

REGULATING NONLINEAR PHASE SPACES USING MACHINE LEARNING AND  
ITERATIVE METHODS

By

Kelly Anderson

A DISSERTATION

Submitted to  
Michigan State University  
in partial fulfillment of the requirements  
for the degree of

Physics—Doctor of Philosophy

2025

## **ABSTRACT**

The long-term stability of particles in accelerators is a fundamental problem in accelerator design and operation that involves analyzing nonlinear dynamical systems by calculating the dynamic aperture. Brute-force particle tracking is too computationally intensive for beams without strong synchrotron radiation damping. Particles, therefore, may survive for millions or more turns before being lost [1]. Frequency map analysis is a commonly used standard for analyzing accelerator phase spaces, but it is still reliant on particle tracking. This dissertation evaluates two approaches to studying nonlinear dynamics in particle accelerators.

First, an iterative method expanded from 2-D to 4-D phase spaces and used to find a diffeomorphism to a simplified phase space in which particle dynamics are rigid rotations [2]. The resulting convergence maps can serve as a practical tool for identifying phase-space structures. The second approach is a data-driven method for calculating the eigenvalues and eigenfunctions of a Koopman operator, which correspond to approximate frequencies and invariants of the system. This approach expanded upon previous approaches by using symplectic neural networks to find eigenfunctions that can improve the efficiency of the models they are trained on [3], [4]. Both of these can aid in dynamic aperture calculations.

Copyright by  
KELLY ANDERSON  
2025

## ACKNOWLEDGMENTS

## TABLE OF CONTENTS

|                                                                       |      |
|-----------------------------------------------------------------------|------|
| LIST OF TABLES . . . . .                                              | vii  |
| LIST OF FIGURES . . . . .                                             | viii |
| CHAPTER 1      INTRODUCTION . . . . .                                 | 1    |
| 1.1    Introduction to Particle Accelerators . . . . .                | 1    |
| 1.2    Linear Dynamics . . . . .                                      | 4    |
| 1.3    Hamiltonian Dynamics in Accelerators . . . . .                 | 5    |
| 1.4    Magnetic Components . . . . .                                  | 11   |
| 1.5    Matrix Approach . . . . .                                      | 14   |
| 1.6    Courant-Snyder Parameters . . . . .                            | 18   |
| 1.7    Action-Angle Variables . . . . .                               | 20   |
| 1.8    Linear Coupling . . . . .                                      | 23   |
| 1.9    Symplectic Condition . . . . .                                 | 25   |
| CHAPTER 2      SINGLE PARTICLE NONLINEAR DYNAMICS . . . . .           | 29   |
| 2.1    Intro . . . . .                                                | 29   |
| 2.2    Poincare Section . . . . .                                     | 30   |
| 2.3    Lie Operators . . . . .                                        | 32   |
| 2.4    Normal Form . . . . .                                          | 35   |
| 2.5    Kolmogorov–Arnold–Moser (KAM) Theorem . . . . .                | 38   |
| 2.6    Numerical Analysis of Fundamental Frequencies (NAFF) . . . . . | 38   |
| 2.7    Frequency Map Analysis . . . . .                               | 40   |
| CHAPTER 3      SQUARE MATRIX APPROACH . . . . .                       | 42   |
| 3.1    Square Matrix Method . . . . .                                 | 42   |
| 3.2    Iterative Method . . . . .                                     | 49   |
| 3.3    National Synchrotron Light Source II (NSLS-II) . . . . .       | 54   |
| 3.4    Crab Cavity Study . . . . .                                    | 58   |
| 3.5    Discussion . . . . .                                           | 65   |
| CHAPTER 4      MACHINE LEARNING APPROACH . . . . .                    | 66   |
| 4.1    Koopman Operator . . . . .                                     | 66   |
| 4.2    The Problem Setup . . . . .                                    | 69   |
| 4.3    Network Setup . . . . .                                        | 70   |
| 4.4    Standard Map . . . . .                                         | 76   |
| 4.5    McMillan Map . . . . .                                         | 79   |
| 4.6    Frequency Prediction . . . . .                                 | 80   |
| 4.7    Errors in Tracking . . . . .                                   | 84   |
| 4.8    Discussion . . . . .                                           | 85   |
| CHAPTER 5      CONCLUSIONS AND FUTURE WORK . . . . .                  | 90   |
| 5.1    Problem Summary . . . . .                                      | 90   |
| 5.2    Conclusions and Future Work . . . . .                          | 91   |

|                        |    |
|------------------------|----|
| BIBLIOGRAPHY . . . . . | 94 |
|------------------------|----|

## LIST OF TABLES

|           |                                                |    |
|-----------|------------------------------------------------|----|
| Table 3.1 | Parameters of Crab Cavity (CC) Model . . . . . | 61 |
|-----------|------------------------------------------------|----|

## LIST OF FIGURES

|             |                                                                                                                                                                                                                   |    |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 2.1  | A visualization of a particle at $x$ passing through a map and returning to the Poincare surface $S$ where its new location is $P(x)$ [13]. . . . .                                                               | 31 |
| Figure 2.2  | A Poincare map of a linear lattice with a single sextupole kick (Henon Map). Insert parameters . . . . .                                                                                                          | 31 |
| Figure 2.3  | 4-D Hénon Map frequency map for $\nu_x = 0.282$ and $\nu_y = 0.6135$ and 5000 turns [20]. . . . .                                                                                                                 | 41 |
| Figure 3.1  | The amplitude-dependent tune shift for the 2-D Hénon map calculated from NAFF and Equation 3.18 for 3rd and 5th order. . . . .                                                                                    | 47 |
| Figure 3.2  | The amplitude-dependent tune shift for the 2-D Hénon map calculated from NAFF and Equation 3.18 for 5th and 7th order using numerical methods. . . . .                                                            | 48 |
| Figure 3.3  | Stability map for the Hénon map using the third-order square matrix method to calculate $\Delta$ from equation 3.29. . . . .                                                                                      | 50 |
| Figure 3.4  | Frequency map for the Hénon map using NAFF. . . . .                                                                                                                                                               | 50 |
| Figure 3.5  | Stability map for the Hénon map using third order square matrix method to calculate $\Delta$ from equation 3.29. . . . .                                                                                          | 54 |
| Figure 3.6  | Convergence map for the Hénon map using 3rd order SMM eigenvectors as the action angle variables. . . . .                                                                                                         | 54 |
| Figure 3.7  | Inverse of the 7th order SMM for the 4-D Hénon Map using the parameters $\nu_x = 0.282$ and $\nu_y = 0.6135$ (a) and using Newton's method (b) [20]. . . . .                                                      | 55 |
| Figure 3.8  | Convergence maps for the 4-D Hénon Map using the parameters $\nu_x = 0.282$ and $\nu_y = 0.6135$ without SMM (a), with 3rd order SMM (b), and with 7th order SMM (c) for the action-angle variables [20]. . . . . | 55 |
| Figure 3.9  | The converge maps for the NSLS-II using a 3rd (a) and 5th (b) order SMM method to get the action angle variables. . . . .                                                                                         | 56 |
| Figure 3.10 | The converge maps for the NSLS-II using a 7th (a) and 10th (b) order SMM method to get the action angle variables. . . . .                                                                                        | 56 |
| Figure 3.11 | The tune maps for the NSLS-II using a 3rd and 5th order SMM method to get the action angle variables. Resonance lines up to the 10th order are shown for reference. . . . .                                       | 57 |



|             |                                                                                                                                                                                                                                                                                                                   |    |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 3.12 | The tune maps for the NSLS-II using a 7th and 10th order SMM method to get the action angle variables. Resonance lines up to the 10th order are shown for reference. . . . .                                                                                                                                      | 57 |
| Figure 3.13 | Frequency map analysis on the NSLS-II bare lattice for 1000 turns calculated in Elegant. . . . .                                                                                                                                                                                                                  | 58 |
| Figure 3.14 | Frequency map analysis in the tune space for the NSLS-II bare lattice for 1000 turns calculated in Elegant. . . . .                                                                                                                                                                                               | 58 |
| Figure 3.15 | Frequency map (a) and tune map (b) of the crab cavity crossing at the EIC using the parameters in Table 3.1. . . . .                                                                                                                                                                                              | 62 |
| Figure 3.16 | Convergence map (a) and tune map (b) of the crab cavity crossing at the EIC using the parameters in Table 3.1. . . . .                                                                                                                                                                                            | 63 |
| Figure 3.17 | Area where we see an increase in the longitudinal tunes compared to the linear tune from Fig. 3.16. . . . .                                                                                                                                                                                                       | 63 |
| Figure 3.18 | Convergence map (a) and tune map (b) with a time independent sextupole and no crab cavity. . . . .                                                                                                                                                                                                                | 64 |
| Figure 3.19 | Convergence map using the parameters in table 3.1 but with no sextupole ( $b_3 = 0$ ). . . . .                                                                                                                                                                                                                    | 64 |
| Figure 4.1  | An illustration of a 4-D representation of the neural networks that were used. The encoder is our representation of the Koopman operator's eigenfunctions. The Linear section applies the Koopman operator with eigenvalues from the Linear parameters. The decoder is the inverse of our eigenfunctions. . . . . | 72 |
| Figure 4.2  | The phase space of the standard map with $K = \mathbf{X}$ (a) and the same phase space after being passed through the encoder to the Koopman subspace (b). . . . .                                                                                                                                                | 77 |
| Figure 4.3  | The tune shift with amplitude calculated from NAFF (blue) and the model (orange). In red are the model's predictions during previous epochs, getting lighter the older the epoch. . . . .                                                                                                                         | 77 |
| Figure 4.4  | The amplitude of the motion of the standard map in the Koopman subspace. . . . .                                                                                                                                                                                                                                  | 78 |
| Figure 4.5  | The amplitude of the motion of the standard map in the C-S formalism (blue) vs the same orbits in the Koopman subspace (red). . . . .                                                                                                                                                                             | 78 |
| Figure 4.6  | Orbits of the standard map plotted by action vs change in phase comparing the C-S formalism (blue) to the Koopman subspace (red). . . . .                                                                                                                                                                         | 79 |
| Figure 4.7  | Orbits of the standard map plotted by action vs change in phase in the C-S formalism (blue) and the Koopman subspace (red). . . . .                                                                                                                                                                               | 80 |

|             |                                                                                                                                                                                                                |    |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 4.8  | The phase space of the McMillan map (a) and the same phase space after being passed through the encoder to the Koopman subspace (b). . . . .                                                                   | 81 |
| Figure 4.9  | The tune shift with amplitude for the McMillan map calculated from NAFF (blue) and the model (orange). In red are the model's predictions during previous epochs, getting lighter the older the epoch. . . . . | 81 |
| Figure 4.10 | The amplitude of the motion of the standard map in the Koopman subspace. . .                                                                                                                                   | 82 |
| Figure 4.11 | The amplitude of the motion of the standard map in the C-S formalism (blue) vs the same orbits in the Koopman subspace (red) plotted by turn. . . . .                                                          | 82 |
| Figure 4.12 | Orbits of the standard map plotted by action vs change in phase comparing the C-S formalism (blue) to the Koopman subspace (red). . . . .                                                                      | 83 |
| Figure 4.13 | Orbits of the standard map plotted by action vs change in phase in the C-S formalism (blue) and the Koopman subspace (red). . . . .                                                                            | 84 |
| Figure 4.14 | The phase space of the Standard map (a) and the same phase space after being passed through the encoder to the Koopman subspace in the noisy model (b). . .                                                    | 85 |
| Figure 4.15 | The tune shift with amplitude calculated from NAFF (blue) and the model (orange). In red are the noisy model's c . . . . .                                                                                     | 86 |
| Figure 4.16 | The amplitude of the motion of the standard map in the noisy Koopman subspace. . .                                                                                                                             | 86 |
| Figure 4.17 | The amplitude of the motion of the standard map in the C-S formalism (blue) vs the same orbits in the noisy Koopman subspace (red) plotted by turn. . . . .                                                    | 87 |
| Figure 4.18 | Orbits of the standard map plotted by action vs change in phase comparing the C-S formalism (blue) to the noisy Koopman subspace (red). . . . .                                                                | 88 |
| Figure 4.19 | Orbits of the standard map plotted by action vs change in phase in the C-S formalism (blue) and the noisy Koopman subspace (red). . . . .                                                                      | 89 |

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction to Particle Accelerators

Particle accelerators have been an important tool for research in several areas of scientific research such as nuclear physics, high-energy physics, biology, and chemistry. They can be used to create high-energy particle collisions to study the properties of atoms, create focused light beams to give insight into chemical reactions, and more. The beam in an accelerator is made up of groups of charged particles. The specific particle can change depending on the use case, but it needs to be charged to be steered by magnetic fields. They are grouped into bunches comprised of billions of individual particles. The accelerator itself consists of a pipe called a beam pipe. This pipe is kept at a vacuum, so particles in the beam do not collide with or ionize any gas particles. Along this beamline, there are various diagnostic components and, most importantly for our purpose, magnets that generate the magnetic fields that will steer the beam and determine the dynamics of the particles. What components are and how they are arranged along the beamline is called a lattice. The effects the components in the lattice have on the particle's dynamics can be categorized as linear and nonlinear. Linear dynamics is well understood, and there are many techniques to approach them with. This chapter will cover the conventions used in accelerator physics for linear dynamics. These conventions will serve as the basis for the study of nonlinear dynamics. It is important to understand the nonlinear dynamics surrounding particle motion inside the accelerator to optimize it for various use cases. For example, a collider for high-energy or nuclear physics is used to study the reactions between particles colliding, so the accelerator should maximize the number of collisions or luminosity. Synchrotron light sources, instead, need to be optimized to increase the brightness of the beam for applications such as high-resolution x-rays.

The full description of the dynamics of a particle in an accelerator requires a six-dimensional phase space, consisting of a position and momentum pair for each spatial dimension. For linear motion, this can often be reduced to three uncoupled two-dimensional systems. These six dimensions are typically grouped into a four-dimensional transverse direction (perpendicular to the beamline)

and a two-dimensional longitudinal direction.

Different types of magnet are used in the lattice to control the motion of the beam. Dipoles are used to steer the beam along the designed trajectory. Quadrupoles provide focusing (and defocusing) forces to confine and guide the particles transversely for transporting the beam. The sextupoles are the first magnetic multipole that introduces nonlinear effects. They are used to correct for chromaticity effects, which arise from a particle's focal length changing based on **beam** energy deviation. Octopoles, which introduce additional nonlinear dependence on the position of the particles, are used to mitigate collective effects and prevent beam instabilities and emittance growth.

One way to categorize accelerators is as linear or circular, based on the geometry of their beam trajectories. In a linear accelerator, the beam will only pass through the lattice once. In contrast, a circular accelerator will have the beam pass through the lattice multiple times, where they can be repeatedly accelerated (synchrotrons) or used for scientific research (storage rings). An example of a circular accelerator is the Large Hadron Collider (LHC) at CERN [5]. To accelerate protons from injection energy (450 GeV) to 7 TeV, it will repeatedly pass particles through radio frequency cavities (RF) that accelerate particles with oscillating electric fields. It will then maintain this energy level for hours of collisions, which can be studied for high-energy physics research.

The particles will lose energy as they pass through the ring due to synchrotron radiation. Maxwell's laws show that as charged particles accelerate on a curved path, they will radiate light. So, RF cavities are introduced to compensate for this or exceed it if they need to accelerate the particles longitudinally. This effect is mainly for electron beams, which have a very large charge-to-mass ratio. Hadron beams, such as proton beams, do not have as strong an effect, so this phenomenon is usually negligible [6]. Another effect of the RF cavity is that it will focus the beam in the longitudinal plane by providing energy to particles that are below the reference energy and dampen particles that are above the reference energy. Our studies will currently only look at the transverse coordinates in either a 2-D or a 4-D phase space and, therefore, will ignore the longitudinal effects for the most part; however, future studies would need to include these in a full

6-D phase space.

As particles pass through circular accelerators, some may drift towards infinity and be lost. The region where these particles are stable and not lost is called the dynamic aperture. For linear motion, we expect an infinitely large aperture; however, the inclusion of components like sextupoles means accelerators are generally very nonlinear systems. It is then important that we calculate the dynamic aperture. The common approach to this is by brute force particle tracking. We start with a model of particle motion through an accelerator or map  $F$ .

$$\mathbf{X}_{n+1} = F(\mathbf{X}_n) \tag{1.1}$$

$$\vec{X}_n \in \mathbb{R}^6$$

$$F : \mathbb{R}^6 \rightarrow \mathbb{R}^6$$

where  $\mathbf{X}_n$  is the horizontal, vertical, and longitudinal position and momenta after the  $n^{\text{th}}$  pass through the accelerator. As previously stated, if  $F$  is linear, then all particles will be stable, and a nonlinear map  $F$  can have certain particles lost. The motion of a particle in a Hamiltonian system will lie on a  $2N$ -D torus in phase space, where  $N$  is the degrees of freedom [7]. For  $N \geq 2$ , there will no longer be any tori that divide the phase space area [7]. This means that, given enough time, a stable particle can eventually trail off towards infinity and be lost in a process called Arnold Diffusion [1]. By picking samples across the transverse plane and repeatedly passing them through  $F$ , we can see where the particles are after that many turns and therefore which ones are lost (position tends towards infinity) and which ones stay stable.

While this brute force method is the most accurate approach, it is not always practical. This is because some accelerators will leave particles circulating for hours. Such is the case for the Recycler Ring (RR) at the Fermi National Accelerator Laboratory (Fermilab). The RR circulates a proton beam to accelerate the particles into the main ring at Fermilab [8]. The cost to calculate turns for thousands of particles across millions of turns quickly becomes too computationally expensive to be practical. Especially since the calculations would need to be redone every time the lattice

design is changed.

If we are dealing with electron particles in a circular accelerator, then the beam will experience a strong damping force from the synchrotron radiation. This means that a particle will enter a stable state within a few thousand turns in most cases. So this "turn-by-turn" particle tracking is a fine approach to take to calculate an accurate dynamic aperture. However, for a proton ring like the LHC or the Fermilab RR, this damping force is much weaker due to the smaller charge-to-mass ratio. So it would take much longer for particles to reach a stable state. So it becomes necessary to develop other approaches to calculate the dynamic aperture for these systems.

Before going into nonlinear dynamics, the following section will go into some details on linear dynamics and some established approaches to calculating the dynamic aperture. The chapters following that will talk about two distinct approaches to these calculations that have been developed over the last few years.

## 1.2 Linear Dynamics

Various mathematical methods have been developed to analyze nonlinear systems in a general sense such as normal form, Lie operators, and perturbation theory. Before going into detail about these methods it is important to review the mathematical formalisms surrounding particle accelerators so we can build off of them in more complicated systems.

The motion of charged particles in an accelerator is governed by the Lorentz force.

$$\vec{F} = q \left( \vec{E} + \vec{v} \times \vec{B} \right) \quad (1.2)$$

where  $\mathbf{E}$  is the electric field,  $\mathbf{B}$  is the magnetic field,  $\mathbf{v}$  is the velocity of the particle and  $q$  is the charge of the particle. Much of the lattice is just for steering the beam and is designed to have no electric field ( $\mathbf{E} = 0$ ), as electric fields are used to accelerate the particles. The simplest magnetic component in these sections is the dipole. The dipole is a constant magnetic field ( $\mathbf{B} \equiv B_o \hat{y}$ ) and simply turns the beam so it travels in a circle. From the Lorenz equation in this type of field, we can calculate a useful parameter called beam rigidity. This will be used to help normalize our

calculations to be energy-independent. It also tells us how difficult it is to steer the beam. A larger beam rigidity means we need a stronger magnetic field to keep the same path.

$$\frac{|\mathbf{P}|}{q} = B_o \rho \quad (1.3)$$

where  $\mathbf{P}$  is the momentum, and  $\rho$  is the bending radius of the beam (the radius of the circular path that the particle travels in this field). This gives us the beam rigidity,  $B\rho$ .

The transverse magnetic components in an accelerator are described by the multipole expansion. The accepted convention is to combine the transverse magnetic fields into a single complex number called the Beth representation:

$$B_y + iB_x = B_0 \sum_{n=0}^{\infty} (b_n + ia_n) (x + iy)^n \quad (\text{U.S. convention}) \quad (1.4)$$

$$= B_0 \sum_{m=1}^{\infty} (b_m + ia_m) (x + iy)^{m-1} \quad (\text{European convention}) \quad (1.5)$$

where  $B_o$  is the normalization field, which is usually chosen as the main dipole field strength, and  $b_n$  and  $a_n$  are the  $2(n + 1)^{\text{th}}$  multipole coefficients.  $b_n$  terms describe the normal fields and  $a_n$  describe the skew fields. The skew field is the same multipole order as the normal field but offset by an angle of  $\frac{\pi}{2(n+1)}$  radians.

### 1.3 Hamiltonian Dynamics in Accelerators

The Lagrangian for a charged particle in an electric and magnetic field is

$$L = -\frac{mc^2}{\gamma} - q\phi + q\vec{A} \quad (1.6)$$

where  $\gamma$  is the relativistic Lorentz factor ( $\gamma = 1/\sqrt{1 - v^2/c^2}$ ),  $\vec{A}$  is the vector potential,  $\Phi$  is the scalar potential,  $q$  is the charge of the particle, and  $m$  is its mass. From this, we can define the canonical momentum, which allows us to extend to Hamiltonian dynamics in a way that is analogous to classical mechanics.

$$\vec{P} = \frac{\partial L}{\partial \vec{v}} = \gamma m \vec{v} - q \vec{A} \quad (1.7)$$

The Hamiltonian for a charged particle is

$$H = \vec{v} \cdot \vec{P} - L = c \sqrt{m^2 c^2 + (\vec{P} - q \vec{A})^2} + q \Phi \quad (1.8)$$

This gives us our conjugate coordinate pairs of

$$\dot{y} = \frac{\partial H}{\partial P_y} \quad (1.9)$$

$$\dot{P}_y = -\frac{\partial H}{\partial y} \quad (1.10)$$

where one can substitute  $y$  for any degree of freedom ( $x, y, z$ ).

It is also common for accelerators to have a specific reference orbit that forms a closed loop when passing through the lattice. We then define our coordinate system in terms of this closed orbit being the origin.

