions which come from the "reset" function
15 corrector_currents = awake.get_initial_conditions()

```

Figure 8.6: Inputting the initial conditions

8.5 Controller Implementation Testing

To test whether the implementation worked, the simulation is ran was run the initial Q and R , matrices are set to be two identity matrices. The initial conditions were also adjusted so that the values do not fall out of the general $2\mu\text{m}$ range.

At this stage it can be concluded that the LQR controller feedback loop has corrected the beam within less than $2\mu\text{m}$ in four iterations. The controller can further tuned by changing the Q and R matrices to set a cost to the state error.

8.5.1 Controller Tuning

By iteratively tuning the Q and R matrices it can be concluded that the effects are barely noticeable. A better tuned controller is shown in *Figure 8.10*.

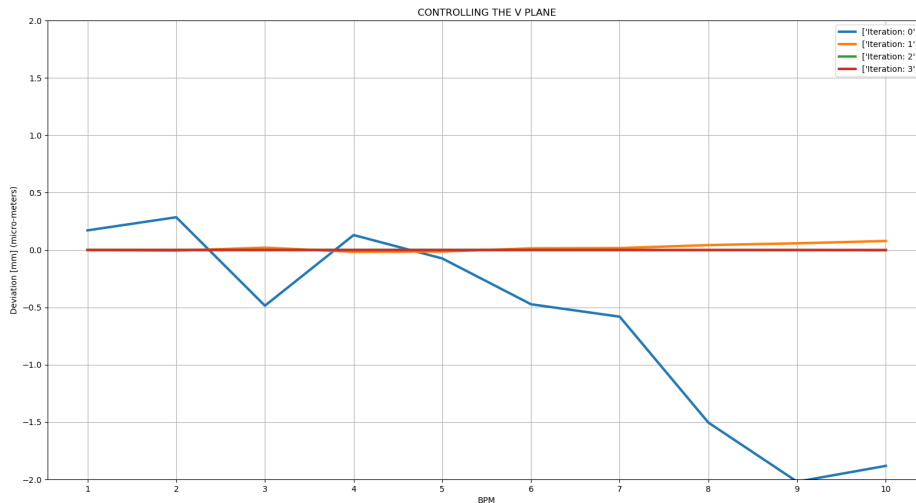
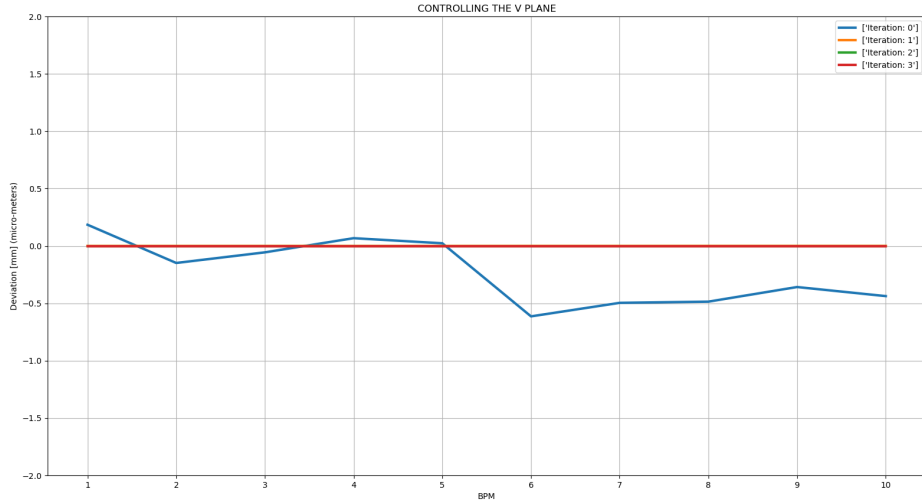


Figure 8.7: LQR result in the vertical plane

Figure 8.8: A better tuned Q and R matrix

The new values for the Q and R are as follows now:

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.001 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix}, \quad (8.2)$$

Only the bottom right values are changed since those are the ones affecting the input u into the system, since the first state is not controllable directly. This is discussed in the stability analysis in the next section.

8.5.2 Stability Analysis

The stability of the LQR can be measured by checking the poles of the system. In *Section 8.4 Controller Implementation* it is shown that with Scipy the poles of the system can also be acquired (in that example the values are called eigenvalues, however, the numbers are a complex number when read). The resulting values show the following, see *Figure 8.9*.

```
1 poles: [ 0.00000000e+00+0.j -5.34017424e-24+0.j]
```

Figure 8.9: Poles of the system

The first pole of the system is zero. This is understandable because the first state of the system at each of the magnets is uncontrollable. For the second state the pole is zero, meaning that it falls onto the left half plane and it is stable. Although, note that the number is extremely small, it is multiplied by 10^{-24} . The likely reason for this is very fast response time.

8.6 Controller Comparison

The tuned controller can now be compared against the SVD and P-controller implementation in the same simulation, as well as recorded results for the deep-RL from the paper [30].

8.6.1 SVD

The SVD algorithm was implemented into the simulation following the state of the art analysis outlined in *Section 4.3 Singular Value Decomposition (SVD) Based Approach*. The initial SVD results show the following; the SVD computation manages to stabilize the beam in four iterations compared to the LQR, where with a tuned Q and R matrices, it can do it in two to three iterations.

It is possible to conclude that SVD is a slower algorithm than the LQR, however, the performance is worse but the implementation cost is much cheaper.

8.6.2 deep-RL

The deep-RL results come directly from the paper. This is because once the simulation was modified it was detached from CERN's internal networks and since strong dependencies exist into internal packaging system, it does not work outside of the CERN network. The results were already shown in *Section 4.4 Deep Reinforcement Learning (deep-RL)*, but are pasted here again, see *Figure 8.11*.

It can be concluded that the results are similar to the LQR implementation. It took approximately three iterations of the algorithm to converge, which is similar to the tuned LQR.

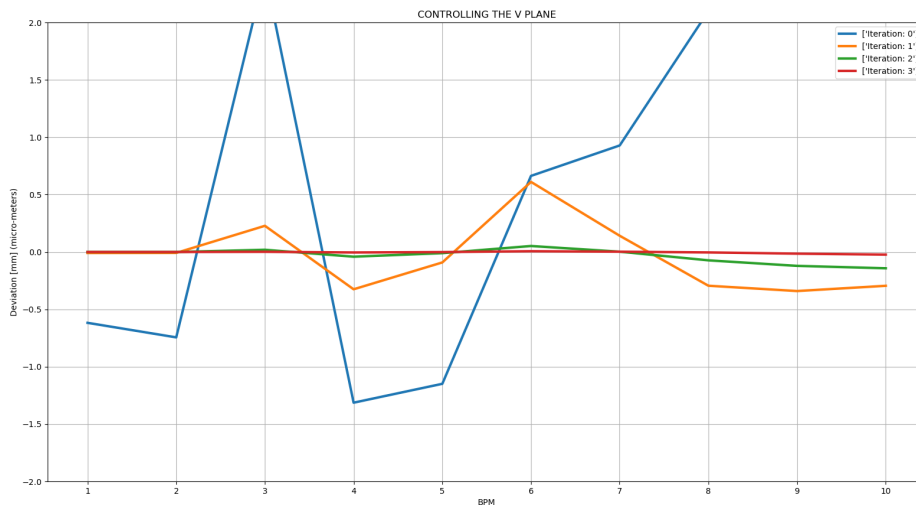


Figure 8.10: SVD implementation stabilizes the beam to zero steady state error in four iterations.

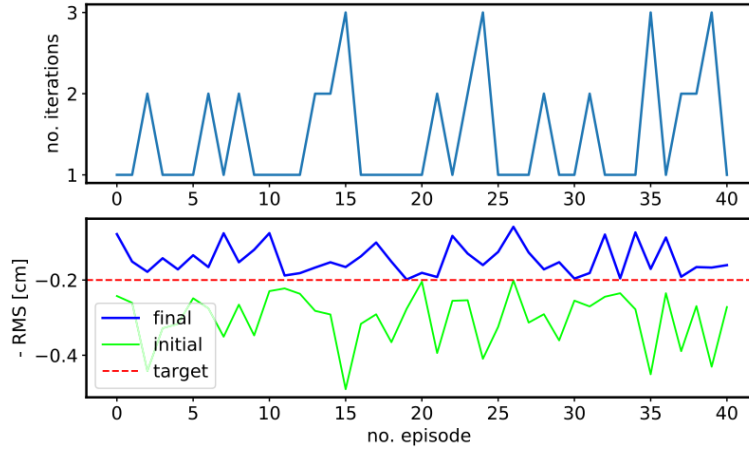


Figure 8.11: Results of a 35 iteration trained agent to correct the RMS of the beam to less than $2\text{ }\mu\text{m}$ [30]

8.7 Conclusion on Controller Testing

Based on the simulations and the testing, it is possible to conclude that the LQR controller managed to correct the beam trajectory towards the steady state at a similar performance as the deep-RL algorithm. It outperformed the SVD algorithm by a slight margin too. Thus, it is possible to also conclude that model-based algorithms are capable of correcting beam trajectories in an environment where a sufficient model is difficult obtain.