If the reference orbit is defined by the path  $\vec{r}_0(s)$  where  $s$  is our independent variable referring to the length along the path  $\vec{r}_o(s)$ .

$$s(t) = \int_0^t |\vec{r}'_0(t')| dt' \quad (1.11)$$

The unit vector along the accelerator  $\hat{s}$  is then

$$\hat{s}(s) = \frac{d\vec{r}_0(s)}{ds} \quad (1.12)$$

Our tangent unit vector is then defined by

$$\hat{x}(s) = -\rho(s) \frac{d\hat{s}(s)}{ds} \quad (1.13)$$

with  $\rho(s)$  is the local radius which can change depending on where we are in the accelerator. For example, circular accelerators have some radius that defines the geometry of the entire ring



but they are not perfect circles. Rather they are straight sections of magnetic lenses (quadrupole, sextupoles, etc) and other components with the dipoles spaced throughout the ring, bending the beam by a radius tighter than the radius of the overall ring. Then to have a orthonormal coordinate system we define the last transverse unit vector simply as

$$\hat{y}(s) = \hat{s}(s) \times \hat{x}(s) \quad (1.14)$$

If our reference orbit exists on a plane then the derivatives of the basis vectors are

$$\frac{d\hat{s}(s)}{ds} = -\kappa(s) \hat{x}(s) \quad (1.15)$$

$$\frac{d\hat{x}(s)}{ds} = \kappa(s) \hat{s}(s) \quad (1.16)$$

$$\frac{d\hat{y}(s)}{ds} = 0 \quad (1.17)$$

where  $\kappa$  is the curvature of the path. From there any single particle's position can be written in the form  $\vec{r}(s)$

$$\vec{r}(s) = \vec{r}_0(s) + x\hat{x}(s) + y\hat{y}(s) \quad (1.18)$$

It is later helpful to do another change of variables. In order to preserve the form of Hamilton's equations of motion we can use canonical transformations. If the old coordinates are  $(\vec{y}, \vec{p})$  and the new coordinates are  $(\vec{Y}, \vec{P})$  then the four types of generating functions are

$$\begin{aligned} F_1 &= F_1(\vec{y}, \vec{Y}, t) & \vec{p} &= \frac{\partial F_1}{\partial \vec{y}}; \vec{P} = -\frac{\partial F_1}{\partial \vec{Y}} \\ F_2 &= F_2(\vec{y}, \vec{P}, t) & \vec{p} &= \frac{\partial F_2}{\partial \vec{y}}; \vec{Y} = \frac{\partial F_2}{\partial \vec{P}} \\ F_3 &= F_3(\vec{p}, \vec{Y}, t) & \vec{y} &= -\frac{\partial F_3}{\partial \vec{p}}; \vec{P} = -\frac{\partial F_3}{\partial \vec{Y}} \\ F_4 &= F_4(\vec{p}, \vec{P}, t) & \vec{y} &= -\frac{\partial F_4}{\partial \vec{p}}; \vec{Y} = \frac{\partial F_4}{\partial \vec{P}} \end{aligned} \quad (1.19)$$

From Equation 1.19 a new Hamiltonian can be calculated in terms of  $(\vec{Y}, \vec{P})$ .

$$\tilde{H}(\vec{Y}, \vec{P}, t) = H(\vec{y}(\vec{Y}, \vec{P}), \vec{p}(\vec{Y}, \vec{P})) + \frac{\partial F}{\partial t} \quad (1.20)$$

Equation 1.18 is called to at the Frenet-Serret coordinate system [6]. To get our Hamiltonian in this coordinate system we use a generating function of type  $F_3$  from Equation 1.19.

$$F_3 = -\vec{P} \cdot (\vec{r}_0(s) + x\hat{x}(s) + y\hat{y}(s)) \quad (1.21)$$

where  $\vec{P}$  is the momentum in cartesian coordinates. The canonical momentum in the Frenet-Serret coordinate system is  $\vec{\Pi}$

$$\Pi_x = -\frac{\partial F_3}{\partial x} = \vec{P} \cdot \hat{x} \quad (1.22)$$

$$\Pi_y = -\frac{\partial F_3}{\partial y} = \vec{P} \cdot \hat{y} \quad (1.23)$$

$$\Pi_s = -\frac{\partial F_3}{\partial s} = \vec{P} \cdot \hat{s} \left(1 + \frac{x}{\rho}\right) \quad (1.24)$$

$$(1.25)$$

Which gives us our new Hamiltonian,  $\tilde{H}$

$$H = c\sqrt{(mc)^2 + (\Pi_x - qA_x)^2 + (\Pi_y - qA_y)^2 + \left(\frac{\Pi_s}{1 + x/\rho} - qA_s\right)^2} + q\phi \quad (1.26)$$

Next we want the independent variable to be  $s$ , the position of the reference particle along the beamline, instead of  $t$  time. We start with Hamilton's equation of motion

$$\dot{x}_i = \frac{\partial H}{\partial \Pi_{x_i}} \quad (1.27)$$

Then to find  $x'$  which is the derivative with respect to  $s$  instead of  $t$

$$x'_i = \frac{dx}{ds} = \frac{dx_i}{dt} \frac{dt}{ds} = \dot{x}_i \left(\frac{ds}{dt}\right)^{-1} = \frac{\partial H}{\partial \Pi_{x_i}} \left(\frac{\partial H}{\partial \Pi_s}\right)^{-1} = -\frac{\partial \Pi_s}{\partial \Pi_{x_i}} \quad (1.28)$$

Therefore

$$x' \equiv \frac{dx}{ds} = -\frac{\partial \Pi_s}{\partial \Pi_x} \quad (1.29)$$

$$\Pi_x' \equiv \frac{d\Pi_x}{ds} = \frac{\partial \Pi_s}{\partial x} \quad (1.30)$$

$$y' \equiv \frac{dy}{ds} = -\frac{\partial \Pi_s}{\partial \Pi_y} \quad (1.31)$$

$$\Pi_y' \equiv \frac{d\Pi_y}{ds} = \frac{\partial \Pi_s}{\partial y} \quad (1.32)$$

$$(1.33)$$

Which means our new Hamiltonian is  $\Pi_s$ . This also leads to  $(t, -H)$  being a pair of conjugate variables. So we can adjust that new Hamiltonian to read as

$$\tilde{H} = -\left(1 + \frac{x}{\rho}\right) \sqrt{\left(\frac{H - q\Phi}{c}\right)^2 - (mc)^2 - (\Pi_x - qA_x)^2 - (\Pi_y - qA_y)^2} - qA_s \left(1 + \frac{x}{\rho}\right) \quad (1.34)$$

There are ways that we can simplify this Hamiltonian. For one, since the magnetic field is just used to steer the particles it is 2-D. Meaning, we only the longitudinal component of the vector potential  $A_s$  will be non-zero. This also means that our canonical momentum in the transverse plane,  $\Pi_{x/y}$ , are equal to the cartesian momenta,  $p_{x/y}$ .

If we do not have a time independent field then another canonical transformation can be performed to change  $(t, -H)$  into  $(t, h)$  where  $h = H - e\Phi$ . Much of the time  $\Phi = 0$  outside the RF cavities so this is a reasonable assumption. Notice that  $h$  is the total energy of the particle, and the momentum of the particle is therefore  $p = \sqrt{(h/c)^2 - (mc)^2}$ . If we define the momentum of the reference particle as  $p_0$  then we can define the momentum of the particle as  $p = p_0(1 + \delta)$  where  $\delta$  is called the fractional momentum deviation. Combining all these simplifications together and dividing by a factor of the reference momentum  $p_0$  we obtain the new form of the Hamiltonian  $\tilde{H}$  which we will call  $K$ .

$$\frac{\tilde{H}}{p_0} \equiv K \left( x, \tilde{p}_x = \frac{p_x}{p_0}, y, \tilde{p}_y = \frac{p_y}{p_0}, t, -h = \frac{-H}{p_0} \right) \quad (1.35)$$

$$= - \left( 1 + \frac{x}{\rho} \right) \sqrt{\left( \frac{h}{c} \right)^2 - \left( \frac{mc^2}{p_0} \right)^2 - \tilde{p}_x^2 - \tilde{p}_y^2} - q \frac{A_s}{p_0} \left( 1 + \frac{x}{\rho} \right) \quad (1.36)$$

We still might not want  $(t, -h)$  to be our variables so we will do a final canonical transformation to use the variables  $(z, \delta)$ . We use the generating function  $F_2 = -ht$ .

$$z = \frac{\partial F_2}{\partial \delta} \quad (1.37)$$

$$= \frac{\partial}{\partial \delta} \left( - \sqrt{c^2(1 + \delta)^2 + (mc^2)^2} \right) t \quad (1.38)$$

$$= \frac{-c^2(1 + \delta)t}{\sqrt{c^2(1 + \delta)^2 + (mc^2)^2}} \quad (1.39)$$

$$= -\beta ct \quad (1.40)$$

Finally, the Hamiltonian is

$$\tilde{K}(x, \tilde{p}_x, y, \tilde{p}_y, z = -\beta ct, \delta, s) = - \left( 1 + \frac{x}{\rho} \right) \sqrt{(1 + \delta)^2 - \tilde{p}_x^2 - \tilde{p}_y^2} - q \frac{A_s}{p_0} \left( 1 + \frac{x}{\rho} \right) \quad (1.41)$$

Since the particle's momentum is close to the reference momentum we can assume  $\delta \ll 1$ . We can then also assume that the transverse momentum  $\tilde{p}_{x/y} \ll 1$ . This allows us to approximate our Hamiltonian further. We drop the tilde for convenience.

$$\begin{aligned} K(x, p_x, y, p_y, z, \delta, s) &= -(1 + \delta) \left( 1 + \frac{x}{\rho} \right) \left( 1 - \frac{p_x^2}{2(1 + \delta)^2} - \frac{p_y^2}{2(1 + \delta)^2} \right) - q \frac{A_s}{p_0} \left( 1 + \frac{x}{\rho} \right) \\ &= - \left( 1 + \frac{x}{\rho} \right) \left( 1 + \delta - \frac{p_x^2}{2(1 + \delta)} - \frac{p_y^2}{2(1 + \delta)} \right) - q \frac{A_s}{p_0} \left( 1 + \frac{x}{\rho} \right) \end{aligned} \quad (1.42)$$

In this approximation the normalized transverse momenta

$$\tilde{p}_x = \frac{\vec{p} \cdot \hat{x}}{p_0} \quad (1.43)$$

$$\tilde{p}_y = \frac{\vec{p} \cdot \hat{y}}{p_0} \quad (1.44)$$

and the derivatives of the position

$$x' = \frac{dx}{ds} = \frac{\vec{p} \cdot \hat{x}}{p_s} \quad (1.45)$$

$$y' = \frac{dy}{ds} = \frac{\vec{p} \cdot \hat{y}}{p_s} \quad (1.46)$$

are approximately equal so we can use these interchangeable from here on out.

#### 1.4 Magnetic Components

As stated in Section 1.3 our magnetic fields can be reasonably approximated by a 2-D transverse field due to the edge effects of the magnet dominating the dynamics. So, we can approximate our magnetic field with the form

$$\vec{B} = B_x(x, y)\hat{x} + B_y(x, y)\hat{y} \quad (1.47)$$

Which in the form of the vector potential,  $\vec{B} = \nabla \times \vec{A}$ , becomes

$$B_x = -\frac{1}{h_s} \frac{\partial A_s}{\partial y} \quad (1.48)$$

$$B_y = \frac{1}{h_s} \frac{\partial A_s}{\partial x} \quad (1.49)$$

Maxwell's equation tells us that the curl of the magnetic field must be zero ( $\nabla \times \vec{B} = 0$ ) we then now that  $A_s$  must satisfy the following:

$$\frac{\partial}{\partial y} \frac{1}{h_s} \frac{\partial A_s}{\partial y} + \frac{\partial}{\partial x} \frac{1}{h_s} \frac{\partial A_s}{\partial x} = 0 \quad (1.50)$$

For a dipole we know that  $\vec{B} = B_o(s)\hat{y}$ . Therefore,

$$B_0(s) = -\frac{1}{1 + \kappa x} \frac{\partial ((1 + \kappa x)A_s)}{\partial x} \quad (1.51)$$

where  $\kappa$  is the curvature, which for a dipole is  $1/\rho$ . But since the horizontal offset  $x$  is much less than the turning radius of the accelerator we can assume that  $|x|\kappa \ll 1$ . So we can approximate the form of  $A_s$  as

$$A_s = -B_0(s)(ax + bx^2) \quad (1.52)$$

which after simplifying Equation 1.51 becomes

$$1 + \kappa x = \frac{\partial ((1 + \kappa x)(ax + bx^2))}{\partial x} \quad (1.53)$$

This gives us  $a = 1$  and  $b = -\kappa/2$ . Therefore the approximate vector potential for a dipole is

$$A_s \approx -B_0 \left( x - \frac{x^2}{2\rho} \right) \quad (1.54)$$

Then given the beth representation of a quadrupole, which we can gain from Equation 1.5, we can find the vector potential for a quadrupole becomes

$$A_{s,q} = -B_0 b_1(s) (x^2 - y^2) / 2 = -G(s) (x^2 - y^2) / 2 \quad (1.55)$$

Where  $G(s)$  is the field gradient of the magnetic field. Since all linear magnetic components are either quadrupoles or dipoles, the Hamiltonian we calculated from Equation 1.42 becomes

$$K(x, p_x, y, p_y, s) \approx -1 + \frac{p_x^2}{2} + \frac{p_y^2}{2} + \frac{x^2}{2\rho^2} + k(s) (x^2 - y^2) / 2 \quad (1.56)$$

for linear accelerator systems. This also shows the usefulness of the magnetic rigidity we calculated in Equation 1.3, which allows us to have the normalized quadrupole strength,  $k(s) = G(s)/B\rho$ , that gives us our energy independent Hamiltonian for linear systems.

Since Hamilton's equations of motion are based on derivatives we can ignore the constant term. From there we see that there is no coupling between the transverse directions. So we can write the Hamiltonian as the sum of an  $x$  and a  $y$  Hamiltonian, i.e.

$$K(x, p_x, y, p_y, s) = K_x(x, p_x, s) + K_y(y, p_y, s) \quad (1.57)$$

where

$$K_x(x, p_x, s) = \frac{p_x^2}{2} + \frac{x^2}{2\rho^2} + \frac{k(s)x^2}{2} \quad (1.58)$$

$$K_y(y, p_y, s) = \frac{p_y^2}{2} - \frac{k(s)y^2}{2} \quad (1.59)$$

Then we can calculate the  $x$  and  $y$  equations of motion separately. For  $K_x$

$$x' = \frac{\partial K_x}{\partial p_x} = p_x \quad (1.60)$$

$$p'_x = -\frac{\partial K_x}{\partial x} = -\left(k(s) + \frac{1}{\rho^2}\right)x \quad (1.61)$$

Then we can combined these two equations into one by taking the derivative of the first one.

$$x'' + \left(k(s) + \frac{1}{\rho^2}\right)x = 0 \quad (1.62)$$

Similarly we perform these steps for  $K_y$  and get

$$y'' - k(s)y = 0 \quad (1.63)$$

We can now see that the general equation for the linear dynamics in quadrupoles and dipoles is in the form of Hill's equation.

$$\frac{d^2x}{ds^2} + f(s)x = 0 \quad (1.64)$$

$$f(s + C) = f(s)$$

where  $f(s)$  is some periodic function with period  $C$  (in this case, the circumference of the ring).

We can treat the magnetic fields in shorter and shorter slices until there is not much longitudinal motion. Then we can treat the magnetic fields as piecewise constant values.

For constant values of  $k$ , we can look at three cases:  $k > 0$  is a focusing quadrupole,  $k = 0$  is a drift space, and  $k < 0$  is a defocusing quadrupole. Assuming these are not also dipoles, i.e.  $\frac{1}{\rho^2} \rightarrow 0$ , then the solutions are:

$$x(s) = \begin{cases} a \cos(\sqrt{k}s) + b \sin(\sqrt{k}s) & k > 0 \\ as + b & k = 0 \\ a \cosh(\sqrt{-k}s) + b \sinh(\sqrt{-k}s) & k < 0 \end{cases} \quad (1.65)$$

In the dipole sections where  $\rho$  is constant and  $k = 0$ , we get the solution:

$$x(s) = a \cos\left(\frac{s}{\rho}\right) + b \sin\left(\frac{s}{\rho}\right) \quad (1.66)$$

$$y(s) = as + b \quad (1.67)$$

## 1.5 Matrix Approach

These continuous equations are accurate but it can be more useful to describe the positions and momenta at discrete points before and after each magnetic component in the lattice since the exact motion inside each component is not important. For linear motion this means for a particle described by the state  $\mathbf{X}$  where

$$\mathbf{X} = \begin{pmatrix} x \\ p \end{pmatrix} \quad (1.68)$$

one can find a matrix  $M$  for each component that describes the state of the particle leaving the component in terms of the state of the particle entering, i.e.

$$\mathbf{X}_{\text{exit}} = M \mathbf{X}_{\text{entrance}} \quad (1.69)$$

$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \quad (1.70)$$



Note that while these descriptions are for 2-D motion, motion in the transverse directions  $x$  and  $y$  are uncouple in the linear case and behave similarly. So we can look at one direction for simplicity. We will discuss the longitudinal motion later as it requires special treatment and often not dependent on the transverse position.

Assuming a component length  $l$ , Equation 1.65 becomes

$$M_{\text{quad}} = \begin{cases} \begin{pmatrix} \cos(\sqrt{k}l) & \sin(\sqrt{k}l)/\sqrt{k} \\ -\sqrt{k}\sin(\sqrt{k}l) & \cos(\sqrt{k}l) \end{pmatrix} & k > 0, \text{Focusing Quad} \\ \begin{pmatrix} \cosh(\sqrt{-k}l) & \sinh(\sqrt{-k}l)/\sqrt{-k} \\ \sqrt{-k}\sinh(\sqrt{-k}l) & \cosh(\sqrt{-k}l) \end{pmatrix} & k < 0, \text{Defocusing Quad} \end{cases} \quad (1.71)$$

And since the length of a quadrupole is relatively short we can define a focal length  $f$  as

$$f = \lim_{l \rightarrow 0} \frac{1}{|k|l} \quad (1.72)$$

and simplify Equation 1.71 to

$$M_{\text{quad}} = \begin{cases} \begin{pmatrix} 1 & 0 \\ -1/f & 1 \end{pmatrix} & \text{Focusing Quad} \\ \begin{pmatrix} 1 & 0 \\ 1/f & 1 \end{pmatrix} & \text{Defocusing Quad} \end{cases} \quad (1.73)$$

Similarly for a dipole Equation 1.66 shows

$$M = \begin{pmatrix} \cos\left(\frac{l}{\rho}\right) & \rho \sin\left(\frac{l}{\rho}\right) \\ -\frac{1}{\rho} \sin\left(\frac{l}{\rho}\right) & \cos\left(\frac{l}{\rho}\right) \end{pmatrix} \quad (1.74)$$

which in the  $\frac{l}{\rho} \ll 1$  limit becomes a drift space

$$M_{\text{drift}} = \begin{pmatrix} 1 & l \\ 0 & 1 \end{pmatrix} \quad (1.75)$$

These matrices can be multiplied together to get a matrix to describe the full ring:

$$M(s_n, s_0) = M(s_n, s_{n-1}) \cdots M(s_2, s_1) M(s_1, s_0) \quad (1.76)$$

where  $M(s_i, s_j)$  is the matrix describing the motion of the particle from  $s_j$  to  $s_i$  along the ring.

The linear dynamics of the ring can then be analyzed using just this matrix. It is common to now perform a change of variables to the Courant-Snyder parameters. From the stable solution in Equation 1.65, we can rework  $x(s)$  into the form

$$x(s) = \sqrt{2J\beta(s)} \cos(\psi(s) + \psi_0) \quad (1.77)$$

where  $J$  is the action,  $\beta(s)$  is the betatron function (a Courant-Snyder Parameter),  $\psi(s)$  is the phase, and  $\psi_0$  is the initial phase. The other two CS parameters are

$$\alpha(s) = -\frac{\beta'(s)}{2} \quad (1.78)$$

$$\gamma(s) = \frac{1 + \alpha(s)^2}{\beta(s)} \quad (1.79)$$

One can then make a general transfer matrix in the form

$$M(s_1 | s_0) = \begin{pmatrix} \sqrt{\frac{\beta_1}{\beta_0}} (\cos \psi + \alpha_0 \sin \psi) & \sqrt{\beta_0 \beta_1} \sin \psi \\ -\frac{1+\alpha_0 \alpha_1}{\sqrt{\beta_0 \beta_1}} \sin \psi + \frac{\alpha_0 - \alpha_1}{\sqrt{\beta_0 \beta_1}} \cos \psi & \sqrt{\frac{\beta_0}{\beta_1}} (\cos \psi - \alpha_1 \sin \psi) \end{pmatrix} \quad (1.80)$$

$$= \begin{pmatrix} \frac{1}{\sqrt{\beta_1}} & 0 \\ \frac{\alpha_1}{\sqrt{\beta_1}} & \sqrt{\beta_1} \end{pmatrix}^{-1} \begin{pmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{\beta_0}} & 0 \\ \frac{\alpha_0}{\sqrt{\beta_0}} & \sqrt{\beta_0} \end{pmatrix} \quad (1.81)$$

which simplifies to

$$M = \begin{pmatrix} \cos \psi + \alpha \sin \Phi & \beta \sin \Phi \\ -\frac{1+\alpha^2}{\beta} \sin \Phi & \cos \psi - \alpha \sin \Phi \end{pmatrix} \quad (1.82)$$

for a periodic ring where  $\Phi$  is the phase advance across the entire ring. Let  $\mathbf{X}_k$  be the position of a particle after  $k$  turns. Looking at the first and last matrix in Equation 1.81 one can notice that we can define a new coordinates with

$$\begin{pmatrix} X \\ \mathcal{P} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{\beta_0}} & 0 \\ \frac{\alpha_0}{\sqrt{\beta_0}} & \sqrt{\beta_0} \end{pmatrix} \begin{pmatrix} x \\ p \end{pmatrix} \quad (1.83)$$

which defines out normalized coordinates. Then

$$\mathbf{X}_k = M^k \mathbf{X}_0 \quad (1.84)$$

Since  $\mathbf{X}_0$  can be expressed as

$$\mathbf{X}_0 = a \mathbf{v}_1 + b \mathbf{v}_2 \quad (1.85)$$

where  $a$  and  $b$  are constants and  $\mathbf{v}_i$  is the  $i^{\text{th}}$  eigenvector of  $M$  corresponding to the  $i^{\text{th}}$  eigenvalue,  $\lambda_i$ . Then by combining Equations 1.84 and 1.85 we see that

$$\mathbf{X}_k = M^k (a \mathbf{v}_1 + b \mathbf{v}_2) = a \lambda_1^k \mathbf{v}_1 + b \lambda_2^k \mathbf{v}_2 \quad (1.86)$$

Then it is obvious from Equation 1.86 that  $|\lambda_i| \leq 1$  must hold true in order to avoid the particle's amplitude growing turn after turn. So from the characteristic equation  $\det(M - \lambda I) = 0$

$$\lambda^2 - \text{Trace}(M)\lambda + 1 = 0 \quad (1.87)$$

Therefore, we need  $|\text{Trace}(M)| \leq 2$  or  $|\cos \Phi| \leq 1$ .

The transverse motion with these variables is referred to as betatron motion. It is also important to note the number of oscillations of the betatron function or tune  $Q$ . This is defined by

$$Q = \frac{\Phi}{2\pi} = \frac{1}{2\pi} \int_{s_0}^{s_0+C} \frac{ds}{\beta(s)} \quad (1.88)$$

where  $C$  is the circumference of the ring.

## 1.6 Courant-Snyder Parameters

The dynamics of linear motion in particle accelerators generally uses the Courant-Snyder (C-S) parameters [9]. Beginning from the general form of Hill's equation (Equation 1.64) we can solve it with the Floquet transformation 1.18

$$x(s) = aw(s)e^{i\psi(s)} \quad (1.89)$$

$$x^*(s) = aw(s)e^{-i\psi(s)}$$

where  $a$  is just a constant and  $w(s)$  is the amplitude function and  $\psi(s)$  is the phase function. We take the derivatives of Equation 1.89

$$x'(s) = aw'e^{i\psi} + iawe^{i\psi}\psi' \quad (1.90)$$

$$x''(s) = aw''e^{i\psi} + 2iaw'e^{i\psi}\psi' - awe^{i\psi}\psi'^2 + iawe^{i\psi}\psi'' \quad (1.91)$$

If we plug these into the general Hill's equation (Equation 1.64) we get

$$ae^{i\psi(s)} \{ [f(s)w(s) - w(s)\psi''(s) + w''(s)] + i[w(s)\psi'(s) + 2w'(s)\psi'(s)] \} = 0 \quad (1.92)$$

Since the real and imaginary parts must individually equal zero we get two equations that must be satisfied:

$$w(s)\psi''(s) + 2w'(s)\psi'(s) = \frac{(w^2(s)\psi'(s))'}{w(s)} = 0 \quad (1.93)$$

$$w''(s) + f(s)w(s) - w(s)\psi'^2(s) = 0 \quad (1.94)$$

Equation 1.93 tells us that  $w^2(s)\psi'(s) = 1$ . If we rewrite this relation and plug that back into Equation 1.94 we get two new equations:

$$w''(s) + f(s)w(s) - \frac{1}{w^3(s)} = 0 \quad (1.95)$$

$$\psi'(s) - \frac{1}{w^2(s)} = 0 \quad (1.96)$$

From here we will introduce the Courant-Snyder parameters. Starting with the beta and alpha functions defined as:

$$\beta(s) = w^2(s) \quad (1.97)$$

$$\alpha(s) = -\frac{1}{2}\beta'(s) = -w(s)w'(s) \quad (1.98)$$

Then the  $\psi$  function is going to define the phase advance, which can be calculated using Equation 1.96:

$$\psi(s, s_0) = \int_{s_0}^s \frac{1}{\beta(s')} ds' \quad (1.99)$$

Note that the phase advance calculated over the entire ring (from  $s_0$  to  $s_0 + C$ ) gives the betatron tune  $Q$ .

$$\psi(s_0 + C, s_0) = \int_{s_0}^{s_0+C} \frac{1}{\beta(s')} ds' = 2\pi Q \quad (1.100)$$

Plugging our beta function into Equation 1.95 we get

$$\frac{1}{2}\beta''(s) + f(s)\beta(s) - \frac{1}{\beta(s)} \left[ 1 + \frac{\beta'^2(s)}{4} \right] = 0 \quad (1.101)$$

Our new transverse equations of motion then become:

$$x(s) = a\sqrt{\beta(s)} \cos(\psi(s)) \quad (1.102)$$

$$x'(s) = -\frac{x(s)}{\beta(s)} (\tan \psi(s) + \alpha(s)) \quad (1.103)$$

## 1.7 Action-Angle Variables

The transverse Hamiltonian from Equation 1.57 is not a constant of motion, i.e.

$$\frac{dH}{ds} = \frac{\partial H}{\partial x} \frac{\partial x}{\partial s} + \frac{\partial H}{\partial p_x} \frac{\partial p_x}{\partial s} + \frac{\partial H}{\partial s} \quad (1.104)$$

$$= \frac{\partial H}{\partial s} \neq 0 \quad (1.105)$$

There is no coupling between the transverse directions in this Hamiltonian so we can work towards a constant of motion in each direction independently by looking at a Hamiltonian of the form

$$H = \frac{1}{2}(x')^2 + \frac{1}{2}fx^2 \quad (1.106)$$

Note we will no longer explicitly show the functions as dependent on  $s$  for convenience, though they still are. We aim to do a canonical transformation from the current conjugate phase space coordinates  $(x, x')$  to the new pair  $(J, \psi)$ . Where  $J$  is a constant of motion called the action. We will do this with a generating function of the form

$$F_1(x, \psi) = \int_0^x x'(\tilde{x}, \psi) d\tilde{x} \quad (1.107)$$

$$= -\frac{x^2}{2\beta} \left( \tan \psi - \frac{\beta'}{2} \right) \quad (1.108)$$

Our action,  $J$ , is then

$$J = -\frac{\partial F_1}{\partial \psi} = \frac{x^2}{2\beta} \sec^2 \psi \quad (1.109)$$

$$= \frac{x^2}{2\beta} \left( 1 + \left( \frac{\beta x'}{x} + \alpha \right)^2 \right) \quad (1.110)$$

$$= \frac{1}{2\beta} \left( x^2 + (\beta x' + \alpha x)^2 \right) \quad (1.111)$$

The new Hamiltonian  $\tilde{H}$  is then

$$\tilde{H} = H + \frac{\partial F_1}{\partial s} \quad (1.112)$$

$$= \frac{1}{2} \frac{x^2}{\beta^2} (\tan \psi + \alpha)^2 + \frac{1}{2} K x^2 + \quad (1.113)$$

$$\begin{aligned} & \frac{x^2}{2\beta^2} \beta' \left( \tan \psi - \frac{\beta'}{2} \right) - \frac{x^2}{2\beta} \left( \psi' \sec^2 \psi - \frac{\beta''}{2} \right) \\ &= \frac{J}{\beta} \end{aligned} \quad (1.114)$$

We can use Equation 1.102 to get  $\psi$  explicitly

$$x' = -\frac{x}{\beta} (\tan \psi + \alpha) \quad (1.115)$$

$$\frac{-\beta x'}{x} = \tan \psi + \alpha \quad (1.116)$$

$$\tan \psi = -\frac{\beta x' + \alpha x}{x} = -\frac{\sqrt{\beta} x' + \alpha x / \sqrt{\beta}}{x / \sqrt{\beta}} \quad (1.117)$$

$$\psi = \arctan \left( -\frac{\sqrt{\beta} x' + \alpha x / \sqrt{\beta}}{x / \sqrt{\beta}} \right) \quad (1.118)$$

And  $J$  is explicitly

$$J = \frac{1}{2} \left( \left( \frac{x}{\sqrt{\beta}} \right)^2 + \left( \sqrt{\beta} x' + \frac{\alpha}{\sqrt{\beta}} x \right)^2 \right) \quad (1.119)$$

Our new Hamilton equations of motion are then

$$\frac{dJ}{ds} = -\frac{\partial \tilde{H}}{\partial \psi} = -\frac{\partial}{\partial \psi} \frac{J}{\beta} = 0 \quad (1.120)$$

$$\frac{d\psi}{ds} = \frac{\partial \tilde{H}}{\partial J} = \frac{1}{\beta} \quad (1.121)$$

Which shows us that  $J$  is a constant of motion. The beta function prevents the Hamiltonian itself from being a constant of motion, so we define another generating function to go from  $(J, \psi)$  to  $(\bar{J}, \bar{\psi})$ .

$$F_2(\psi, \bar{J}, s) = \left( \psi - \int_0^s \frac{1}{\beta(s')} ds' + 2\pi Q \frac{s}{C} \right) \bar{J} \quad (1.122)$$

Our new action angle variables are then

$$\bar{\psi} = \psi - \int_0^s \frac{1}{\beta(s')} ds' + 2\pi Q \frac{s}{C} \quad (1.123)$$

$$\bar{J} = J \quad (1.124)$$

This gives us the Hamiltonian  $\bar{H}$

$$\bar{H} = \tilde{H} + \frac{\partial F_2(\psi, \bar{J})}{\partial s} = \frac{2\pi}{C} Q \bar{J} \quad (1.125)$$

By scaling the Hamiltonian by a factor of  $C/2\pi$  we get the final form of our Hamiltonian

$$H(\bar{\psi}, \bar{J}) = Q \bar{J} \quad (1.126)$$

With our equation of motion being

$$x = \sqrt{2\beta\bar{J}} \cos \left( \bar{\psi} + \int_0^s \frac{1}{\beta(s')} ds' - Q\theta \right) \quad (1.127)$$

$$\alpha x + \beta x' = -\sqrt{2\beta\bar{J}} \sin \left( \bar{\psi} + \int_0^s \frac{1}{\beta(s')} ds' - Q\theta \right) \quad (1.128)$$



As a result of the new Hamiltonian  $H$  being independent of the angle  $\bar{\psi}$  Hamilton's equations of motion become

$$\frac{d\bar{J}}{ds} = -\frac{\partial H}{\partial \bar{\psi}} = 0 \quad (1.129)$$

$$\frac{d\bar{\psi}}{ds} = \frac{\partial H}{\partial \bar{J}} = Q(J) \quad (1.130)$$

where  $Q(J)$  is the frequency of oscillation, which in the general case can be dependent on the action, and as Equation 1.129 shows, the action is a constant of motion. If we are able to find an action such that the Hamiltonian only depends on that action as shown in Equation 1.126 then we call the system integrable [7].

## 1.8 Linear Coupling

So far the transverse motion has been independent from one another. That is because we have only used ideal quadrupoles and dipoles. There are other linear elements that can introduce coupling between the transverse motion. This can be intentionally like is the case with solenoids, which are often used in low energy sections of the beam as they are not effective at high energies, and skew quadrupoles. Skew quadrupoles can be implemented intentionally but more frequently they are a results of the magnets being placed with some small error in the angle of the magnet. Higher order magnetic components like sextupoles, octupole, etc. can also create linear coupling as there is a cascading effect onto the lower order fields when there is an error in the position of these magnets. The matrix of a skew quadrupole magnet is

$$M_{sq} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1/kl & 0 \\ 0 & 0 & 1 & 0 \\ 1/kl & 0 & 0 & 1 \end{pmatrix}. \quad (1.131)$$

A solenoid has the transfer matrix

$$M_{\text{sol}} = \begin{pmatrix} \cos^2(\theta_s) & \frac{\sin(2\theta_s)}{2g_s} & -\frac{\sin(2\theta_s)}{2} & \frac{\cos(2\theta_s)-1}{2g_s} \\ -\frac{g_s \sin(2\theta_s)}{2} & \cos^2(\theta_s) & \frac{g_s(1-\cos(2\theta_s))}{2} & -\frac{\sin(2\theta_s)}{2} \\ \frac{\sin(2\theta_s)}{2} & \frac{1-\cos(2\theta_s)}{2g_s} & \cos^2(\theta_s) & \frac{\sin(2\theta_s)}{2g_s} \\ \frac{g_s(\cos(2\theta_s)-1)}{2} & \frac{\sin(2\theta_s)}{2} & -\frac{g_s \sin(2\theta_s)}{2} & \cos^2(\theta_s) \end{pmatrix} \quad (1.132)$$

where  $\theta_s$  is

$$\theta_s = \frac{eB_0 l}{2p_0}. \quad (1.133)$$

One method to address these linear couplings is Edward-Teng parameterization [10]. The idea is that we can find a symplectic rotation that decouples the transverse coordinates, allowing them to be treated with the C-S formalism. Using block matrices, this is depicted as

$$T = \begin{pmatrix} M & n \\ m & N \end{pmatrix} \quad (1.134)$$

$$P = R^{-1}MR = \begin{pmatrix} E & 0 \\ 0 & F \end{pmatrix} \quad (1.135)$$

where  $T$  is our rotation matrix  $E$  and  $F$  have the form of Equation 1.82. The matrix  $R$  will have the form

$$R = \begin{pmatrix} I \cos \phi & D^{-1} \sin \phi \\ -D \sin \phi & I \cos \phi \end{pmatrix} \quad (1.136)$$

and  $D$  is a 2-by-2 matrix with determinant 1. As done in his original paper, the solutions then become

$$\cos \mu_1 - \cos \mu_2 = \frac{1}{2} \text{Tr}(M - N) \left( 1 + \frac{2 \det(m) + \text{Tr}(nm)}{(\frac{1}{2} \text{Tr}(M - N))^2} \right)^{1/2} \quad (1.137)$$

$$\cos 2\phi = \frac{\frac{1}{2} \text{Tr}(M - N)}{\cos \mu_1 - \cos \mu_2} \quad (1.138)$$

$$D = -\frac{m + \Omega n^T \Omega^T}{(\cos \mu_1 - \cos \mu_2) \sin 2\phi} \quad (1.139)$$

where the assumptions made are that  $-\frac{\pi}{4} \leq \phi \leq \frac{\pi}{4}$  and the sign of  $\sin 2\phi$  is determined by making  $\text{Tr}D > 0$ .  $\Omega$  and the derivation of this solution comes from the symplectic condition, which will be discussed in Section 1.9. This parameterization means that for 4-D linear transverse dynamics we can find coordinates where the motion is uncoupled and therefore analogous to two independent 2-D linear systems.

## 1.9 Symplectic Condition

Since we are looking at Hamiltonian systems we can expect our lattice to follow the symplectic condition:

$$\Omega = M \Omega M^T \quad (1.140)$$

where  $\Omega$  is the non-singular skew-symmetric matrix

$$\Omega = \begin{pmatrix} \Omega_2 & 0 & \dots & 0 \\ 0 & \Omega_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \Omega_2 \end{pmatrix} \quad (1.141)$$

and  $\Omega_2$  is

$$\Omega_2 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad (1.142)$$

The properties of the  $\Omega$  matrix are as follows:

$$\det(\Omega) = 1 \quad (1.143)$$

$$\Omega^{-1} = -\Omega \Rightarrow \Omega^2 = -I \quad (1.144)$$

$$\Omega^T = -\Omega \quad (1.145)$$

We can show that Hamiltonian systems follow this by starting with the equations of motion.

$$\begin{aligned} \frac{dx_i}{ds} &= \frac{\partial H}{\partial p_i} \\ \frac{dp_i}{ds} &= -\frac{\partial H}{\partial x_i} \end{aligned} \quad (1.146)$$

Now if we define  $M$  to be our transfer matrix from  $s_1$  to  $s_2$ :

$$\vec{X}(s_2) = M\vec{X}(s_1) \quad (1.147)$$

where  $\vec{X}(s_i)$  are the phase space coordinates at  $s_i$ . Since this matrix form is for a linear system the elements of  $M$  are defined by

$$M_{ij} = \frac{\partial X_i(s_2)}{\partial X_j(s_1)} \quad (1.148)$$

so

$$\begin{aligned} \frac{dX_i(s_2)}{ds} &= \frac{\partial X_i(s_2)}{\partial X_j(s_1)} \frac{dX_j(s_1)}{ds} \\ &= M_{ij} \Omega_{jk} \frac{\partial H}{\partial X_k(s_1)} \\ &= M_{ij} \Omega_{jk} \frac{\partial H}{\partial X_l(s_2)} \frac{\partial X_l(s_2)}{\partial X_k(s_1)} \\ &= M_{ij} \Omega_{jk} M_{kl}^T \frac{\partial H}{\partial X_l(s_2)} \end{aligned} \quad (1.149)$$

We can rewrite Equation 1.146 in matrix form with the phase space vector  $\vec{X}$  as

$$\frac{d}{ds}\vec{X} \equiv \Omega \frac{\partial H}{\partial \vec{X}} = \begin{pmatrix} \Omega_2 & 0 & \cdots & 0 \\ 0 & \Omega_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \Omega_2 \end{pmatrix} \frac{\partial H}{\partial \vec{X}} \quad (1.150)$$

Combining Equations 1.149 and 1.150 we get our original symplectic condition of Equation 1.140. From the symplectic condition we will know the following properties of the symplectic transfer matrix  $M$ .

If  $M$  is symplectic then so is  $M^T$  and  $M^{-1}$ .

If  $M$  and  $N$  are symplectic then so is  $MN$

$\det(M) = 1$

We can then look at the general form for a "one turn map" shown in Equation 1.82, this is because the data and models we will be covering in this thesis involving analyzing some one turn map for the lattice or beam position monitoring data which follows some unknown one turn map. This is only for a 2-D phase space but if we look at a general 4-D transfer matrix in block matrix form:

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad (1.151)$$

Since  $M$  will be a symplectic we know several constraints. These are:

$$\det A = \det D \equiv a \quad (1.152)$$

$$\det B = \det C \equiv b \quad (1.153)$$

$$a + b = 1 \quad (1.154)$$

This means that each individual matrix elements are not independent of one another which reduces the number of free parameters in our system. Now that we have covered the mathematical properties of a symplectic matrix we can show what these properties mean for our physical system.

One physical property is Liouville's theorem. This theorem states that if we have a collection of particles in phase space bounded by some volume  $V$  in phase space, then that volume will be preserved as the system advances in time. We can prove this by assuming a collection of particles that follow some Hamiltonian  $H(\vec{X}, t)$ . The volume,  $V_0$ , bounding the particles in the region  $R_0$  is

$$V_0 = \int_{R_0} d\vec{X}_0 \quad (1.155)$$

After some time  $t$ , the new region is  $R_1$  and the new volume is defined by

$$V_1 = \int_{R_1} d\vec{X}_1 \quad (1.156)$$

$d\vec{X}_1$  and  $d\vec{X}_0$  are related by a symplectic transformation.

$$V_1 = \int_{R_1} d\vec{X}_1 \quad (1.157)$$

$$= \int_{R_0} \left| \frac{\partial \vec{X}_1}{\partial \vec{X}_0} \right| d\vec{X}_0 \quad (1.158)$$

$$= \int_{R_0} \left| \det \left( \frac{\partial \vec{X}_1}{\partial \vec{X}_0} \right) \right| d\vec{X}_0 \quad (1.159)$$

$$= \int_{R_0} d\vec{X}_0 = V_0 \quad (1.160)$$

Where  $\frac{\partial \vec{X}_1}{\partial \vec{X}_0}$  is the Jacobian of the Hamiltonian, which is symplectic and therefore has a determinant of 1. From this, we know that the density of particles in the phase space does not change as they pass through the accelerator. This idea is known as Liouville's theorem, and it directly follows from the symplectic condition.

## CHAPTER 2

### SINGLE PARTICLE NONLINEAR DYNAMICS

#### 2.1 Intro

Our Hamiltonian will not generally lead to linear equations of motion. An important approach to problems like these is perturbation theory [11]. With this idea, we begin by representing our Hamiltonian  $H$  as a linear Hamiltonian  $H_0$  plus some small perturbative nonlinear term  $H_1$ .

$$H(p, q, s) = H_0(q, p, s) + \epsilon H_1(q, p, s) \quad (2.1)$$

where  $\epsilon \ll 1$ . We then use a generating function  $S(q, J, t)$  that is a canonical transformation for  $(q, p)$  to our action-angle  $(J, \nu)$  which are constant in the linear Hamiltonian and  $S$  satisfies

$$H_0\left(q, \frac{\partial S}{\partial q}, s\right) + \frac{\partial S}{\partial s} = 0. \quad (2.2)$$

The new Hamiltonian is then

$$\tilde{H} = H_0 + \epsilon H_1 + \frac{\partial S}{\partial s} = \epsilon H_1(J, \nu, s). \quad (2.3)$$

now the equations of motion are

$$\dot{J}_i = -\frac{\partial(\epsilon H_1)}{\partial \nu_i} \quad (2.4)$$

$$\dot{\nu}_i = \frac{\partial(\epsilon H_1)}{\partial J_i} \quad (2.5)$$

then we perform a Taylor series on the new variables around  $\epsilon$  of the form

$$J_i(J, \nu) = J_i^{(0)} + \epsilon J_i^{(1)} + \epsilon^2 J_i^{(2)} + O(\epsilon^3) \quad (2.6)$$

$$\nu_i(J, \nu) = \nu_i^{(0)} + \epsilon \nu_i^{(1)} + \epsilon^2 \nu_i^{(2)} + O(\epsilon^3). \quad (2.7)$$

We can then solve the higher order LHS with the lower order solution on the RHS i.e.

$$j_i^{(n)} = -\frac{\partial(\epsilon H_1)}{\partial v_i} \Big|_{(n-1)} \quad (2.8)$$

$$\dot{v}_i^{(n)} = \frac{\partial(\epsilon H_1)}{\partial J_i} \Big|_{(n-1)} \quad (2.9)$$

where evaluated at  $n$  means evaluating the term with the  $J_i^{(n)}$  and  $v_i^{(n)}$  terms. The higher-order corrections of  $v$  will give us our new frequencies.

This can be a slow way to calculate these properties, and we will frequently encounter small denominators during the order-by-order calculations due to resonance frequency terms. It is therefore helpful to consider other approaches to finding properties in dynamical systems, such as frequency.

## 2.2 Poincare Section

Particle beams are made up of many particles and pass through circular machines millions of times. In order to understand their motion, it can be useful to simplify the problem and only look at single particles as they move through the machine, ignoring intra-beam effects like space charge. At a specific point in the machine, a particle in one position  $\vec{X}_i$  will be in another position  $\vec{X}_{i+1}$  after one pass or "turn" through the machine. Rather than describing and analyzing the motion of the particle at every point, we can gain sufficient information about the system by using a Poincaré section [12]. These can be thought of as placing a sheet perpendicular to the motion of the particles at a location in the accelerator. Every time the particle passes through this sheet, we mark where it intersects. By plotting these intersections over many turns, we can extract a lot of information about the dynamical system. Figure 2.1 shows an illustration of this idea from [13]. Doing this for a linear accelerator lattice with a single sextupole kick results in Figure 2.2, which is called the Hénon Map [14].

By using this system we can visually see structures in the phase space that come from the nonlinearities of the system. For example, the five islands are from a resonance of the tune being close to 0.2. They are formed by stable fixed points in the middle of the islands and unstable fixed points between them.



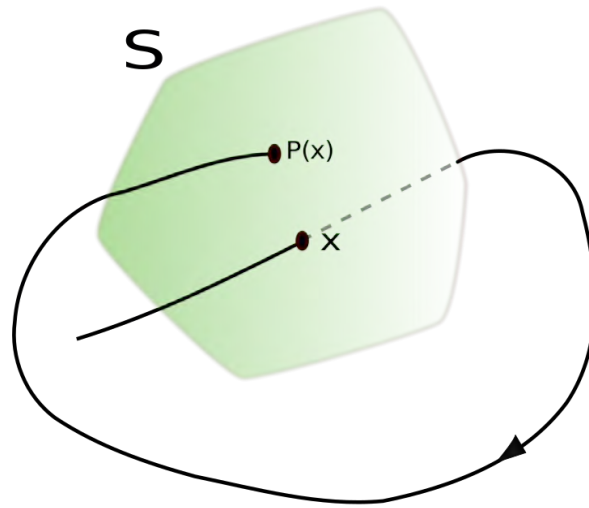


Figure 2.1 A visualization of a particle at  $x$  passing through a map and returning to the Poincare surface  $S$  where its new location is  $P(x)$  [13].

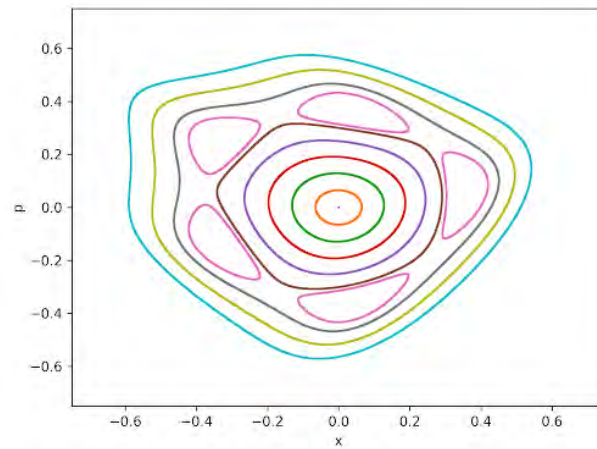


Figure 2.2 A Poincare map of a linear lattice with a single sextupole kick (Henon Map). Insert parameters

Our aim is to gain long-term stability of the particles in the system from these maps, i.e. the dynamic aperture. As stated in the introduction, the most accurate way to gain this long-term information is from tracking. In the context of Poincaré maps, this means finding an equation that relates one point to the next. But as we also discussed, this tracking can be impractical, so we must use other tactics to garner this information. One widely used method is frequency map analysis (FMA) [15]–[18]. To discuss how this method works it is useful to introduce the Kolmogorov–Arnold–Moser (KAM) Theorem [7] which we will discuss in section 2.5.

### 2.3 Lie Operators

In Chapter 1.1, we introduced our normalized coordinates to describe the position of a singular particle in phase space. We also introduced the matrix notation for describing the different elements of an accelerator,  $M_i$ , ( $i = [1, n]$ ), and combining them into a one-turn map  $M$  by

$$M = M_m M_{m-1} \dots M_2 M_1 \quad (2.10)$$

where  $m$  is the number of elements included in the map. If  $\vec{X} = (q_1, p_1, \dots, q_n, p_n)$  is a  $2n$ -D vector where  $n$  is the number of spatial dimensions in our system and  $(q_i, p_i)$  are conjugate position and momentum pairs, then  $M$  describes how these coordinates change after one pass through the accelerator map.

$$X_{j+1}^{\vec{}} = M X_j^{\vec{}} \quad (2.11)$$

where  $j$  is the turn number. However, this matrix notation can only describe a linear transformation. In general, particle accelerators are very nonlinear systems and we therefore need a more general nonlinear form for  $M$ . Lie Operators are one way we can describe these maps [19].