Chapter 9

Global Conclusion

This project dealt with model based optimal control of a CERN based experimental particle accelerator electron transfer line. The work heavily revolved around state of the art analysis, particle modelling and controller design.

At the beginning, the project outlines the fundamentals of CERN. Then the physical structure of the AWAKE is described in detail with a special focus on the electron beam line. Particle motion modeling then followed, which resulted in a set of linear second order differential equations. The work also included modelling of magnetic fields which resulting in the modelling of the magnet contributions to the particle motion.

State of the art analysis then revealed that, at the time of writing this report, there were no clear model based controller attempts to solve the trajectory correction problem. The analysis visited simple feedforward models to complex deep neural network approximated non-linear system identification approaches utilized in the context of reinforcement learning. Subsequently, based on this analysis, a set of requirements were defined along with the specific control goal to guide the rest of the work.

A solution proposal is then presented and put into state space form, then discretized. A controller proposal is then created using an LQR controller on the bilinear system. The controller design then implemented a controller which was designed to be running on each magnet. This allowed for the expansion of the simple model onto all corrector magnets of the electron beam line.

The proposed controller algorithm was then simulated in a python simulation based on the deep-RL environment. This simulation uses a parsed version of CERN's scripting languages which encodes the physical characteristics of the electron beam line. Using this simulation, the LQR controller was implemented and the full state feedback was used to compute magnet currents.

The LQR controller was successfully implemented in the simulation and results show that the algorithm is capable at to correct the beam trajectory within three iterations as the other state of the art controller. Finally, it can be concluded the model-based approaches for beam trajectory correction are in-fact a viable approach and should be further investigated in the future. A short section now follows with some ideas about future work.

9.1 Future Work

There is strong potential for future work on the model based approach. In this project, beam focusing was not considered, only trajectory correction. Since the EOM are already defined for the quadrupole action, it would be possible to augment the controller to compute quadrupole magnet gains too.

Another possibility would be to simultaneously include the other axes into the computation, as in, include the coupled y axes.

Furthermore, emittance based model based control could also be implemented, where the state of the art approaches already include a time based approach instead of the time-like longitudinal displacement dependence.

In the end, this thesis project proved that model based optimal control is a viable option for control of beam trajectories in particle accelerators.

Bibliography

- [1] History of cern. <https://home.cern/about/who-we-are/our-history>. Accessed: 2021-01-05.
- [2] Mapcern. <https://maps.web.cern.ch/?xmin=2481319.56&ymin=1117786.43&xmax=2509428.95&ymin=1132399.92&basemap=plan&mode=2D>. Accessed: 2020-11-18.
- [3] Experiments. <https://home.cern/science/experiments>. Accessed: 2021-01-05.
- [4] The large hadron collider. <https://home.cern/science/accelerators/large-hadron-collider>. Accessed: 2021-01-05.
- [5] Rf cavities. https://www.lhc-closer.es/taking_a_closer_look_at_lhc/0.rf_cavities. Accessed: 2021-08-24.
- [6] Lhc in the tunnels. <https://home.cern/resources/faqs/facts-and-figures-about-lhc>. Accessed: 2021-08-27.
- [7] Linac4. <https://home.cern/science/accelerators/linear-accelerator-4>. Accessed: 2021-01-05.
- [8] The awake experiment. <https://home.cern/science/accelerators/awake>. Accessed: 2021-08-24.
- [9] Awake successfully accelerates electrons. <https://home.cern/news/news/experiments/awake-successfully-accelerates-electrons>. Accessed: 2021-08-27.
- [10] Slac starts up new facility to revolutionize particle accelerators. <https://www6.slac.stanford.edu/news/2020-10-12-slac-starts-new-facility-revolutionize-particle-accelerators.aspx>. Accessed: 2021-07-14.
- [11] The berkeley lab laser accelerator (bella) center. <https://bella.lbl.gov/>. Accessed: 2021-07-14.
- [12] Plasma wakefield acceleration. https://portal.slac.stanford.edu/sites/ard_public/facet/Pages/rpwa.aspx. Accessed: 2021-08-27.