If we have two arbitrary functions  $f$  and  $g$  based on our canonical coordinates and independent variable  $s$  i.e.  $f(\vec{X}, s)$  and  $g(\vec{X}, s)$ , then the Poisson brackets are defined as

$$[f, g] = \sum_{i=1}^n \left( \frac{\partial f}{\partial q_i} \frac{\partial g}{\partial p_i} - \frac{\partial f}{\partial p_i} \frac{\partial g}{\partial q_i} \right) \quad (2.12)$$

The Lie operator is a compact notation for Poisson brackets defined as

$$: f : g = [f, g] \quad (2.13)$$

Using our Lie notation we can show several properties of the Poisson bracket.  $h$  is also a function of our canonical coordinates and  $s$  and  $a$  and  $b$  are scalars and constant.

$$:f:g = -:g:f \quad (2.14)$$

$$:af + bg:h = a:f:h + b:g:h \quad (2.15)$$

$$:f:(gh) = (:f:g)h + g(:f:h) \quad (2.16)$$

$$:f:(:g:h) + :g:(:h:f) + :h:(:f:g) = 0 \quad (2.17)$$

where equation 2.17 is called the Jacobian identity. We can define the effect an exponential effects on the operator as

$$(:f:)^k(gh) = \sum_{m=0}^k \frac{k!}{m!(k-m)!} [(:f:)^m g] [(:f:)^{k-m} h] \quad (2.18)$$

where  $(:f:)^2 g = [f, [f, g]]$  for example. We can also define the commutator of two operators as

$$\{ :f:, :g: \} = :f::g: - :g::f: \quad (2.19)$$

which using Equation 2.17 we can show:

$$\{ :f:, :g: \} = :[f, g]: \quad (2.20)$$

What will be shown to be a particularly useful case of the Lie operator is it's use in an exponential function. We extend a Taylor series expansion of  $e^x$  to obtain

$$e^{:f:} = \sum_{k=0}^{\infty} \frac{1}{k!} (:f:)^k \quad (2.21)$$

One useful property of the Lie operator is our ability to take derivatives in Hamiltonian systems. Again we have a function  $f(\vec{X}, s)$

$$\frac{df}{ds} = \frac{\partial f}{\partial s} + \frac{\partial f}{\partial \vec{X}} \frac{d\vec{X}}{ds} \quad (2.22)$$

Using Equation 1.150, Equation 2.22 becomes

$$\begin{aligned}\frac{df}{ds} &= \frac{\partial f}{\partial s} + \frac{\partial f}{\partial \vec{X}} \Omega \frac{\partial H}{\partial \vec{X}} \\ &= \frac{\partial f}{\partial s} + :f:H\end{aligned}\tag{2.23}$$

where  $\Omega$  is the same matrix we introduced in the symplectic section of chapter 1. So we can know if a function  $f(\vec{X}, s)$  is constant if it is not explicitly dependent on  $s$  and it commutes with the Hamiltonian, i.e.,  $:f:H = 0$  and  $\frac{\partial f}{\partial s} = 0$ .

There is also a general formula that we can use for the dynamics in the accelerator with the Lie operator. Starting with the derivative of our canonical coordinates,  $\vec{X}$ , with respect to  $s$  in an  $n$ -D Hamiltonian system

$$\begin{aligned}\frac{d\vec{X}}{ds} &= \sum_{i=0}^n \left( \frac{dp_i}{ds} \frac{\partial \vec{X}}{\partial p_i} + \frac{dq_i}{ds} \frac{\partial \vec{X}}{\partial q_i} \right) \\ &= - \sum_{i=0}^n \left( \frac{\partial H}{\partial q_i} \frac{\partial \vec{X}}{\partial p_i} - \frac{\partial H}{\partial p_i} \frac{\partial \vec{X}}{\partial q_i} \right) \\ &= -:H:\vec{X}\end{aligned}\tag{2.24}$$

One can perform this operation again and see that

$$\frac{d^k \vec{X}}{ds^k} = (-:H:)^k \vec{X}\tag{2.25}$$

We can then incorporate Equations 2.24 and 2.25 into a Taylor series of  $\vec{X}$

$$\begin{aligned}\vec{X}(s) &= \sum_{k=0}^{\infty} \frac{s^k}{k!} \left. \frac{d^k \vec{X}}{ds^k} \right|_{s=0} \\ &= \sum_{k=0}^{\infty} \frac{s^k}{k!} (-:H:)^k \vec{X}(0) \\ &= e^{-:sH:} \vec{X}(0)\end{aligned}\tag{2.26}$$

So for each element of our lattice we have

$$\vec{X}_f = e^{-:LH:} \vec{X}_0\tag{2.27}$$

where  $L$  is the length of the element.

Each element of the accelerator in this form can be combined together to form a one turn map for the entire ring. If  $e^{-:L_i H_i:}$  is the map for element  $i$  then all elements together are

$$e^{-:L_1 H_1:} e^{-:L_2 H_2:} e^{-:L_3 H_3:} \dots e^{-:L_N H_N:} = \prod_{i=1}^N e^{-:L_i H_i:} := e^{-:CH_{\text{eff}}:} \quad (2.28)$$

where  $C$  is the circumference of the ring.

The calculation for  $H_{\text{eff}}$  is not a trivial addition as the operators do not necessarily commute. For this calculation, one needs the Baker-Campbell-Hausdorf (BCH) formula [19]:

$$e^{:f:} e^{:g:} = e^{:h:} \quad (2.29)$$

where  $h$  is

$$\begin{aligned} h = & f + g + \frac{1}{2}:f:g + \frac{1}{12}:f:^2g + \frac{1}{12}:g:^2f + \frac{1}{24}:f::g:^2f - \frac{1}{720}:g:^4f - \frac{1}{720}:f:^4g \\ & + \frac{1}{360}:g::f:^3g + \frac{1}{360}:f::g:^3f + \frac{1}{120}:f:^2:g:^2f + \frac{1}{120}:g:^2:f:^2g + O((f, g)^6) \end{aligned} \quad (2.30)$$

If  $f$  and  $g$  commute or equal a constant that is only dependent on  $s$ , then the higher order terms will vanish.

## 2.4 Normal Form

With Lie operators, we can perform what is called Normal Form [19]. We seek a canonical transformation that simplifies the motion of our system to be as close to a pure rotation as possible. In linear systems, this is the Courant-Snyder transformation we discussed. If  $\vec{X}$  are our original phase space and our one-turn map is the matrix  $M$  then

$$\vec{X}_f = M \vec{X}_0 \quad (2.31)$$

Using the Courant-Snyder parameters, we transform the coordinates  $\vec{X}$  to  $U$  via the matrix  $A_2$

$$\vec{U} = A_2^{-1} \vec{X} \quad (2.32)$$

such that Equation 2.31 takes the form

$$\begin{aligned} \vec{X}_f &= A_2 R A_2^{-1} \vec{X}_0 \\ A_2^{-1} \vec{X}_f &= R A_2^{-1} \vec{X}_0 \\ \vec{U}_f &= R \vec{U}_0 \end{aligned} \quad (2.33)$$

where  $R$  is the rotation matrix. For 2-D phase spaces,  $A_2$  is the matrix we found in Equation 1.83. So for  $k$  number of turns Equation 2.31 becomes

$$\vec{U}_k = R^k \vec{U}_0 \quad (2.34)$$

The idea of the C-S formalism is to simplify the phase space to one where the dynamics are a simple rotation. Normal form aims to generalize this idea for nonlinear dynamics by simplifying these dynamics order by order. So, if we can generalize this transformation from  $A_2$  to  $A$  and from  $R$  to  $N$  then we can study the much simpler dynamics of  $N$  instead of  $M$  to gain the same information.

Many normal form methods are perturbative, so we will show an example of a third-order system. Consider a map  $M$  that has the form

$$M = e^{i f_2} e^{i f_3} \quad (2.35)$$

Here,  $f_2$  is a second-order Hamiltonian that is linearized by the C-S transformation  $A_2$ . Let us consider the transformation.

$$A = e^{i F_3} A_2 \quad (2.36)$$

where  $F_3$  is the transformation for which we solve. Then our simplified transformation  $N$  is as follows.

$$\begin{aligned} N &= A M A^{-1} = e^{i F_3} A_2 e^{i f_2(\vec{X})} e^{i f_3(\vec{X})} A_2^{-1} e^{-i F_3(\vec{X})} \Big|_{\vec{X}=\vec{U}} \\ &= e^{i F_3(\vec{U})} e^{i f_2(A_2 \vec{U})} e^{i f_3(A_2 \vec{U})} e^{-i F_3(\vec{U})} \end{aligned} \quad (2.37)$$

It can be shown that

$$e^{i f} F(\cdot; g) e^{-i f} = F(\cdot; e^{i f} g) \quad (2.38)$$

which means when we use the BCH formula (Equations 2.29 and 2.30) and expand to the second order, Equation 2.37 becomes

$$\begin{aligned} N &= e^{:f_2(A_2\vec{U}):} e^{-:f_2(A_2\vec{U}):} e^{:F_3(\vec{U}):} e^{:f_2(A_2\vec{U}):} e^{:f_3(A_2\vec{U}):} e^{-:F_3(\vec{U}):} \\ &= e^{:f_2(A_2\vec{U}):} \exp[:(e^{:f_3(A_2\vec{U}):} - 1)F_3(\vec{U}) + f_3(A_2\vec{U}):] + :O(\vec{U}^3): \end{aligned} \quad (2.39)$$

We define a new basis from the eigenmodes of  $:f_2:$  as

$$\begin{aligned} |abcd, e\rangle &\equiv (\sqrt{A_x}e^{i\phi_x})^a (\sqrt{A_x}e^{-i\phi_x})^b (\sqrt{A_y}e^{i\phi_y})^c (\sqrt{A_y}e^{-i\phi_y})^d \delta^e \\ &= A_x^{(a+b)/2} A_y^{(c+d)/2} e^{i(a-b)\phi_x} e^{i(c-d)\phi_y} \delta^e \end{aligned} \quad (2.40)$$

where

$$:f_2: |abcd, e\rangle = i[(a-b)\mu_x + (c-d)\mu_y] |abcd, e\rangle \quad (2.41)$$

where  $\mu_{x/y}$  are the frequencies of the normal modes.

It is then possible to write a general  $:f_n:$  in the form

$$\begin{aligned} f_n &= \sum_{n=0}^{\infty} C_{abcd,e}^{(n)} |abcd, e\rangle \\ &\begin{cases} a, b, c, d, e = 0 \\ a + b + c + d + e = n \end{cases} \end{aligned} \quad (2.42)$$

If we are away from resonance, then we can choose  $F_3$  of the form

$$F_3 = \left( \frac{1}{1 - e^{-:f_2:}} \right) f_3 \quad (2.43)$$

which will transform our map  $N$  into just  $e^{:f_2:}$  for up to the third order. The reason for needing to be away from resonances is more clear if we use equation 2.42 to rewrite 2.43.

$$\begin{aligned}
F_3 &= \left( \frac{1}{1 - e^{-i f_2}} \right) \sum_{a,b,c,d,e=0}^3 C_{abcd,e}^{(3)} |abcd, e\rangle \\
&= \sum_{a,b,c,d,e=0}^3 C_{abcd,e}^{(3)} \left( \frac{1}{1 - e^{-i[(a-b)\mu_x + (c-d)\mu_y]}} \right) |abcd, e\rangle
\end{aligned} \tag{2.44}$$

Notice that from the denominator, we would have an issue at resonance when

$$(a - b) \frac{\mu_x}{2\pi} + (c - d) \frac{\mu_y}{2\pi} = p \tag{2.45}$$

where  $p$  is an integer or when  $a = b$  and  $c = d$ .

## 2.5 Kolmogorov–Arnold–Moser (KAM) Theorem

In general, the dynamical systems from accelerator lattices are not integrable. However, they are sometimes very close to an integrable system. So we can characterize their Hamiltonian as an integrable one plus some small perturbation. Under no perturbation, these solutions will follow the surface of an  $n$  dimensional torus, where  $n$  is the number of degrees of freedom. KAM theory discusses what happens with these tori when we perturb the Hamiltonian [7].

When the perturbations are added, many tori are destroyed and what is left are regions of chaotic motion between the surviving tori. All the surviving tori, and some that do not survive, meet the so called Diophantine Condition [7]

$$|\langle k, \omega \rangle| \geq \frac{\alpha}{|k|^\tau} \text{ for all } 0 \neq k \in \mathbb{Z} \tag{2.46}$$

This condition suggest that for a tori to survive the perturbation. It needs a sufficiently irrational frequency,  $\omega$ .

## 2.6 Numerical Analysis of Fundamental Frequencies (NAFF)

The originally used to find chaotic motion in planetary orbits, the Numerical Analysis for Fundamental Frequencies (NAFF) algorithm has been applied to study chaotic motion in particle accelerators [15]–[18]. This algorithm is for finding the primary frequencies of quasi-periodic orbits whose frequencies met the Diophantine condition.



We begin with our quasi-periodic solution

$$f(t) = e^{i\nu_1 t} + \sum_{k \in \mathbb{Z}^n - (1,0,0,\dots,0)} a_k e^{i\langle k, \nu \rangle t}; a_k \in \mathbb{C} \quad (2.47)$$

where  $a_k$  are complex amplitudes and  $\nu$  is the frequency of the orbit which satisfies the Diophantine condition in Equation 2.46. The values of  $f(t)$  are sampled over a time interval  $[T_1, T_2]$  with a small enough time step,  $h$ , that accurate integrals can be computed. The one turn of a particle accelerator should be a sufficiently small time step. The goal is to find an approximate analytical form of  $f(t)$  which we will call  $f'(t)$ . To make the data more symmetric with no loss of generality we will perform the following change of variables for time

$$T = \frac{1}{2}(T_2 - T_1) \quad (2.48)$$

$$t' = t - \frac{1}{2}(T_1 + T_2) \quad (2.49)$$

so our time goes from  $[-T, T]$ . The approximate solution found by NAFF will have the form

$$f'(t) = \sum_{k=1}^N a'_k e^{i\omega'_k t} \quad (2.50)$$

where each subsequent  $a'_k$  is progressively smaller. The first frequency  $\omega'_1$  is found by finding the maximum of

$$\phi(\sigma) = \langle f(t), e^{i\sigma t} \rangle \quad (2.51)$$

where

$$\langle f(t), g(t) \rangle = \frac{1}{2T} \int_{-T}^T f(t) g^*(t) \chi(t/T) dt \quad (2.52)$$

and  $\chi(t)$  is a weight function with the property

$$\frac{1}{2} \int_{-1}^1 \chi(t) dt = 1 \quad (2.53)$$

Once  $\omega'_1$  is found,  $a'_1$  is found by an orthogonal projection. Then the frequency is removed from the data by

$$f_1(t) = f(t) - a'_1 e^{i\omega'_1 t} \quad (2.54)$$

and the process is done again to find the subsequent frequencies and amplitudes. As discussed by Laskar in his introduction to frequency maps, the accuracy of the approximated frequency  $\nu'_1$  converges to the actual frequency  $\nu_1$  as

$$|\nu'_1 - \nu_1| \sim O(1/T^{2p+2}) \quad (2.55)$$

when using the weight function

$$\chi_p(t) = \frac{2^p (p!)^2}{(2p)!} (1 + \cos(\pi t))^p \quad (2.56)$$

[17]. This algorithm results in faster convergence than a standard Fourier transformation which will only be accurate to about  $\pi/T$  [16]. With this algorithm we can perform frequency map analysis to analyze the motion from Poincare maps.

## 2.7 Frequency Map Analysis

As we discussed in section 2.5, particles with sufficiently irrational frequencies that survive nonlinear perturbations will trace an  $n$ -dimensional torus in phase space. The motion is quasi-periodic because it will not cross the same point but approach arbitrarily close. The torus is traced by some action being rotated at some frequency; this is why it will trace an  $n$ -dimensional torus in a  $2n$ -dimensional phase space. Section 2.6 discussed one common approach to finding these frequencies. We can expect particles in the dynamic aperture will be on these surviving tori. If they are on the surviving tori, then we can expect their frequencies to be stable.

Therefore, the frequency map analysis technique starts by collecting some particle tracking data  $(\vec{x}_0, \vec{x}_1, \vec{x}_2, \dots, \vec{x}_N)$ , either from simulation or measurements.  $N$  is the number of turns corresponding to  $N + 1$  data points. We split these data points into two groups,  $(\vec{x}_0, \vec{x}_1, \dots, \vec{x}_{N/2})$

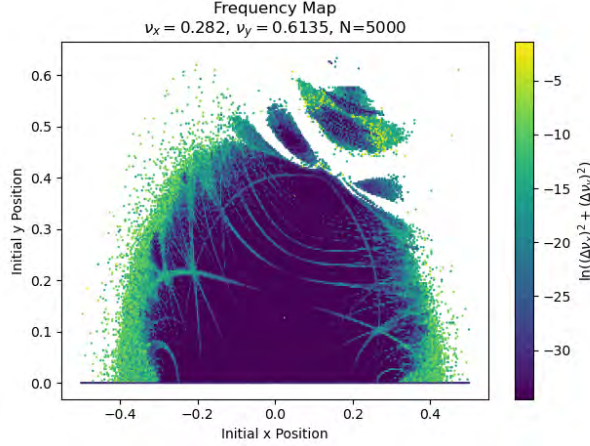


Figure 2.3 4-D Hénon Map frequency map for  $v_x = 0.282$  and  $v_y = 0.6135$  and 5000 turns [20].

and  $(\vec{x}_{N/2+1}, \vec{x}_{N/2+2}, \dots, \vec{x}_N)$ . We measure the frequencies collected from NAFF for each group which we will call  $\vec{v}_1$  and  $\vec{v}_2$  respectively. The measure for our stability is then

$$|\vec{v}_2 - \vec{v}_1|. \quad (2.57)$$

If the frequencies for both sets agree well, then we say it is likely that the particles are stable. Otherwise, this would indicate the motion is chaotic, and it is possible the particles could be lost. It is not a guarantee, however, that they will be lost. Particle motion cannot cross the tori of different particles, or else they would be on the same trajectory. So the stable tori provide boundaries for the regions of chaotic motion. If the phase space is larger than 2 dimensions, then these  $n$ -dimensional tori cannot create a bounded area, which means there is no region where a particle's motion is cut off from any other part of the phase space. So it is possible that a particle's motion can be relatively bounded for a number of turns but eventually go to infinity. It is also possible that a particle with unstable frequencies can stay in a relatively small region for a long period of time and therefore not be lost in the accelerator.

Figure 2.3 shows an example of the 4-D Hénon Map frequency map. The figure shows the initial condition of particles colored by the measure in Equation 2.57. The lines correspond to resonances, a known source of instability in accelerators [18].

## CHAPTER 3

### SQUARE MATRIX APPROACH

#### 3.1 Square Matrix Method

As we try to move away from computationally intensive turn-by-turn tracking, many Normal Form methods are perturbative or based on power series of the map. Another way to approach these power series is by extending our phase space. The Square Matrix Approach, developed by Li-Hua Yu [21], is an approach that avoids order-by-order calculations, unlike the normal form discussed in Chapter 2.

In this extended phase space, we are trading our  $2n$ -dimensional phase space with a nonlinear map for an infinite-dimensional phase space with a linear map. To perform calculations, we need to truncate this map to some order. But afterwards, we can use well-established linear matrix techniques to analyze the system. For example, a 2-D system with phase space coordinates  $(x, p)$  could be extended to 4th order, creating an extended 14-D phase space where the extended phase space vector is of the form

$$\vec{X} = (x, p, x^2, xp, p^2, \dots, p^4) \quad (3.1)$$

and the transformation is a linear one of the form

$$\vec{X}_f = M\vec{X}_o \quad (3.2)$$

where  $M$  is the square matrix with real coordinates, we define a complex coordinate  $z$  such that it is the coordinate for our diagonalized linear matrix from the Courant-Snyder parameters. If we truncate to the third order, our expanded phase space where the square matrix  $M_s$  exists will act on the vector  $Z$  of the form

$$Z = (1, z, z^*, z^2, zz^*, z^{*2}, z^3, z^2 z^*, zz^{*2}, z^{*3}) \quad (3.3)$$

or for 2 spatial dimensions

$$\begin{aligned}
Z = & (1, z_x, z_x^*, z_y, z_y^*, z_x^2, z_x z_x^*, (z_x^*)^2, z_x z_y, z_x^* z_y, z_y^2, z_x z_y^*, z_x^* z_y^*, z_y z_y^*, (z_y^*)^2, \\
& z_x^3, z_x^2 z_x^*, z_x (z_x^*)^2, (z_x^*)^3, z_x^2 z_y, z_x z_x^* z_y, (z_x^*)^2 z_y, z_x z_y^2, z_x^* z_y^2, z_y^3, z_x^2 z_y^*, \\
& z_x z_x^* z_y^*, (z_x^*)^2 z_y^*, z_x z_y z_y^*, z_x^* z_y z_y^*, z_y^2 z_y^*, z_x (z_y^*)^2, z_x^* (z_y^*)^2, z_y (z_y^*)^2, (z_y^*)^3).
\end{aligned} \tag{3.4}$$

$$Z = M_s Z_0 \tag{3.5}$$

From Equation 3.5 we see that the mappings for the higher orders of  $z$  are calculated by simply taking the map of  $z$  in terms of  $z_0$  and raising it to that power. Therefore, there will be redundant information in the matrix  $M_s$ . It would be natural to try to diagonalize this matrix, but the matrix is not generally diagonalizable due to degenerate eigenvalues. However,  $M_s$  is guaranteed to be a square matrix, so it is possible to use Jordan decomposition to find generalized eigenvectors. While not diagonalizing the matrix, they can greatly simplify it.

Jordan decomposition is generally not numerically stable. However, the stability is dependent on the accuracy of the eigenvalues. Because  $M_s$  is not only square, but also upper triangular, we know that the eigenvalues will be terms on the diagonal. So, the sufficient accuracy of the eigenvalues can be assured.