- [13] 2020 roadmap on plasma accelerators. <https://iopscience.iop.org/article/10.1088/1367-2630/abcc62#njpabcc62s8>. Accessed: 2021-07-14.

- [14] Ls2 report: A new schedule. <https://home.cern/news/news/accelerators/l2-report-new-schedule>. Accessed: 2021-08-24.
- [15] T. Tajima and J. Dawson. Laser electron accelerator. *Physical Review Letters*, 43:267–270, 1979.
- [16] T. Katsouleas; J. M. Dawson; J. M. Kindel C. Joshi, W. B. Mori and D. W. Forslund. Ultrahigh gradient particle acceleration by intense laser-driven plasma density waves. *Nature*, 311:525–529, 1984.
- [17] Introduction to plasma wakefield acceleration. https://indico.cern.ch/event/932976/contributions/3920626/attachments/2135461/3596993/Wakefield_intro.pdf. Accessed: 2021-08-27.
- [18] E. Gschwendtner, E. Adli, and L. Amorim et al. Awake, the advanced proton driven plasma wakefield acceleration experiment at cern. *Nuclear Instruments and Methods in Physics Research A*, 9, 2016.
- [19] M. Okamura, J. Fite, V. Lodestro, D. Raparia, and J. Ritte. Steering magnet design for a limited space. *6th International Particle Accelerator Conference*, 6, 2015.
- [20] The awake electron beam. https://indico.cern.ch/event/985251/contributions/4149139/attachments/2163990/3651823/AwakeSummary_ABPLAT_16Dec20.pdf. Accessed: 2021-08-27.
- [21] F. Zimmermann M.G. Minty. *Measurement and Control of Charged Particle Beams*. Springer, Berlin, 2003.
- [22] Helmut Wiedemann. *Particle Accelerator Physics*. Springer, Stanford USA, 2015.
- [23] Wolfgang Hillert. Transverse linear beam dynamics, 2021.
- [24] Lorentz transformation. <http://hyperphysics.phy-astr.gsu.edu/hbase/Relativ/ltrans.html>. Accessed: 2021-08-09.
- [25] Basic theory of magnets. https://www.bnl.gov/magnets/staff/gupta/scmag-course/Talk1_manuscript.pdf. Accessed: 2021-08-18.
- [26] J. S. Schmidt, J. Bauche, and B. Biskup et al. The awake electron primary beam line. *Proceedings of PAC09*, 9, 2009.
- [27] Paraxial optics. https://spie.org/publications/fg01_p19_paraxial_optics?SSO=1. Accessed: 2021-08-09.
- [28] Alexander Scheinker, Simon Hirlander, Francesco Maria Velotti, Spencer Gessner, Giovanni Zevi Della Porta, Verena Kain, Brennan Goddard, and Rebecca Ramjiawan. Online multi-objective particle accelerator optimization of the awake electron beam line for simultaneous emittance and orbit control. *AIP Advances*, 10(5):055320, 2020.
- [29] Aimin Xiao Georg H. Hoffstaetter, Joachim Keil. Orbit-response matrix analysis at hera. *Proceedings of EPAC*, 8:407–409, 2002.

- [30] Verena Kain, Simon Hirlander, Brennan Goddard, Francesco Maria Velotti, Giovanni Zevi Della Porta, Niky Bruchon, and Gianluca Valentino. Sample-efficient reinforcement learning for cern accelerator control. *Phys. Rev. Accel. Beams*, 23:124801, Dec 2020.
- [31] Openai. <https://gym.openai.com/>. Accessed: 2021-08-29.
- [32] Iterative linear quadratic regulator design for nonlinear biological movement systems. <https://homes.cs.washington.edu/~todorov/papers/LiICINC004.pdf>. Accessed: 2021-08-31.
- [33] Beam position monitoring system in accelerators. <https://www.eit.lth.se/sprapport.php?uid=563>. Accessed: 2021-08-29.
- [34] Mohsen Vatani Morten Hovd and Sorin Olaru. Control design and analysis for discrete time bilinear systems using sum of squares methods. -, -.
- [35] H.H.J. Bloemen M. Cannon and B. Kouvaritakis. Closed-loop stabilizing mpc for discrete-time bilinear systems. *European Journal of Control*, 8:304–314, 2002.
- [36] Madx. <https://mad.web.cern.ch/mad/>. Accessed: 2021-08-29.
- [37] Scipy. <https://www.scipy.org/>. Accessed: 2021-08-29.