By performing a Jordan decomposition the new form of  $M_s$  becomes  $U^{-1}NU$  where

$$N = \begin{pmatrix} N_0 & 0 & \cdots & \cdots & 0 \\ 0 & N_1 & \cdots & \cdots & 0 \\ 0 & 0 & N_{-1} & \cdots & 0 \\ 0 & 0 & 0 & N_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{3.6}$$

and

$$N_k = e^{2\pi i k \nu} I + \tau^\dagger. \tag{3.7}$$

$\tau$  is a matrix where the diagonal below the main diagonal of the matrix is one, and the rest of the elements are zero.

$$\tau = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (3.8)$$

From the Jordan decomposition, the rows of the transformation matrix  $U$  will be generalized eigenvectors from a set of different Jordan chains. Each Jordan chain is a set of vectors  $C$  that satisfy

$$C(x) = \{x, (M_s - \lambda I)x, (M_s - \lambda I)^2 x, \dots, (M_s - \lambda I)^{k-1} x\} \text{ s.t. } (M_s - \lambda I)^k x = 0 \quad (3.9)$$

where  $\lambda$  is an eigenvalue of the matrix. These generalized eigenvectors form a linearly independent basis for each subspace of  $M_s$ . We only need the information from the longest of these chains. The transformation matrix  $U$  satisfies

$$UM = e^{2\pi i \nu I + \tau} U. \quad (3.10)$$

We will then define the vector  $W$  such that

$$W = UZ \quad (3.11)$$

From KAM theory, we expect stable orbits to have a stable frequency. Using Equations 3.10 and 3.5 this approximation takes the form

$$UMZ_0 = UW = e^{2\pi i \nu I + \tau} W_0 \approx e^{i(2\pi \nu I + \phi)} W_0 \quad (3.12)$$

where  $\phi$  represents the amplitude-dependent tune shift. We see that this approximation gives the equation:

$$\tau W \approx i\phi W \quad (3.13)$$

The approximation becomes exact as our truncation order goes to infinity.  $\tau$  can be thought of as a raising operator on the orders of  $W$  so,

$$\tau W = \tau \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{m-1} \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ 0 \end{pmatrix} \approx \begin{pmatrix} i\phi w_0 \\ i\phi w_1 \\ \vdots \\ i\phi w_{m-1} \end{pmatrix} \quad (3.14)$$

where  $w_j$  are the rows of  $W$ . So we can relate the rows of  $W$  to  $\phi$  by

$$i\phi = \frac{w_1}{w_0} \approx \frac{w_2}{w_1} \approx \dots \approx \frac{w_{m-1}}{w_{m-2}} \quad (3.15)$$

The real part of  $\phi$  relates to the tune, and the imaginary part relates to amplitude changes over time. So for stable orbits, we expect

$$\text{Im}(\phi) \approx 0. \quad (3.16)$$

So, from the first Equation 3.15 we can define the stability indicator

$$\Delta = \frac{w_2}{w_0} - \left( \frac{w_1}{w_0} \right)^2 \approx 0. \quad (3.17)$$

The reason the first two terms of Equation 3.15 are used is that  $w_i$  are increasingly higher monomials of a small value as  $i$  increases, so the numerator and denominator will both be small and possibly inaccurate at the higher values of  $i$ .

An additional feature of this square matrix method is that it provides a function for the amplitude-dependent tune shift.

$$\phi(Z) = -i \frac{w_1(Z)}{w_2(Z)} \quad (3.18)$$

We can compare this to the form of the traditional amplitude dependence. If we have a linear Hamiltonian  $H_0$ , and we assume that our nonlinearity can be in the form of an additional Hamiltonian term  $H_1$ , then a perturbed system's Hamiltonian can be written as

$$H(\vec{x}) = H_0(\vec{x}) + H_1(\vec{x}). \quad (3.19)$$

We can rewrite the linear Hamiltonian in terms of action-angle variables, which means  $H_0$  will only depend on the action  $J$ .

$$H(J, \phi) = H_0(J) + H_1(J, \phi) = \nu J + H_1(J, \phi) \quad (3.20)$$

When we use the Hamiltonian equations of motion, we then find

$$\frac{\partial H}{\partial \phi}(J, \phi) = -\frac{\partial H_1}{\partial \phi}(J, \phi) \quad (3.21)$$

$$\frac{\partial H}{\partial J}(J, \phi) = \nu + \frac{\partial H_1}{\partial J}(J, \phi). \quad (3.22)$$

Where compared to a linear system, where there would be no dependence on  $\phi$  and a constant frequency  $\nu$ , the perturbative term  $H_1$  introduces a changing action called action smear (Equation 3.21) and an action-dependent tune shift (Equation 3.22). So we expect that the action-dependent tune shift in Equation 3.22 will approximately equal that in Equation 3.18.

### 3.1.1 2-D Hénon Map

We will use the Hénon map as an example to compare different methods. The Hénon map can have different parameters, but for our purpose, we will use the following Hamiltonian

$$H(x, p) = \frac{1}{2}x^2 + \frac{1}{2}p^2 + \frac{1}{3}Sx^3 \sum_n \delta(t - nT). \quad (3.23)$$

Equation 3.23 will give the discrete map



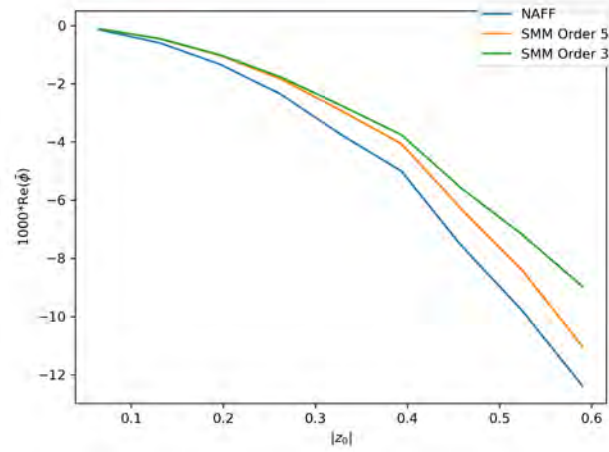


Figure 3.1 The amplitude-dependent tune shift for the 2-D Hénon map calculated from NAFF and Equation 3.18 for 3rd and 5th order.

$$x_{n+1} = x_n \cos \mu + (p_n - x_n^2) \sin \mu \quad (3.24)$$

$$p_{n+1} = -x_n \sin \mu + (p_n - x_n^2) \cos \mu. \quad (3.25)$$

Physically, this represents an accelerator that has a linear lattice with a periodic sextupole kick. This kick is where we will see the nonlinear effects, such as amplitude-dependent tune shift. In Equation 3.23,  $S$  is the strength of the sextupole kick and  $\mu$  is the frequency of the linear lattice. The  $(x, p)$  coordinates are scaled by the sextupole strength in Equation 3.25 for simplicity. Figure 2.2 shows the Poincaré section for this map in 2-D.

We can compare the results of Equation 3.18 to NAFF and see that it gives a close approximation to the amplitude-dependent tune shift without order-by-order calculations like in normal form. These results are in Figure 3.1. These calculations can be done with symbolic methods [22], but these will quickly become very intensive. Since we know the exact form of the eigenvalues from Equation 3.7, we can perform the Jordan decomposition fairly accurately using numerical methods. This allows us to use higher orders in the square matrix method. These additional orders are shown in Figure 3.2.

### 3.1.2 4-D Hénon Map

We can expand this 2-D method to apply to 4-D systems. The 4-D Hénon Map is as follows

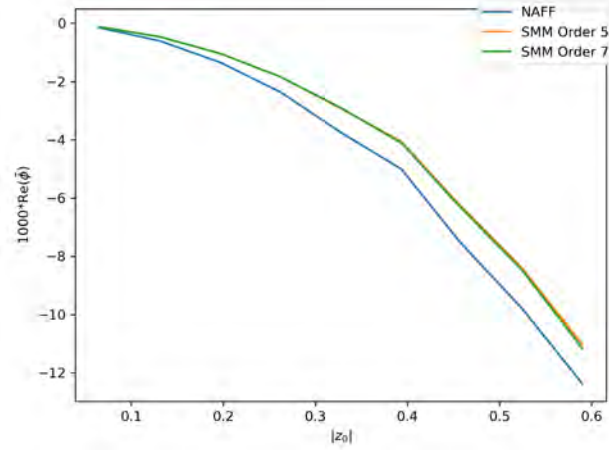


Figure 3.2 The amplitude-dependent tune shift for the 2-D Hénon map calculated from NAFF and Equation 3.18 for 5th and 7th order using numerical methods.

$$\begin{pmatrix} x \\ p_x \\ y \\ p_y \end{pmatrix}_{n+1} = \begin{pmatrix} \mathbf{R}(\mu_x) & 0 \\ 0 & \mathbf{R}(\mu_y) \end{pmatrix} \begin{pmatrix} x \\ p_x - x^2 + y^2 \\ y \\ p_y + 2xy \end{pmatrix}_n \quad (3.26)$$

where  $R(\theta)$  is a 2-D rotation by angle  $\theta$ . Physically, it represents the same accelerator structure as the 2-D version but includes both transverse directions, introducing nonlinear coupling into the system. Similar to the 2-D version, the 4-D Square Matrix will have the structure

$$N = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & N_{1,0} & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & N_{-1,0} & \cdots & \cdots & 0 \\ 0 & 0 & 0 & N_{0,1} & \cdots & 0 \\ 0 & 0 & 0 & 0 & N_{0,-1} & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (3.27)$$

and

$$N_{l,k} = e^{2\pi i(l\nu_x + k\nu_y)} I + \tau^\dagger. \quad (3.28)$$

By examining the chains that arise from the  $v_x$  and  $v_y$  terms, we find a Jordan chain for each. They each give us a transformation to  $w_x$  and  $w_y$ , respectively. Once we have the transformations, we can perform the same steps as the 2-D version for each and get  $\Delta_x$  and  $\Delta_y$  terms from Equation 3.17. We combine these terms to find the new delta term

$$\Delta = \Delta_x^2 + \Delta_y^2. \quad (3.29)$$

Looking at the exponent of Equation 3.29, we get the map in Figure 3.3 using the third-order square matrix method. We can see from Figures 3.1 and 3.2 that the prediction from the SMM deviates at larger amplitudes. This is expected from a perturbative method. Unfortunately, the most important amplitudes for calculating the dynamic aperture are those near the aperture boundary. So the SMM will need to be expanded upon for more accurate dynamic aperture calculations.

### 3.2 Iterative Method

The stable orbits in nonlinear systems like accelerators are bounded pseudo-periodic orbits,  $\left(z_{(x,y)}^{(0)}, z_{(x,y)}^{(1)}, \dots, z_{(x,y)}^{(n)}, \dots\right)$ . As we discussed in section 2.5, KAM theory showed that the invariant tori survive under small nonlinear perturbations, so we expect that there exists a diffeomorphism such that the motion will be a rigid rotation. That is, an orbit with a constant amplitude rotating by a constant angle in phase space. The 2-D version has been outlined in [23]. Here we will outline the approach for 4-D systems as was briefly discussed in [20], [24], this introduces coupling between the coordinates and the diffeomorphisms are then dependent on both rotation angles, as we will show.

First we express out coordinates  $z_{(x,y)}$  in terms of a complex phase:

$$z_x = e^{i\theta_x} \quad (3.30)$$

$$z_y = e^{i\theta_y}. \quad (3.31)$$

The real part of  $\theta$  is related to the angle in phase space while the imaginary part relates to the amplitude of the motion. We then define  $f_x$  and  $f_y$  as the functions of the changes in  $\theta_x$  and  $\theta_y$  after

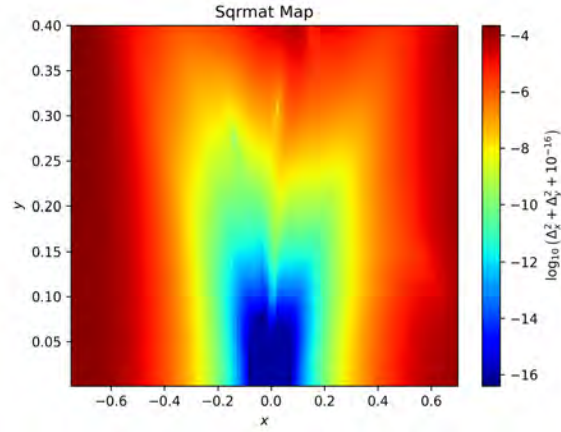


Figure 3.3 Stability map for the Hénon map using the third-order square matrix method to calculate  $\Delta$  from equation 3.29.

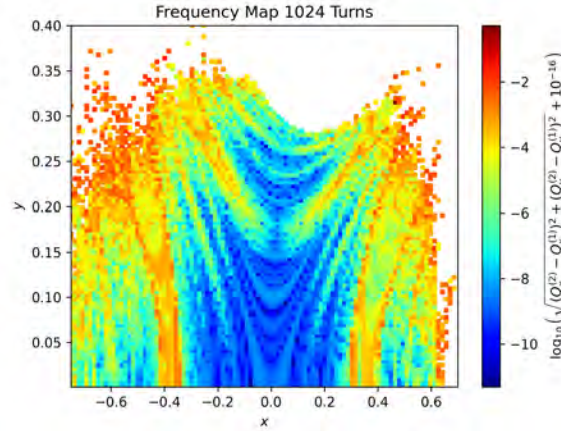


Figure 3.4 Frequency map for the Hénon map using NAFF.

one turn respectively. Both are dependent on  $(\theta_x, \theta_x^*, \theta_y, \theta_y^*)$ .

$$\frac{z'_{(x,y)}}{z_{(x,y)}} = \exp i(\theta'_{(x,y)} - \theta_{(x,y)}) = \exp i f_{(x,y)}(\theta_x, \theta_x^*, \theta_y, \theta_y^*) \quad (3.32)$$

The diffeomorphism to a rigid rotation that we expect to find when the motion is pseudo-periodic and bounded is

$$\theta_x = \alpha + h(\alpha, \beta) \quad (3.33)$$

$$\theta_y = \beta + g(\alpha, \beta). \quad (3.34)$$

where  $\beta$  and  $\alpha$  are the angles of pure rigid rotations in our new phase space coordinates.  $h$  and  $g$  are smooth complex functions, which are periodic with respect to  $\alpha$  and  $\beta$ , and each has a period of  $2\pi$ . Note that they are coupled and dependent on both angles.

$$\alpha_{n+1} = \alpha_n + \rho_x \quad (3.35)$$

$$\beta_{n+1} = \beta_n + \rho_y \quad (3.36)$$

here  $\rho_x$  and  $\rho_y$  are the rotation numbers, which are defined as:

$$\rho_{(x,y)} = \lim_{n \rightarrow \infty} \frac{\theta_{(x,y)}^{(n)} - \theta_{(x,y)}^{(0)}}{n} \quad (3.37)$$

Since  $h$  and  $g$  are periodic, it is useful to express them as their Fourier series.

$$h = \sum_{m,n=-\infty}^{\infty} \hat{h}_{n,m} e^{im\alpha + in\beta} \quad (3.38)$$

$$g = \sum_{m,n=-\infty}^{\infty} \hat{g}_{n,m} e^{im\alpha + in\beta} \quad (3.39)$$

From equation 3.32 we use an iterative method to solve for the diffeomorphisms and the rotation numbers starting from a given initial phase space coordinate. By combining equation 3.32 with equations 3.38 and 3.39 we can see how  $h$  and  $g$  change between turns. The difference between  $h$  and  $g$  from one turn to the next is  $f_x$  and  $f_y$  minus the rotation number  $\rho_x$  and  $\rho_y$ , respectively.

For simplicity, will define  $f_x$  and  $f_y$  minus their respective rotation numbers as  $\eta_x(\rho_x, \alpha, \beta, h(\alpha, \beta), g(\alpha, \beta))$  and  $\eta_y(\rho_y, \alpha, \beta, h(\alpha, \beta), g(\alpha, \beta))$ .

So for each iteration, we solve

$$h^{(n+1)}(\alpha + \rho_x^{(n+1)}, \beta + \rho_y^{(n+1)}) - h^{(n+1)}(\alpha, \beta) = \eta_x(\rho_x^{(n+1)}, \alpha, \beta, h^{(n)}(\alpha, \beta), g^{(n)}(\alpha, \beta)) \quad (3.40)$$

and

$$g^{(n+1)}(\alpha + \rho_x^{(n+1)}, \beta + \rho_y^{(n+1)}) - g^{(n+1)}(\alpha, \beta) = \eta_y(\rho_y^{(n+1)}, \alpha, \beta, h^{(n)}(\alpha, \beta), g^{(n)}(\alpha, \beta)). \quad (3.41)$$

Note that the  $(n)$  in the superscript denotes iteration number, not turn number. We begin the first iteration with

$$h^{(0)} = \theta_{xo} - \alpha_0 \quad (3.42)$$

$$g^{(0)} = \theta_{yo} - \beta_0. \quad (3.43)$$

Equations 3.42 and 3.43 mean we begin by assuming the motion is already a pure rotation. Each iteration will progressively correct this assumption. First, the zeroth order Fourier component of the left-hand side of Equations 3.40 and 3.41 is zero since the zeroth order components of  $g$  and  $h$  are constants and therefore do not change from one turn to the next. By substituting Equations 3.38 and 3.39 into Equations 3.40 and 3.41 we get the higher order Fourier terms of  $h$  and  $g$  have the form

$$\hat{h}_{n,m}^{(n+1)}(\alpha, \beta) = \frac{\hat{\eta}_{x,nm}^{(n)}}{e^{im\rho_x^{(n+1)} + in\rho_y^{(n+1)}} - 1} \quad (3.44)$$

$$\hat{g}_{n,m}^{(n+1)}(\alpha, \beta) = \frac{\hat{\eta}_{y,nm}^{(n)}}{e^{im\rho_x^{(n+1)} + in\rho_y^{(n+1)}} - 1} \quad (3.45)$$

One can see that the exponential in the denominator cannot equal 1, or else that component will go to infinity, and the method will fail to find a solution. This zero denominator is more difficult to land on in the 4-D version since it requires both the estimated rotation numbers to be on a resonance.

As stated before, the zeroth order terms on the left hand side of Equations 3.40 and 3.41 go to zero, but we still need to find what these zero terms ( $\hat{h}_{0,0}$  and  $\hat{g}_{0,0}$ ) are in order to get the complete diffeomorphism. This is done with the initial position in phase space. We constrain the diffeomorphism to pass through the initial angles. So we choose  $\hat{h}_{0,0}$  and  $\hat{g}_{0,0}$  such that for some existing  $\alpha^*$  and  $\beta^*$  we'll have:

$$\theta_{xo} = \alpha^* + h^{(n+1)}(\alpha^*, \beta^*) \quad (3.46)$$

$$\theta_{yo} = \beta^* + g^{(n+1)}(\alpha^*, \beta^*) \quad (3.47)$$

We continue this process until we converge on a solution. The solution converges to the tunes of  $x$  and  $y$ , as well as the KAM surface the particle orbits exist on.

There are a few important notes to make about implementing this method. For one, the function  $\eta_i$  needs to be sufficiently smooth for the method to converge. To aid this, we use the SMM transformation as our action-angle variables in the iteration method. There is another issue created by this, though. The inverse function from the SMM method is often not exact, which can lead to inaccuracies when it is called multiple times in the code. To avoid this, we implement Newton's method. If we have some number  $y$ , which is the results of some unknown number  $x$  being passed into the function  $f(x)$ , we can find  $x$  by defining a new function  $g(x)$  such that

$$g(x) = f(x) - y. \quad (3.48)$$

We can then use Newton's method to find the root of a function numerically, since the root of  $g(x)$  will give us the inverse of  $f(x)$ . This is implemented as follows

$$x^{i+1} = x^i - (J_f(x^i))^{-1} (f(x^i) - y) \quad (3.49)$$

where  $J_f$  is the Jacobian of  $f(x)$  and  $i$  refers to the  $i^{\text{th}}$  iteration of Newton's method.

This can introduce extra computation time, but as the results will show, it can greatly increase the area of convergence for some maps. Figure 3.7 shows that using Newton's method to improve the inverse results in a much more accurate inverse. We can also see how much using the SMM method improves the iteration method in Figure 3.8. It shows the 4-D Henon Map using the parameters  $\nu_x = 0.282$  and  $\nu_y = 0.6135$  when we have different orders of the SMM for our action angle variables. We see that for no square matrix, the iteration method quickly fails to converge at moderately larger amplitudes. Adding even a 3rd order SMM method greatly improves the convergence area. We can see though that 7th order SMM actually decreases the convergence area.

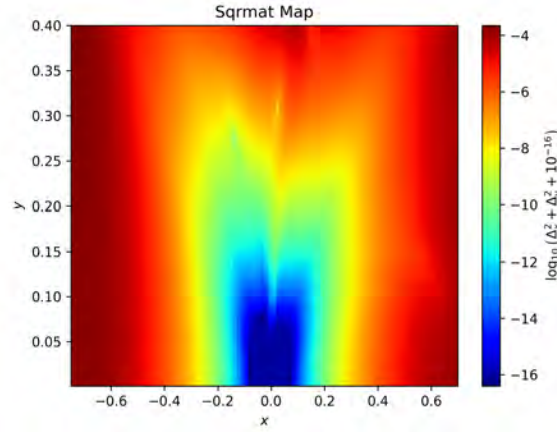


Figure 3.5 Stability map for the Hénon map using third order square matrix method to calculate  $\Delta$  from equation 3.29.

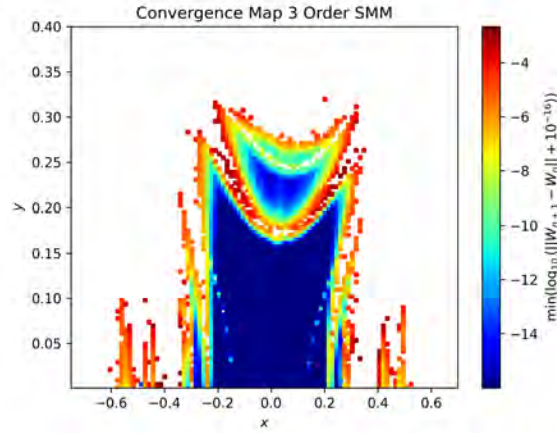


Figure 3.6 Convergence map for the Hénon map using 3rd order SMM eigenvectors as the action angle variables.

This can be explained by the fact that when we use the 7th order SMM we have more higher order terms in the action angle variables. This can excite more higher order terms in the Fourier expansion of the transformation which can lead to a failure to converge.

### 3.3 National Synchrotron Light Source II (NSLS-II)

While toy models, such as the Hénon map, are useful baselines for testing our methods, it is essential to compare the results when applied to real accelerators. The National Synchrotron Light Source II (NSLS-II) is a 3 [GeV] second-generation electron light source at Brookhaven National Lab (BNL) [25]. It is an electron storage ring which uses the synchrotron radiation from



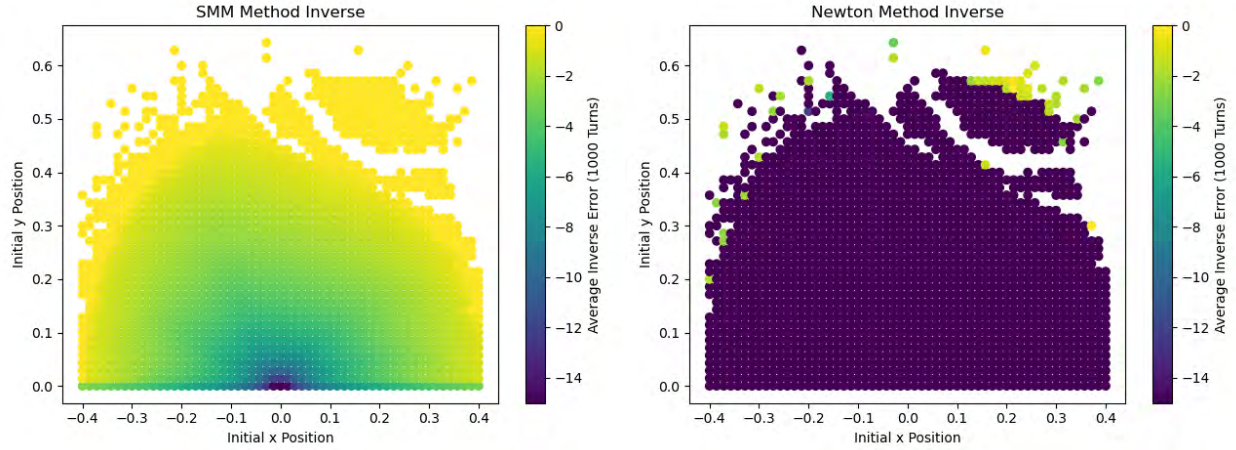


Figure 3.7 Inverse of the 7th order SMM for the 4-D Hénon Map using the parameters  $\nu_x = 0.282$  and  $\nu_y = 0.6135$  (a) and using Newton's method (b) [20].

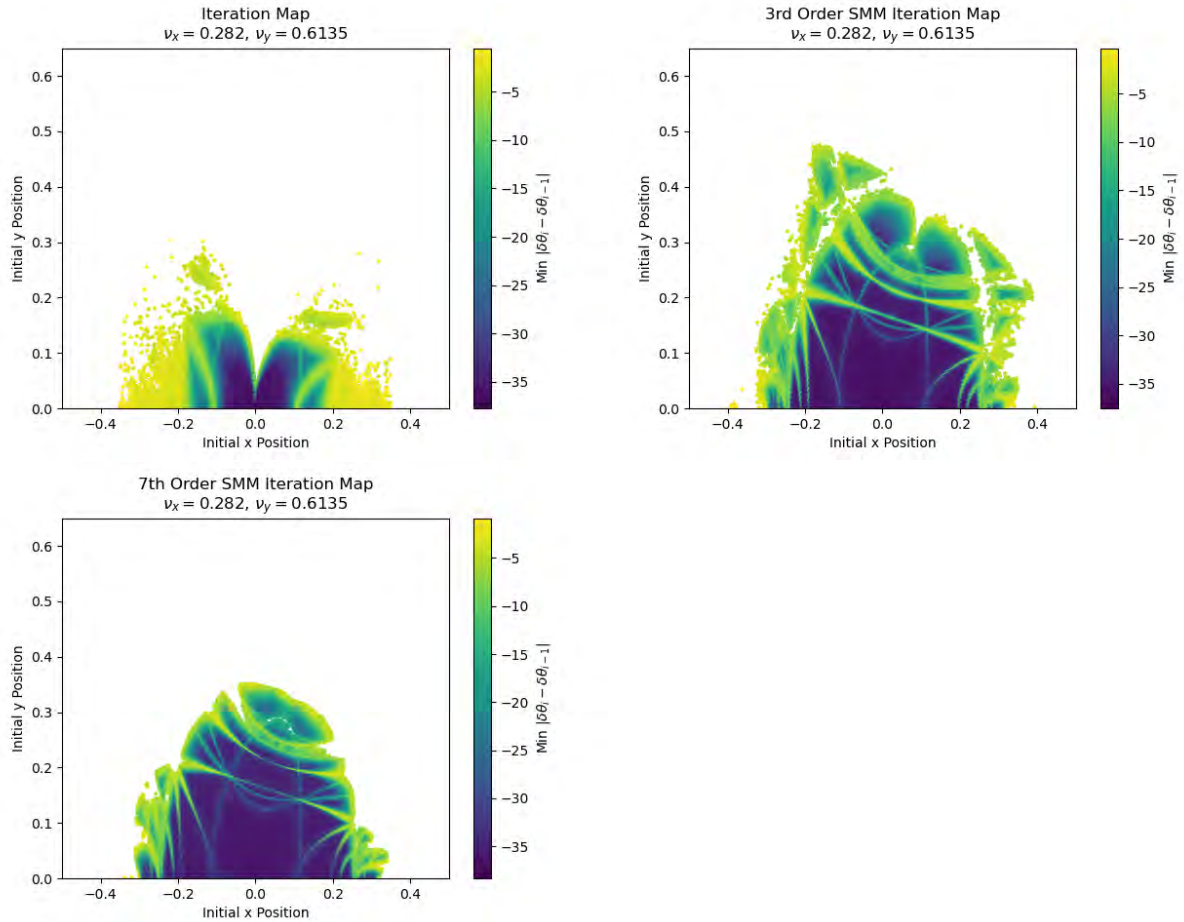


Figure 3.8 Convergence maps for the 4-D Hénon Map using the parameters  $\nu_x = 0.282$  and  $\nu_y = 0.6135$  without SMM (a), with 3rd order SMM (b), and with 7th order SMM (c) for the action-angle variables [20].

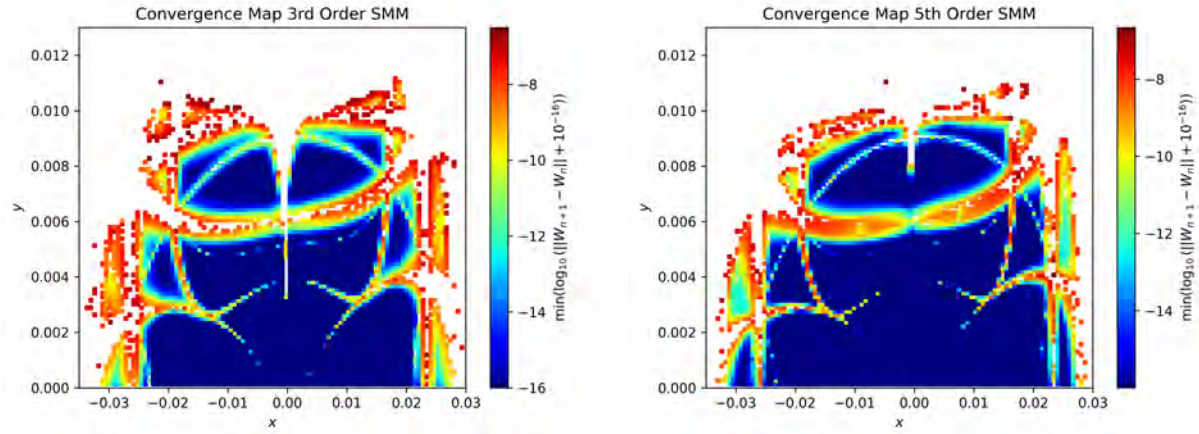


Figure 3.9 The converge maps for the NSLS-II using a 3rd (a) and 5th (b) order SMM method to get the action angle variables.

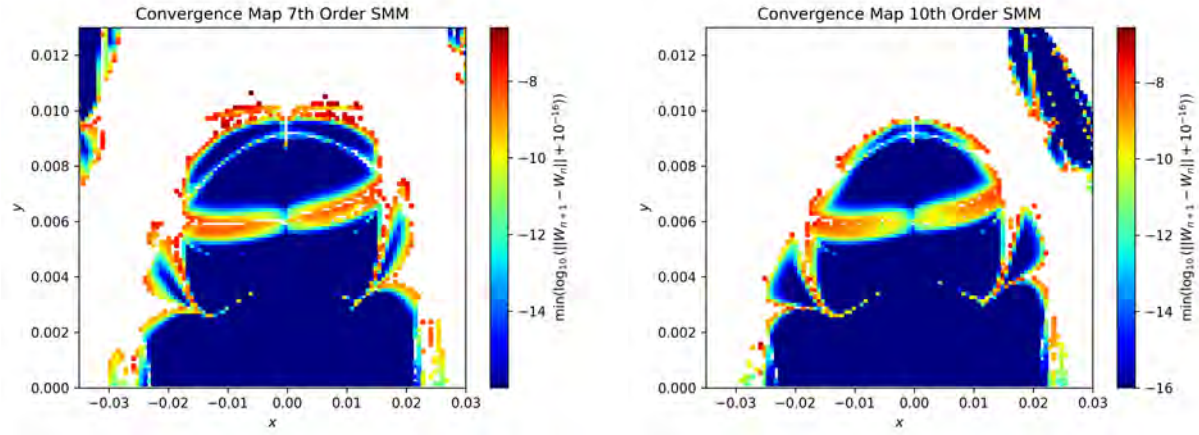


Figure 3.10 The converge maps for the NSLS-II using a 7th (a) and 10th (b) order SMM method to get the action angle variables.

the electrons for experiments in biology and chemistry and biology. This radiation can range from 0.1 eV to >300 keV, depending on what is required [25]. Therefore, it is important to have a very small emittance, in the hundreds or thousands of pm-rad, for brighter light. The aperture is larger horizontally where the electrons are injected into the storage ring.

The electrons orbit a 791.958 [m] storage ring, and their synchrotron radiation is used in various applications in chemistry, biology, and physics.

In the following section, we will detail the comparison of the results from FMA approaches to the NSLS-II, the square matrix method, and the iteration method described in the last section.

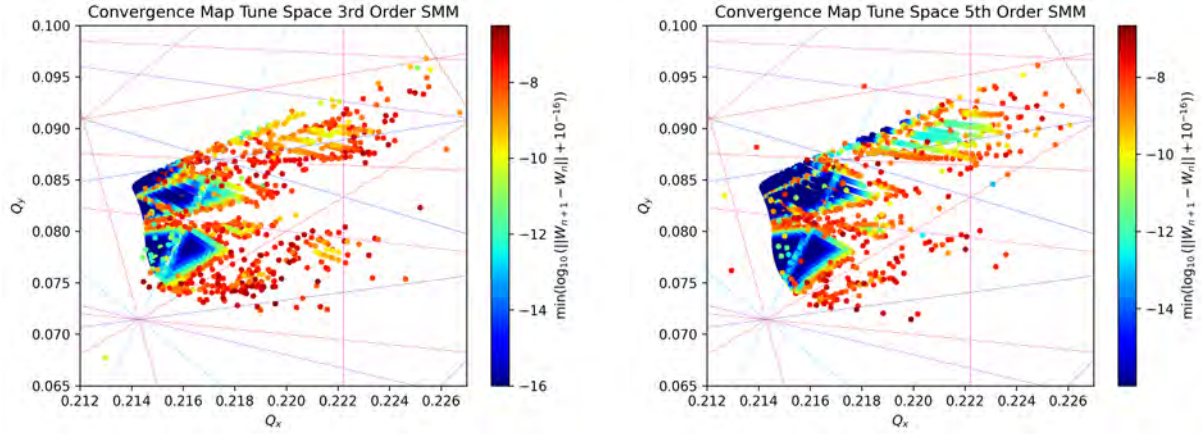


Figure 3.11 The tune maps for the NSLS-II using a 3rd and 5th order SMM method to get the action angle variables. Resonance lines up to the 10th order are shown for reference.

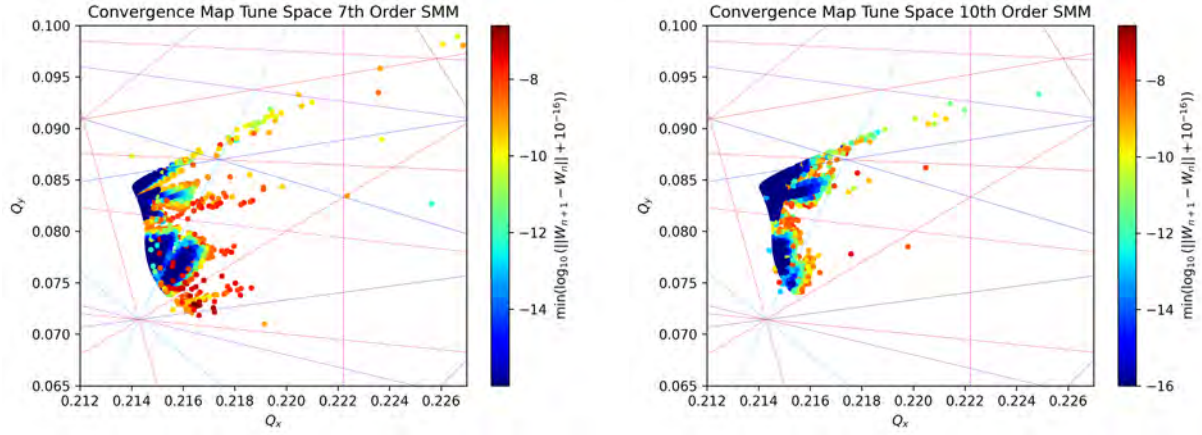


Figure 3.12 The tune maps for the NSLS-II using a 7th and 10th order SMM method to get the action angle variables. Resonance lines up to the 10th order are shown for reference.

Compared to the convergence map in Figure 3.9, we notice that the vertical aperture is much smaller than the vertical aperture from the frequency map in Figure 3.13. One can see, though, that there is a resonance line in the frequency map where the boundary of the convergence map is. As we pointed out when introducing the SMM action-angle variables to the iteration method, since the convergence method can encounter the small denominator problem we pointed out from Equations 3.44 and 3.45, it can be more sensitive to resonances, which cause higher order terms to blow up, and we are unable to converge. This also contributes to the decrease in the convergence area we see in Figures 3.9 and 3.10 when increasing the order of the SMM used.



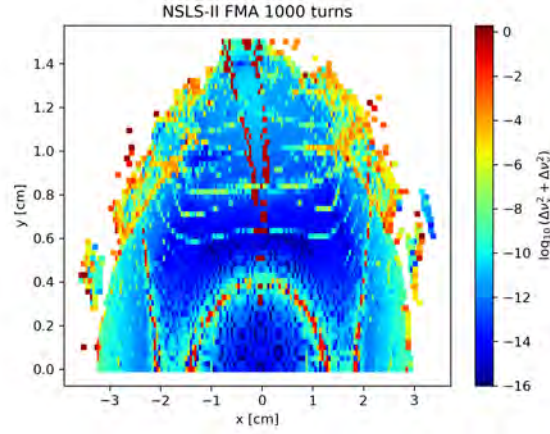


Figure 3.13 Frequency map analysis on the NSLS-II bare lattice for 1000 turns calculated in Elegant.

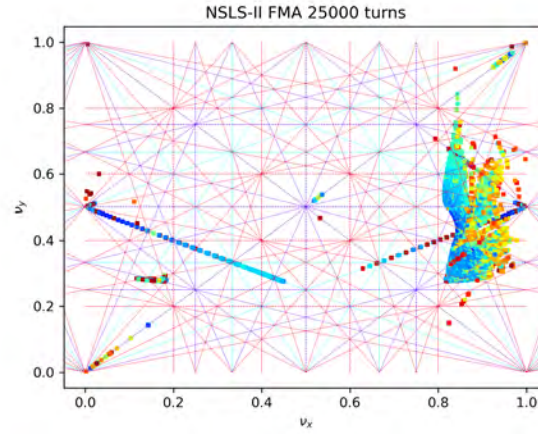


Figure 3.14 Frequency map analysis in the tune space for the NSLS-II bare lattice for 1000 turns calculated in Elegant.

### 3.4 Crab Cavity Study

This section will cover some of the results we showed when we introduced the study at IPAC 2024 [26]. The crab cavity is a proposed component in collider rings to increase luminosity [27]. The idea being that for colliders, beams are transported in separate beam pipes to get them to the desired energy. These beam pipes then need to converge so the bunches can collide. This means that they will cross at some angle. This means the beams will not perfectly overlap, and there will be fewer collisions, in other words, lower luminosity. To increase this value, the crab cavity proposes that a sinusoidal kick is applied to each bunch. They then propagate to the interaction point at an angle in the lab frame. But when we look at the bunches' perspectives, they will collide head-on,

leading to many more collisions. They will have an opposite kick applied after the interaction point to offset the first crab cavity kick. This will naturally introduce new coupling between the longitudinal coordinates and the transverse coordinates. Since our model is a 4-D one, we will not look at both transverse coordinates but instead at one transverse coordinate and the longitudinal one. The details of how this model is implemented are in the next section.

### 3.4.1 Theoretical Model

The model used for this study was a 4-D system looking at the transverse and longitudinal components of the crab cavity (CC) crossing. This is still a 4-D phase space like the other maps, however, there is a significant change by using the longitudinal coordinate in the calculations vs using both transverse coordinates. Both transverse coordinates tend to behave analogously to each other. The longitudinal coordinate tends to be significantly different. The longitudinal coordinates will generally have actions that are orders of magnitude larger than the transverse coordinates, and the longitudinal tunes will generally have smaller tunes as well.

The lattice consists of two crab cavities placed a phase of  $\pi/2$  apart from one another on either side of the interaction point (IP). A simple nonlinear lattice in the longitudinal direction and a linear transverse lattice were used for the rest of the accelerator. It is assumed that there are minimal longitudinal effects from the lattice between the two crab cavities. The crab cavities provide a sinusoidal kick to a proton beam as well as a quadrupole and sextupole kick of the following form [28]:

$$\Delta p_x = \frac{-\tan \theta_c \sin(k_c z)}{k_c \sqrt{\beta_{cc} \beta_{IP}}} + b_2 x \sin(k_c z) + b_3 x^2 \sin(k_c z) . \quad (3.50)$$

To preserve the symplectic nature of the cavity, a longitudinal kick is also imparted on the beam of the following form:

$$\Delta p_z = \frac{-x \tan \theta_c \cos(k_c z)}{\sqrt{\beta_{cc} \beta_{IP}}} + \frac{b_2 k_c}{2} x^2 \cos(k_c z) + \frac{b_3 k_c}{3} x^3 \cos(k_c z) . \quad (3.51)$$

where  $\theta_c$  is the half crossing angle,  $k_c = 2\pi f_c/c$  is the wave number of the crab cavity, and  $b_2, b_3$  are the integrated quadrupole and sextupole strengths respectively.

The parameters of the model are given in Table 3.1. Note that the sextupole strength is unrealistically high. This is due to the rest of the transverse lattice being linear and therefore a larger value was needed in order to see its effects. The rest of the lattice has the following form longitudinally

$$z' = z - \frac{2\pi h c \eta \beta_s}{f_{\text{rf}}} p'_z, \quad (3.52)$$

where

$$p'_z = p_z + \frac{eV_{\text{rf}}}{E\beta_s^2} (\sin \phi - \sin \phi_s), \quad (3.53)$$

$\eta$  is the slip factor,  $\phi$  is the longitudinal phase, and  $\beta_s$  is the ratio of the longitudinal velocity and the speed of light.

### 3.4.2 Results

Figure 3.15(a) shows frequency map analysis using 75,000 turns and NAFF [29] to calculate the tunes of the parameters in Table 3.1. We can compare this to the same parameters and initial conditions but using the iteration method in Fig. 3.16(a). The color bar on this map shows the base 10 log of the minimum error that the method converges to, the error being  $\|\mathbf{X}_{n+1} - \mathbf{X}_n\|$  where  $n$  is the iteration number and  $\mathbf{X}$  is the phase space vector.

We can see when we plot the same points in tune space in Fig. 3.16(b), that the higher errors correspond to the resonance lines of  $3Q_x + mQ_z = 1$ , where  $m = 5, 6, 7$  which correspond to an 8th, 9th, and 10th order resonance respectively. The 8th order resonance corresponds to the points on the corners of the map in  $x, z$  space. The next lines in from those corners correspond to the 9th order resonance line. Finally, the lines near  $x = 0$  correspond to the 10th order resonance lines.

This  $m = 7$  line also appears when we turn off the sextupole as shown in Fig. 3.19. This is because with no sextupole, the iteration method only shows a tune shift in the  $Q_z$  which still crosses

the  $3Q_x + 7Q_z = 1$  resonance line. This suggests that the crab cavity with a sextupole will result in an increase in horizontal tunes that then encounter the 8th and 9th order resonances described before.

Figure 3.17 shows the points where we see an increase in the longitudinal tunes. This further shows how the crab cavity causes the tune shifts to be dependent on the particles transverse and longitudinal positions.

We can also gather more information by removing the time dependency. Figure 3.18 shows the convergence map when there is no time dependency, i.e., just a normal sextupole instead of a crab cavity with a time-dependent sextupole. This removes the coupling between the transverse and longitudinal coordinates. However we still see some small traces of the resonances which we

Table 3.1 Parameters of Crab Cavity (CC) Model

| Parameter                                | Symbol          | Value                       |
|------------------------------------------|-----------------|-----------------------------|
| Half crossing angle                      | $\theta_c$      | 25 [mrad]                   |
| CC wave number                           | $k_c$           |                             |
| Transverse Beta Function at CC           | $\beta_{cc}$    | 1300 [m]                    |
| Transverse Beta Function at IP           | $\beta_{IP}$    | 90 [cm]                     |
| Transverse Position RMS                  | $\sigma_x$      | 120 [ $\mu\text{m}$ ]       |
| Longitudinal Position RMS                | $\sigma_z$      | 7 [cm]                      |
| Longitudinal Momentum Deviation RMS      | $\sigma_{p_z}$  | $6.6 \times 10^{-4}$        |
| Integrated Quadrupole Strength           | $b_2$           | 0 [1/m]                     |
| Integrated Sextupole Strength            | $b_3$           | 500,000 [1/m <sup>2</sup> ] |
| Synchronous Phase                        | $\phi_s$        | 0                           |
| Beam Energy                              | $E$             | 275 [GeV]                   |
| RF Voltage                               | $V_{rf}$        | 15.8 [MV]                   |
| RF Frequency                             | $f_{rf}$        | 591 [MHz]                   |
| Harmonic Number                          | $h$             | 7560                        |
| Momentum Compaction Factor               | $\alpha_c$      | $1.5 \times 10^{-3}$        |
| Linear Tunes (Transverse / Longitudinal) | $\nu_x / \nu_z$ | 0.310 / 0.015               |

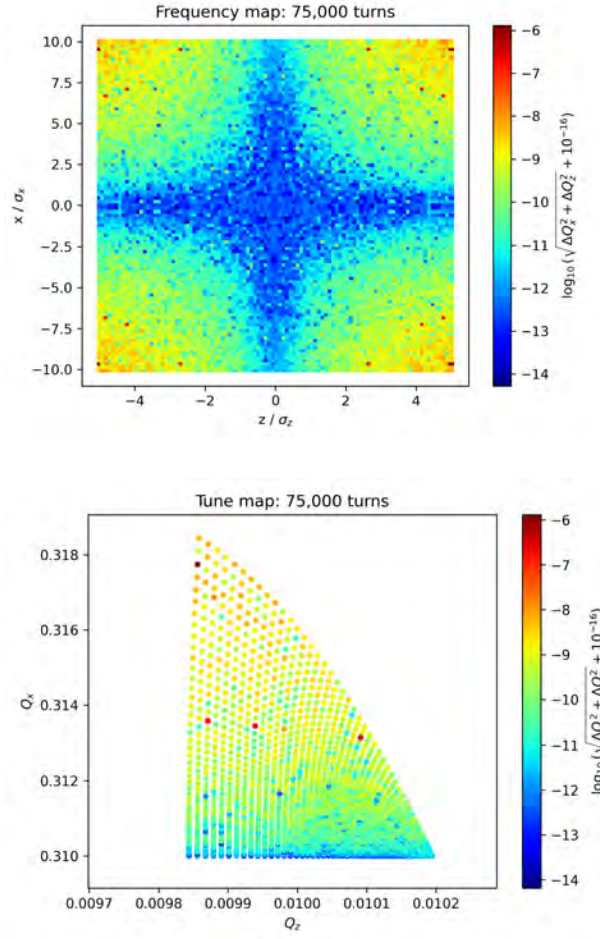


Figure 3.15 Frequency map (a) and tune map (b) of the crab cavity crossing at the EIC using the parameters in Table 3.1.

do not expect due to the lack of coupling. FMA does not show any coupling either. This would suggest that this  $m = 7$  resonance is most likely a numerical effect from the iteration method. More specifically, this could be that during one of the iterations, the predicted tunes land very close to this resonance. This would cause the corresponding Fourier component to explode as being close to a resonance gives a near zero denominator when we update it. I would also think this is why we see the lines in Fig. 3.19. But this wouldn't necessarily discount their appearance in Fig. 3.16 as the color scales are different.

Figure 3.15(a) shows instability near the corners where we see the  $m = 5, 6$  resonances in Fig. 3.16(a). This could mean that the instabilities we see in the frequency map are caused by the



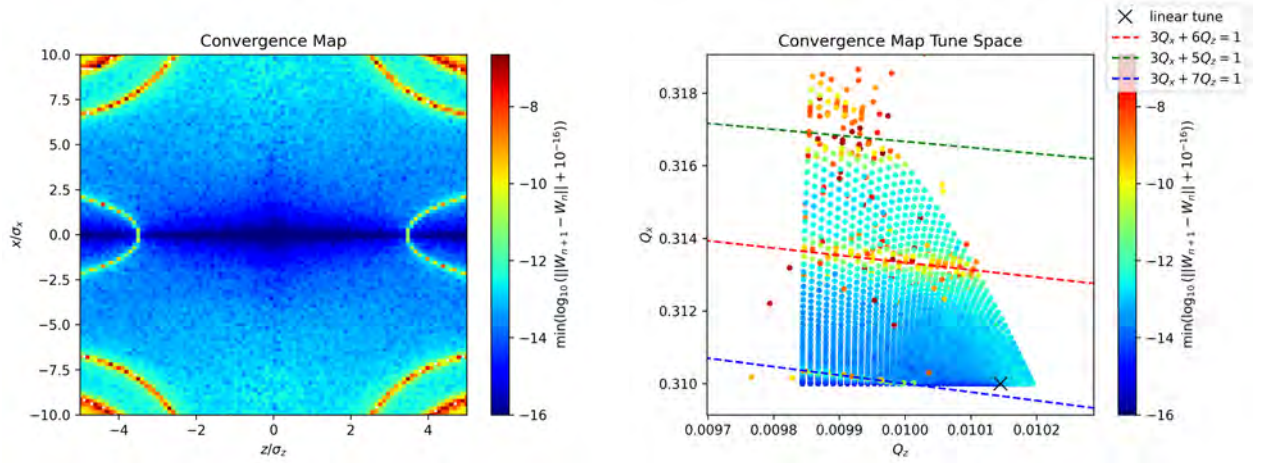


Figure 3.16 Convergence map (a) and tune map (b) of the crab cavity crossing at the EIC using the parameters in Table 3.1.

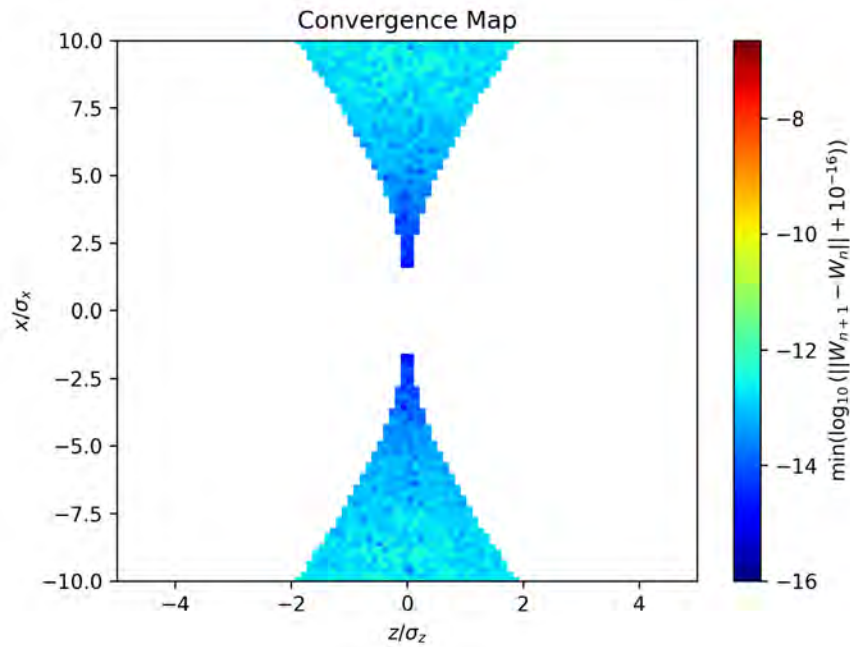


Figure 3.17 Area where we see an increase in the longitudinal tunes compared to the linear tune from Fig. 3.16.

resonance lines that are more clearly shown in the convergence map. Since resonances appear more clearly in the convergence maps, the iteration method could be useful for finding resonances that FMA does not show as clearly.

The shape of the tune space in Fig. 3.18(b) shows that the increase in the longitudinal tunes, as

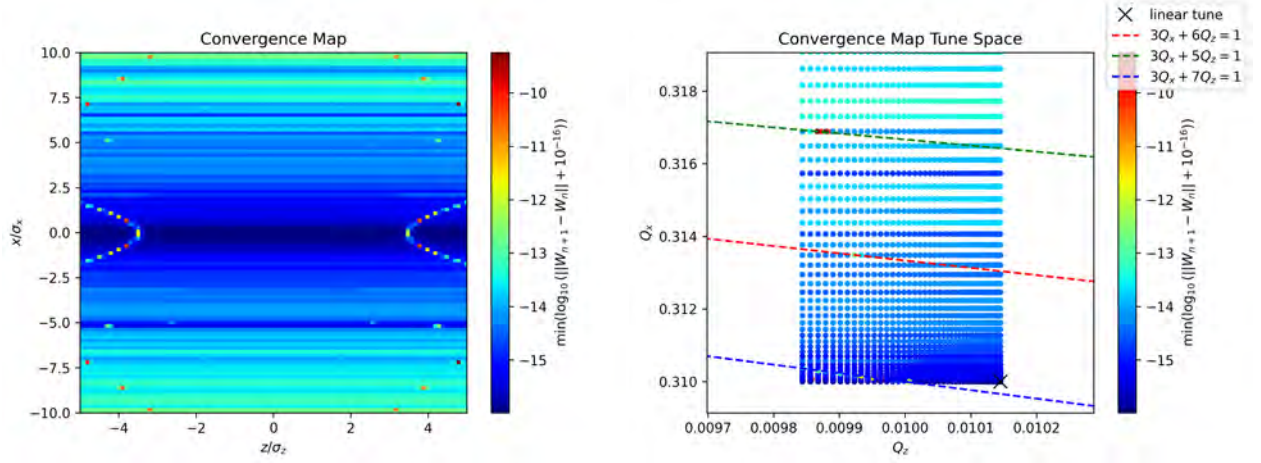


Figure 3.18 Convergence map (a) and tune map (b) with a time independent sextupole and no crab cavity.

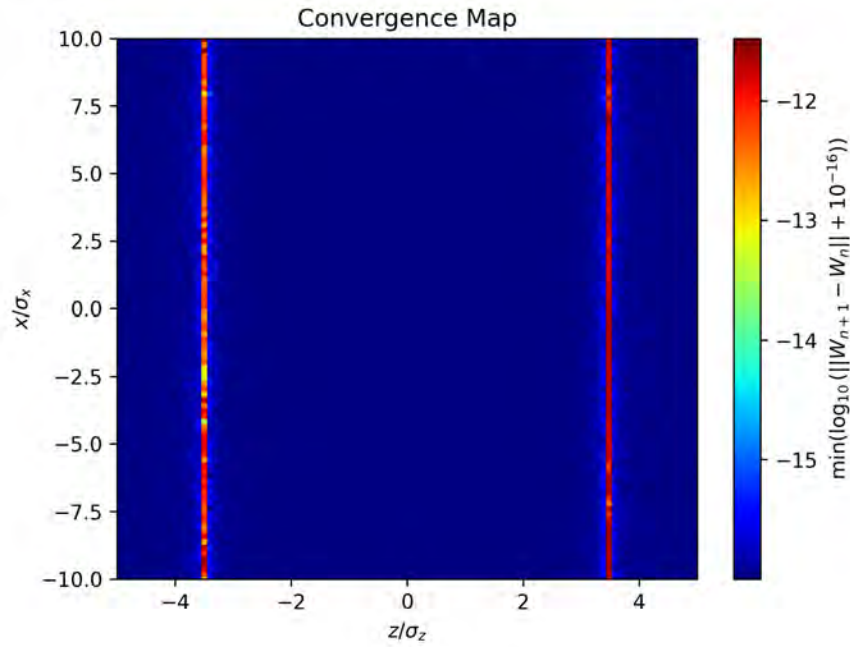


Figure 3.19 Convergence map using the parameters in table 3.1 but with no sextupole ( $b_3 = 0$ ).

well as the curved horizontal and longitudinal tune boundary we see on the right in Fig. 3.15(b), are due to the coupling caused by the crab cavity. The crab cavity also seems to strengthen the resonances discussed previously.

### **3.5 Discussion**

The iterative method has proven to be able to analyze the crab cavity crossing at the Electron Ion Collider. It has identified similar tune shifts as FMA such as the increase in longitudinal tune caused by the crab cavity. It has also identified resonances encountered due to tune shifts caused by the crab cavity and its time dependent sextupole. Some of these resonances seem to coincide with areas where FMA shows instability. Future work to analyze the full 6-D phase with this method is being conducted.

## CHAPTER 4

### MACHINE LEARNING APPROACH

#### 4.1 Koopman Operator

Data-driven methods have become helpful in analyzing accelerator systems. This data can be obtained through repeated iterations of a mathematical model of a lattice or through beam position monitor measurements on an actual beam. Koopman operators [30] are a tool that benefits from these data-driven techniques. Here, we plan to leverage the Koopman operator so that we can leverage turn-by-turn tracking data in order to calculate the frequencies of different particles and relate them to constants of motion. These insights can provide possible long-term information about the system by identifying resonances and unstable regions of the phase space.

The Koopman operator is a method for simplifying a phase space. What Koopman described was that for a nonlinear dynamical system, one can represent the evolution of the measurables of that system as a linear operator acting on all possible measurables of that system. This means that the linear space that this exists in has infinitely many degrees of freedom [31]. It is therefore useful to find some finite approximation of this operator. We can do this by finding an invariant subspace spanned by a finite number of eigenfunctions of the Koopman Operator. If we can find a set of eigenfunctions that, when acted on by the Koopman operator, remain in that same subspace, then this is an invariant subspace and the dynamics will be linear [30].

The Koopman theory applies to both continuous and discrete dynamical systems, but since our research focuses on the analysis of the transfer map, we will only discuss the latter. If we define our discrete system as

$$\mathbf{X}_{k+1} = \mathbf{F}(\mathbf{X}_k). \quad (4.1)$$

$$\mathbf{X} \in \mathbb{R}^n \quad (4.2)$$

$$\mathbf{F} : \mathbf{X} \rightarrow \mathbf{X} \quad (4.3)$$

where  $\mathbf{F}$  is the operator that represents our discrete map, sometimes referred to as a flow map

operator. We have a vector  $g$  that is a set of measurement functions of the state vector  $\mathbf{X}$ . The Koopman operator  $\mathcal{K}$  is then

$$\mathcal{K}g(\mathbf{X}) = g(F(\mathbf{X})). \quad (4.4)$$

$$\mathcal{K} : g(\mathbf{X}) \rightarrow g(\mathbf{X}) \quad (4.5)$$

$$g : \mathbf{X} \rightarrow \mathbb{C} \quad (4.6)$$

If  $g$  is a linear vector space of functions, Equation 4.6 shows that the Koopman operator must be a linear operator.

$$\mathcal{K}(a_1g_1(\mathbf{X}) + a_2g_2(\mathbf{X})) = a_1g_1(F(\mathbf{X})) + a_2g_2(F(\mathbf{X})) \quad (4.7)$$

$$= a_1\mathcal{K}g_1(\mathbf{X}) + a_2\mathcal{K}g_2(\mathbf{X}). \quad (4.8)$$

To begin our approach to using the Koopman operator to simplify our phase space, it can be helpful to try to find eigenfunctions of the operator. That is,  $\varphi$  is an eigenfunction of  $\mathcal{K}$  if it satisfies

$$\varphi(\mathbf{X}_{k+1}) = \mathcal{K}\varphi(\mathbf{X}_k) = \lambda\varphi(\mathbf{X}_k) \quad (4.9)$$

where  $\lambda$  is its corresponding eigenvalue. So if we want to simplify our phase space using Koopman theory, we should look to find these eigenfunctions of the Koopman operator as they evolve linearly based on the dynamics of our nonlinear system. These functions and their eigenvalues can provide insight into the properties of the system, similar to how we gain information from the eigenvalues and eigenvectors of linear systems.

We then aim to construct a neural network to approximate an encoder that represents an eigenfunction  $\varphi_j$  with an eigenvalue of  $\exp(2\pi\nu_j)$  and its associated properties.

$$\varphi : \mathbb{R}^{2m} \rightarrow \mathbb{C}^{2m} \quad (4.10)$$

$$\mathbf{W}_n = \varphi(\mathbf{X}_n) \quad (4.11)$$

$$\mathbf{W}_{n+1} = \begin{pmatrix} e^{2\pi i \nu_1} & 0 & \dots & 0 \\ 0 & e^{-2\pi i \nu_1} & \dots & 0 \\ \vdots & \ddots & \dots & \vdots \\ 0 & \dots & e^{2\pi i \nu_m} & 0 \\ 0 & \dots & 0 & e^{-2\pi i \nu_m} \end{pmatrix} \mathbf{W}_n \quad (4.12)$$

where  $m$  is the number of spatial dimensions in the system and the matrix in equation 4.12 is diagonal. Note that each pair of coordinates in  $\mathbf{W}$  is a complex conjugate pair. In practice, this turns  $2m$  real coordinates into  $m$  complex coordinates, where the real part is position-like and the imaginary part is momentum-like coordinate.

To find the representation of the encoder, we construct the following neural network 4.1. The data on which the network will be trained are discrete data from some Poincaré map. This can be turn-by-turn data from an accelerator, modeled or experimental, or iterations of some other discrete nonlinear mathematical map, i.e., the Standard map, Henon map, etc. When the model acts on a phase space coordinate, it is passed to two distinct models: one is a traditional neural network that predicts the particle's frequency, and the other is a symplectic neural network that represents the approximate eigenfunction. The output of the frequency model  $r_i$  for each spatial dimension  $i$  is passed through 4.13. After the transformation and tune prediction, the coordinates are rotated by an angle of  $\nu_i$  times  $2\pi$  in the  $w_i$  plane for each dimension. Then, the inverse of the encoder is applied, which should result in the coordinates after one turn or iteration from the starting coordinates.

$$\nu_i = 1 - \frac{1}{1 + e^{\alpha r_i}}, \quad \text{with } r_i \in (-\infty, \infty), \quad \nu_i \in (-1, 1) \quad (4.13)$$

where  $\alpha \leq 1$  is a hyper-parameter to give the function a smaller slope.

To obtain a symplectic transformation for the encoder, we implemented linear and activation layers called SympNets [4]. Equations 4.21 and 4.23 show examples of these symplectic network layers.  $a$  is an unbounded parameter of the diagonal of an  $m \times m$  matrix and  $S_j$  are  $m \times m$  symmetric matrices, where  $m$  is the number of spatial dimensions and  $p$  and  $q$  are  $m \times 1$  vectors with momentum and position data respectively.

Previous attempts at finding this encoder with neural networks have been made. Our structure is similar to that of Lusch et al. [3]. The key difference in ours is the use of symplectic neural networks in the encoder. This is a reasonable approach because the encoder is unitary in Hamiltonian systems [30]. That condition will be met by the symplectic condition, which the structure of the SympNets enforces despite having unbounded network parameters. This means no additional terms are needed in the loss function to maintain symplecticity, which are required when using traditional neural networks. The layers also use significantly fewer parameters than a traditional network. This naturally increases the network training speed. The structure of SympNets also gives an exact inverse for each layer. For our network, this means that we do not need additional network layers for the inverse of the Koopman operator, which is used during training. Traditional networks trying to find Koopman operators must include extra layers for the inverse and add a term to the loss function to encourage those layers to approximate an inverse, which likely will not be exact. The following sections will discuss the details of the SympNet structure and loss function used.

## 4.2 The Problem Setup

We tested the network on two different 2-D nonlinear maps, the Standard map and the Octopole McMillan map. The Octopole McMillan map is integrable, so we expect the model to be able to find an action for the entire phase space [32]. On the other hand, the standard map is not integrable, so there is no constant of motion for the entire phase space. Nevertheless, we aim to find a local action that is approximately constant for a range of amplitudes. The Standard map is described by the following:

$$p_{n+1} = p_n + K \sin \theta_n \quad (4.14)$$

$$\theta_{n+1} = \theta_n + p_{n+1}$$

where  $\theta$  is then acted on by modulus  $2\pi$ . The Octopole McMillan map [32], which will be called the McMillan map for simplicity, is described by the following:

$$\begin{aligned} q_{n+1} &= p_n \\ p_{n+1} &= -q_n - \frac{2\epsilon p_n}{p_n^2 + \lambda} \end{aligned} \tag{4.15}$$

We are seeking a transformation of variables through the encoder, which has the following properties:

$$\varphi : \mathbb{R}^2 \rightarrow \mathbb{C}^2 \tag{4.16}$$

$$\mathbf{W} = \varphi(\mathbf{X}) \tag{4.17}$$

$$\mathbf{W}_{n+1} = \begin{pmatrix} e^{2\pi i \nu} & 0 \\ 0 & e^{-2\pi i \nu} \end{pmatrix} \mathbf{W}_n \tag{4.18}$$

where  $\mathbf{X} \in \mathbb{R}^2$  is the original phase space coordinates and  $\nu \in [0, 1]$  is the betatron tune of the system.

Since many nonlinear systems that describe accelerators are non-integrable, no transformation can perfectly transform the entire phase space into a pure rotation. In this case, however, the McMillan map is integrable, so a true constant of motion exists. But in general, for near-integrable systems, we can find a transformation that gives an approximate constant for some fraction of the phase space.

### 4.3 Network Setup

#### 4.3.1 Data

The models in the following sections had a similar process for preparing the data to be used in the model. Multiple orbits from a range of initial amplitudes were selected and tracked for a number of turns. In a practical accelerator, this could be data from a beam position monitor, but in our case, simulations were used. So we have  $m$  particles tracked for  $n$  turns. Each phase space coordinate belonging to the same initial condition will be denoted with a superscript and each turn will be denoted with a subscript. So  $\vec{x}_i^j$  is the  $i$ th turn of the  $j$ th particle. Each training vector,  $\vec{v}$ , will hold a subset of consecutive phase space coordinates belonging to the same initial



condition. For example, the collection of  $l$  phase space coordinates from the  $j$ th initial condition,  $(\vec{x}_i^j, \vec{x}_{i+1}^j, \vec{x}_{i+2}^j, \dots, \vec{x}_{i+l}^j)$ , would be organized as

$$\vec{v}_k = \begin{pmatrix} \vec{x}_i^j \\ \vec{x}_{i+1}^j \\ \dots \\ \vec{x}_{i+l}^j \end{pmatrix} \quad (4.19)$$

The output from the training vector in Equation 4.19 would be all the same phase space coordinates one turn later, i.e.

$$\vec{v}_k = \begin{pmatrix} \vec{x}_{i+1}^j \\ \vec{x}_{i+2}^j \\ \dots \\ \vec{x}_{i+l+1}^j \end{pmatrix}. \quad (4.20)$$

The way to select  $l$  will become apparent in the next section describing the model structure as well as why the training vectors are groups of phase space coordinates instead of the individual points.

### 4.3.2 Model

Similar to a traditional neural network, the Koopman operator network consists of a series of linear and activation layers. The layers are split into three main sections, one is the encoder which aims to find the transformation from the phase space of the tracking data into a space where the motion is a pure rotation. The second is the frequency layers, which aim to predict the frequency of the pure rotation. They are composed of traditional network layers as the symplectic condition is not needed. Then there is the decoder, which aims to be the inverse of the encoder.

Using SympNets[4] versions of these layers instead of the traditional ones gives a few advantages, one being that the transformation found in training obeys the symplectic condition required for Hamiltonian systems. Another is that due to the configuration of these layers, an exact inverse of the encoder can be systematically constructed from the weights of the layers. This means that there

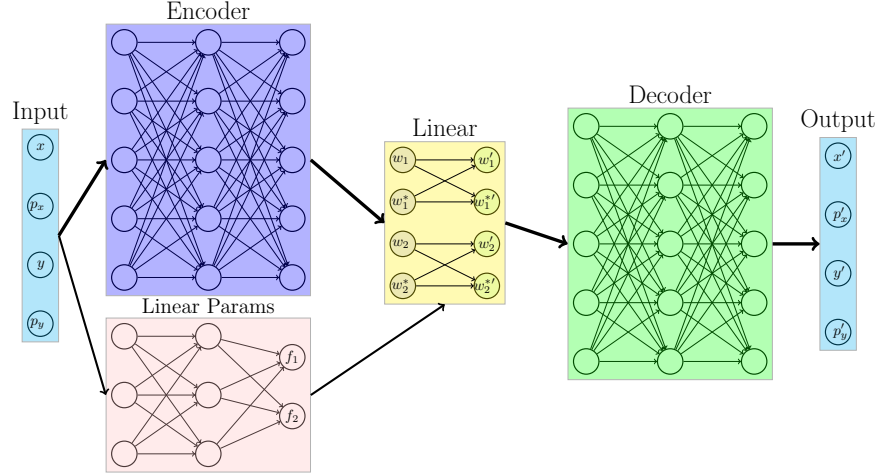


Figure 4.1 An illustration of a 4-D representation of the neural networks that were used. The encoder is our representation of the Koopman operator's eigenfunctions. The Linear section applies the Koopman operator with eigenvalues from the Linear parameters. The decoder is the inverse of our eigenfunctions.

is no need to have additional network layers to create a decoder, which would slow training, and there are no approximate methods needed to decode the data either. Figure 4.1 shows an example of this structure in 4-D.

Equations 4.21 and 4.23 show examples of a SympNet upper linear layer of size  $n$  and a lower activation layer with function  $\sigma$ .  $a$  is an unbounded parameter of the diagonal of a  $m \times m$  matrix and  $S_j$  are symmetric  $m \times m$  matrices, where  $m$  is the number of spatial dimensions and  $p$  and  $q$  are  $m \times 1$  vectors with momentum and position data, respectively. For our models, the encoder was composed of a lower linear layer, an upper activation layer, an upper linear layer, a lower activation layer, and an upper linear layer.

The frequency is retrieved from a traditional network comprised of a linear layer to go from 2 input parameters to 100 output parameters, a hyperbolic tangent activation layer, another linear layer with 100 inputs and outputs, another hyperbolic tangent activation layer, and a final linear layer to go from 100 inputs to 1 output. This one output is the parameter  $r$ , which is then put into equation 4.13 where  $\alpha$  is some predefined hyperparameter of the frequency model, 0.25 in our case. This ensures that the unbounded output  $r$  always gives a unique tune,  $Q$ , between 0 and 1.

$$\mathcal{L}_n^{\text{up}} = \begin{pmatrix} I & 0/S_n \\ S_n/0 & I \end{pmatrix} \cdots \begin{pmatrix} I & 0 \\ S_2 & I \end{pmatrix} \begin{pmatrix} I & S_1 \\ 0 & I \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} \quad (4.21)$$

$$\begin{pmatrix} I & S \\ 0 & I \end{pmatrix}^{-1} = \begin{pmatrix} I & -S \\ 0 & I \end{pmatrix} \quad (4.22)$$

$$\mathcal{N}^{\text{low}} \begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} p \\ \text{diag}(a)\sigma(p) + q \end{pmatrix} \quad (4.23)$$

$$\left(\mathcal{N}^{\text{low}}\right)^{-1} \begin{pmatrix} P \\ Q \end{pmatrix} = \begin{pmatrix} P \\ Q - \text{diag}(a)\sigma(P) \end{pmatrix} \quad (4.24)$$

### 4.3.3 Training

With the training vectors as described in the previous section, the way the input vector  $\vec{v}_k$  passes through the model is that each phase space vector will be individually transformed by the encoder. This encoder acts as the eigenvector of the Koopman operator.

$$\tilde{\varphi}(\vec{v}_k) = \begin{pmatrix} w_i^j \\ w_{i+1}^j \\ \dots \\ w_{i+l}^j \end{pmatrix} \equiv \vec{V}_k. \quad (4.25)$$

where  $w_i^j$  is the coordinate of the complex phase space  $\vec{x}_i^j$  after being transformed into the transformed phase space, where ideally the motion is a rotation from one turn to the next. The real and imaginary components of  $w_i^j$  correspond to the two phase space dimensions in  $\vec{x}_i^j$ . As such, these transformed coordinates will all rotate by an angle given by the frequency layers after the coordinate is passed into it. So  $\vec{x}_i^j$  will be passed into the frequency layers and given some tune  $Q_i^j$ . Then the corresponding transformed phase space coordinate is rotated by that angle

$$\vec{w}_{i+1}'^j = e^{2\pi i Q_i^j} \vec{w}_i^j \quad (4.26)$$

where the prime denotes that it is an approximation, not the actual value of  $\vec{x}_{i+1}^j$  after passing through the encoder. So, the phase space vector  $\vec{V}_k$  then becomes  $\vec{V}_k'$  where

$$\vec{V}_k' = \begin{pmatrix} w_{i+1}'^j \\ w_{i+2}^j \\ \dots \\ w_{i+l+1}^j \end{pmatrix}. \quad (4.27)$$

The decoder is applied to all transformed phase space coordinates of  $\vec{V}_k'$ . This gives us our output vector  $\vec{v}_k'$ , not to be confused with  $\vec{v}_k$ ,

$$\vec{v}_k' = \varphi^{-1}(\vec{V}_k') = \begin{pmatrix} \vec{x}_{i+1}'^j \\ \vec{x}_{i+2}^j \\ \dots \\ \vec{x}_{i+l+1}^j \end{pmatrix}. \quad (4.28)$$

The mean square error between the approximated one turn through the map,  $\vec{v}_k'$ , and the actual one turn through the map,  $\vec{v}_k$ , is one of the components of our loss function. The rest of the components of the loss model are described in the following section.

#### 4.3.4 Loss Function

Defining the loss function is a critical part of finding the Koopman Operator. We want to model to find its own frequencies and constants in the transformed phase space, as opposed to training them on values from NAFF, as this would defeat the purpose of this being a different approach not dependent on the other methods discussed. So we want our loss function to train our model on the properties of the transformed phase space, not any actual values. This is why we grouped the training vectors.

There are three main features of our loss function. The first part is the MSE Loss from the tracking data. This is included to ensure that the rotation in our phase space corresponds to one iteration of the map. This is defined as

$$\text{MSE} = |\vec{v}'_k - \vec{v}_k|^2. \quad (4.29)$$

The second property we want to ensure is that all particles on the same KAM torus have the same amplitude in the transformed phase space. Since each component of the vector  $\vec{V}_k$  is from the same orbit, the second component of our loss function corresponds to the variation in the amplitude of each component.

$$\text{VAR} = \text{Var}(|\vec{V}_k|) \quad (4.30)$$

where

$$|\vec{V}_k| \equiv \begin{pmatrix} |w_i^j| \\ |w_{i+1}^j| \\ \dots \\ |w_{i+l}^j| \end{pmatrix} \quad (4.31)$$

The third property is a constant frequency for the rotation of all particles on the KAM torus. So for each component of  $\vec{V}_k$  we define an angle  $\theta_m$  such that

$$\theta_m = -i \log \left( \frac{w_{m+1}}{w_m} \right). \quad (4.32)$$

Then we define  $\vec{\Theta}$  as

$$\vec{\Theta} = \begin{pmatrix} \cos \theta_i \\ \cos \theta_{i+1} \\ \dots \\ \cos \theta_{i+l-1} \end{pmatrix} \quad (4.33)$$

and the component of our loss function is the variance of each component of  $\vec{\Theta}$ .

$$\text{TUNE} = \text{Var}(\vec{\Theta}) \quad (4.34)$$

We then need to combine the values of Equations 4.29, 4.30, and 4.34 into one loss value. The important part to note when doing this operation is that the values of each component could be very different in magnitude. So we first normalize each component. But when we normalize the components, it is important to still retain the gradient calculated with PyTorch so that the gradient descent operation can still be performed and the model can be trained. So, the operation is

$$\text{Loss} = \sum_i^3 \frac{L_i}{\tilde{L}_i} \quad (4.35)$$

where  $L_i$  is one of the loss components, and  $\tilde{L}_i$  is that same loss component, but detached from the gradient [33]. So while the values of  $L_i$  will be 1, they will each have different non-zero gradients that will train the model parameters.

#### 4.4 Standard Map

We begin with the standard map shown in Equation 4.14 using  $K = -1/2$ . The negative is so that the map has a stable fixed point at  $q = 0$ . The model was trained on 15 orbits with  $q \in [10^{-3}, 1.2]$  for  $2 \times 10^9$  iterations. Figure 4.2 (a) shows multiple orbits in the standard map phase space. Figure 4.2 (b) shows the same orbits after being acted on by the encoder. It is easy to see that the dynamics in the Koopman space are much closer to linear. The amplitudes are plotted by turn in Figure 4.4, which shows a clearer picture of this effect. Ideally, the line width would be zero, but we can see the variations in the constant of motion increasing with amplitude.

Figure 4.5 shows the amplitude from the encoder turn by turn compared to the action from the C-S parameters (Equation 1.119). We can see that the variance is greatly reduced by the encoder, meaning our Koopman operator is close to linear. This would again suggest that our encoder is an accurate approximate eigenfunction of the Koopman operator.

However, looking only at the amplitudes does not give the full picture of the reduction in nonlinearity. We can instead look at polar plots of the motion, Figures 4.6 and 4.7. The amplitude

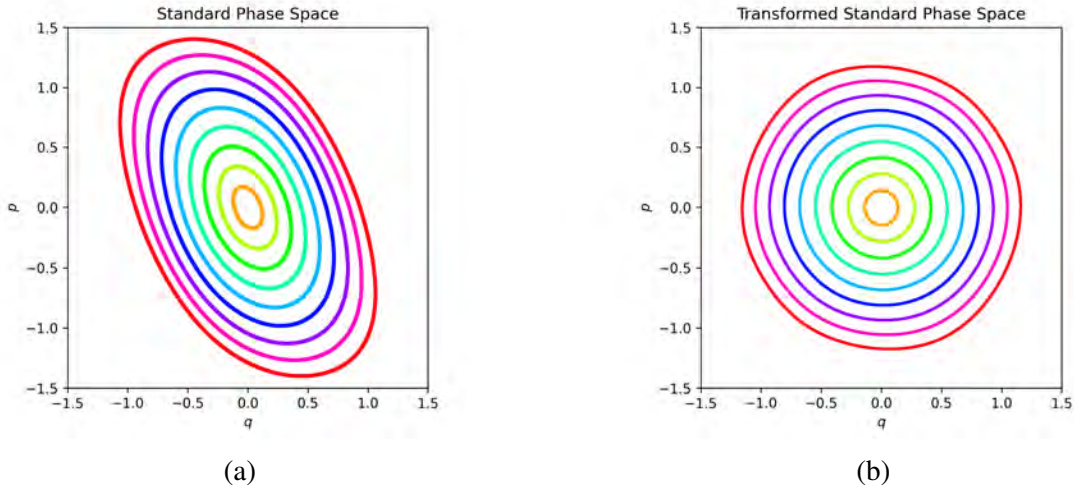


Figure 4.2 The phase space of the standard map with  $K = \mathbf{X}$  (a) and the same phase space after being passed through the encoder to the Koopman subspace (b).

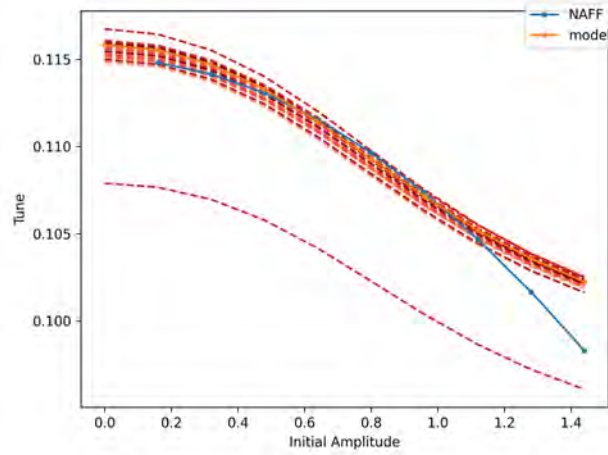


Figure 4.3 The tune shift with amplitude calculated from NAFF (blue) and the model (orange). In red are the model's predictions during previous epochs, getting lighter the older the epoch.

of these plots is naturally the action, and the angle we place them is the change in phase from one turn to the next. Recall that when we characterize the Hamiltonian in terms of action-angle variables, it becomes

$$H(J, \phi) = H_0(J) + H_1(J, \phi) = J\nu + H_1(J, \phi) \quad (4.36)$$

where  $H_0$  is our linear Hamiltonian and  $H_1$  is the nonlinear component. We find from Hamilton's equations of motion

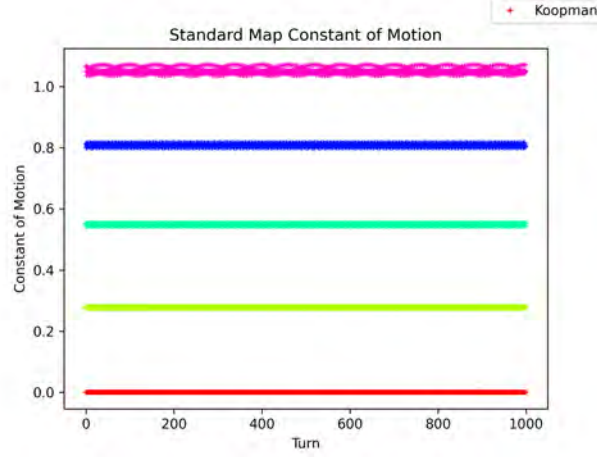


Figure 4.4 The amplitude of the motion of the standard map in the Koopman subspace.

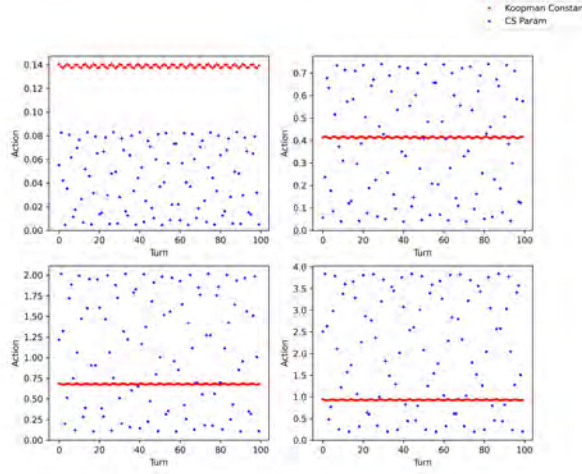


Figure 4.5 The amplitude of the motion of the standard map in the C-S formalism (blue) vs the same orbits in the Koopman subspace (red).

$$\frac{\partial H}{\partial J} = \dot{\phi} = \nu + \frac{\partial H_1}{\partial J} \quad (4.37)$$

$$\frac{\partial H}{\partial \phi} = -j = \frac{\partial H_1}{\partial \phi} \quad (4.38)$$

that nonlinearity will give us an action smear (Equation 4.38) and tune shift (Equation 4.37). When we plot linear motion like this, we would see that each orbit is on a single point because there is no tune shift or action smear. In Figures 4.6 and 4.7, we can see a clearer picture of how the encoder reduces nonlinearity by reducing these hallmarks of nonlinearity. With Figure 4.6 showing the comparison to the C-S action-angle variables and the encoder for each orbit in each plot, and Figure



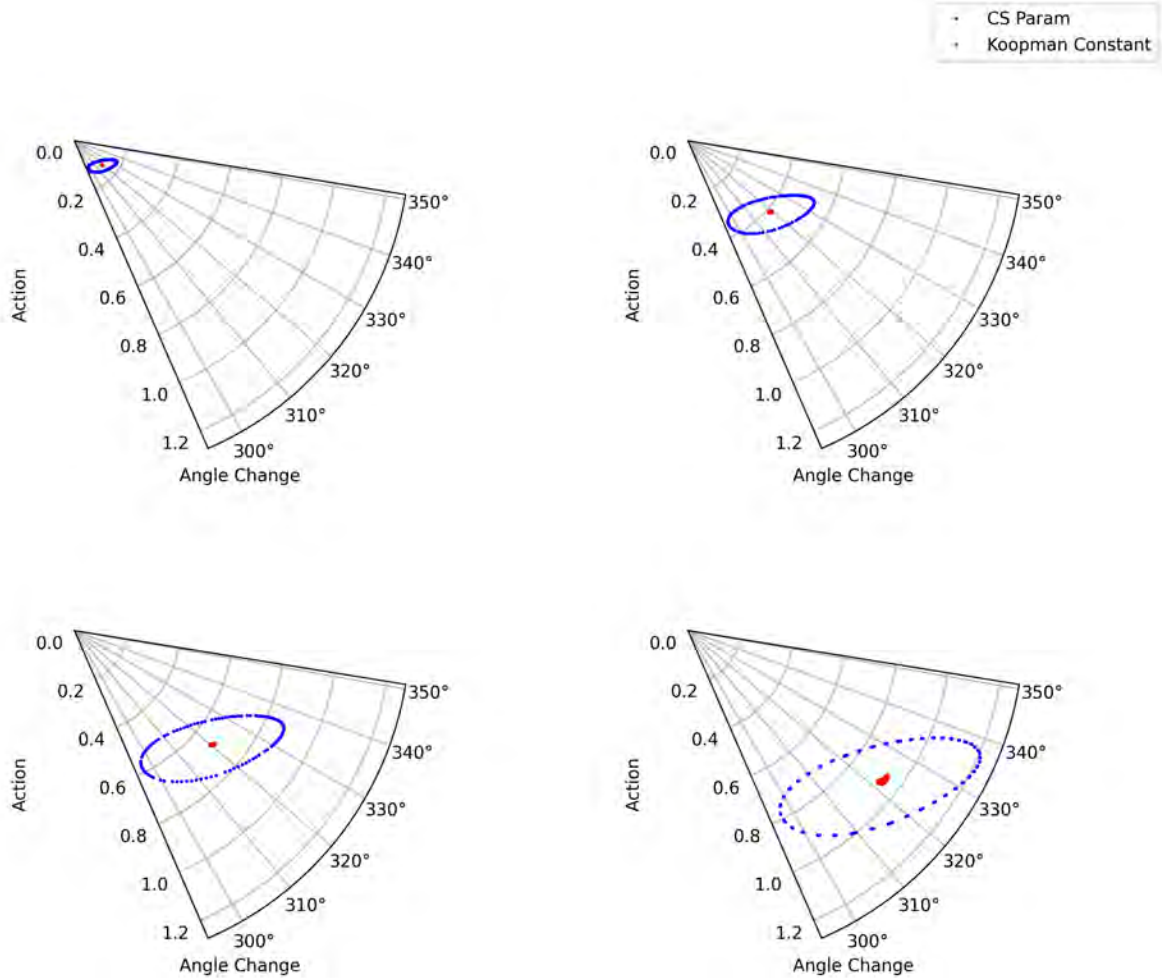


Figure 4.6 Orbits of the standard map plotted by action vs change in phase comparing the C-S formalism (blue) to the Koopman subspace (red).

4.7 showing those same points collectively for the C-S and the encoder in each plot.

## 4.5 McMillan Map

For the McMillan map, the network had the same structure as the standard map. The parameters for the map were  $\lambda = 1$  and  $\epsilon = 0.1$ , and the model was trained on 15 orbits with  $q \in [10^{-3}, 1.2]$  for  $2 \times 10^9$  iterations. Figure 4.8 (a) shows the original orbits in the phase space and (b) shows the same orbits after passing through the encoder. We see similar reductions in the nonlinearities in the McMillan map as we did for the Standard map. However, the model did not do as well for the McMillan as it did for the standard map.

This could be due to the sharper angles in the phase space of the McMillan map, compared to

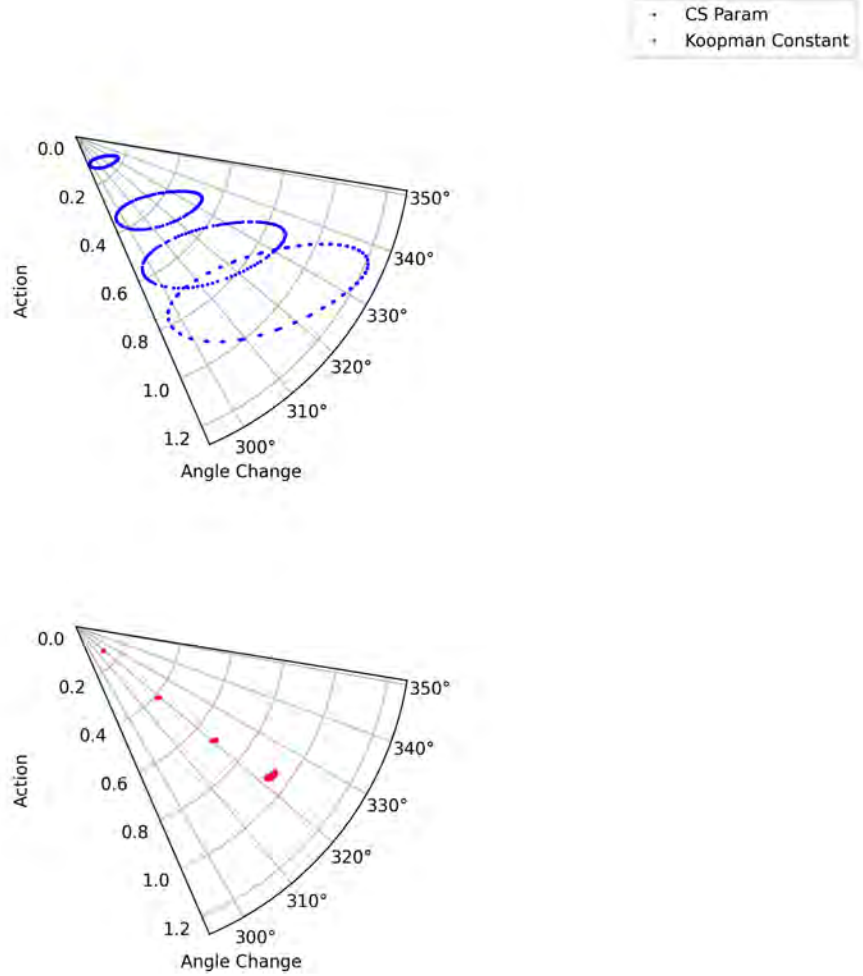


Figure 4.7 Orbits of the standard map plotted by action vs change in phase in the C-S formalism (blue) and the Koopman subspace (red).

the smoother, more elliptical shapes of the Standard map as shown in Figure 4.2 (a) and 4.8 (a). We can also see in Figure 4.13 that the C-S parameters seem to fold over and create a concave feature that gets more exaggerated at larger amplitudes. The model may have more difficulty treating irregular features such as this. It is possible that more layers would be needed to resolve features such as these.

#### 4.6 Frequency Prediction

The frequency model is a traditional network. In order to train the frequency prediction and operator transformation at the same time, we first assume that the tune is the linear tune and does not change with amplitude. This assumption is fairly accurate at smaller amplitudes. The

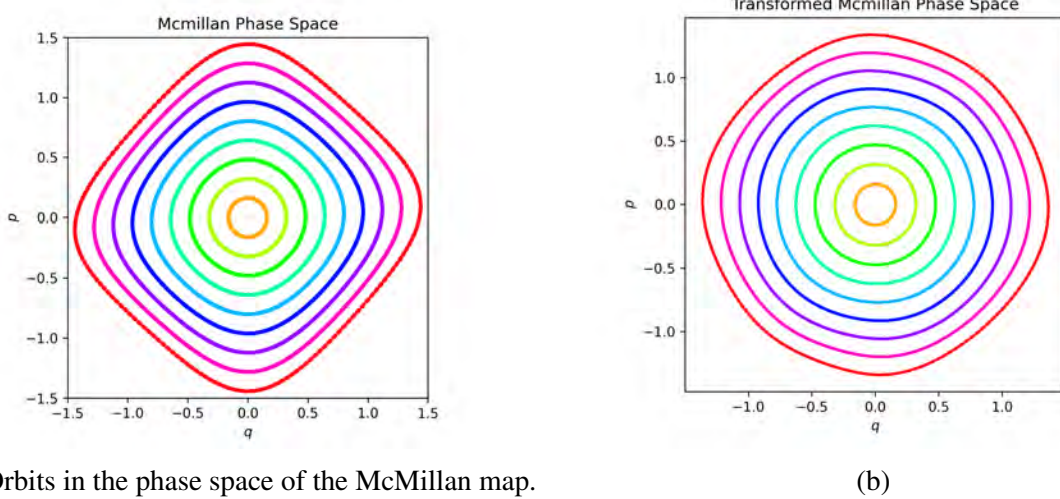


Figure 4.8 The phase space of the McMillan map (a) and the same phase space after being passed through the encoder to the Koopman subspace (b).

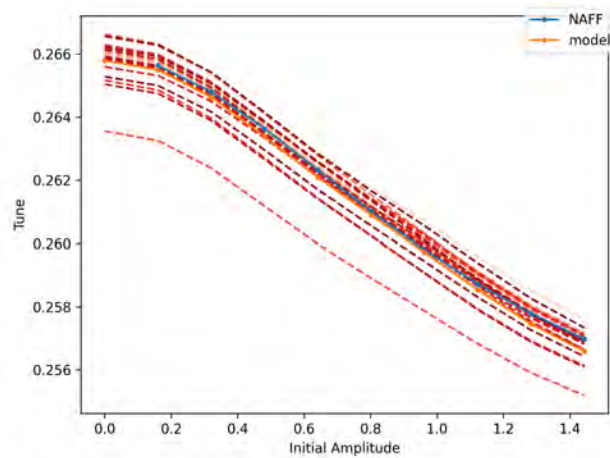


Figure 4.9 The tune shift with amplitude for the McMillan map calculated from NAFF (blue) and the model (orange). In red are the model's predictions during previous epochs, getting lighter the older the epoch.

first third of epochs, when training the model, assumes a constant tune. This allows the coordinate transformation to get closer to the desired solution and be closer to the identity at smaller amplitudes. Afterward, we allow the frequency model to begin to make changes to the tune; there is no change to the loss function. The frequency model will predict a tune shift and add that to the linear tune in the training. This network structure predicted a tune shift with amplitude close to that calculated by turn-by-turn tracking and NAFF. However, unlike previous versions of this network, as discussed at IPAC 2024, this model did not use any NAFF data [26]. This could suggest that a

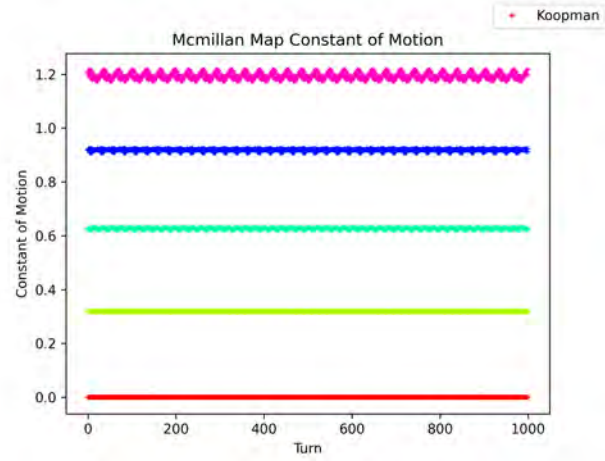


Figure 4.10 The amplitude of the motion of the standard map in the Koopman subspace.

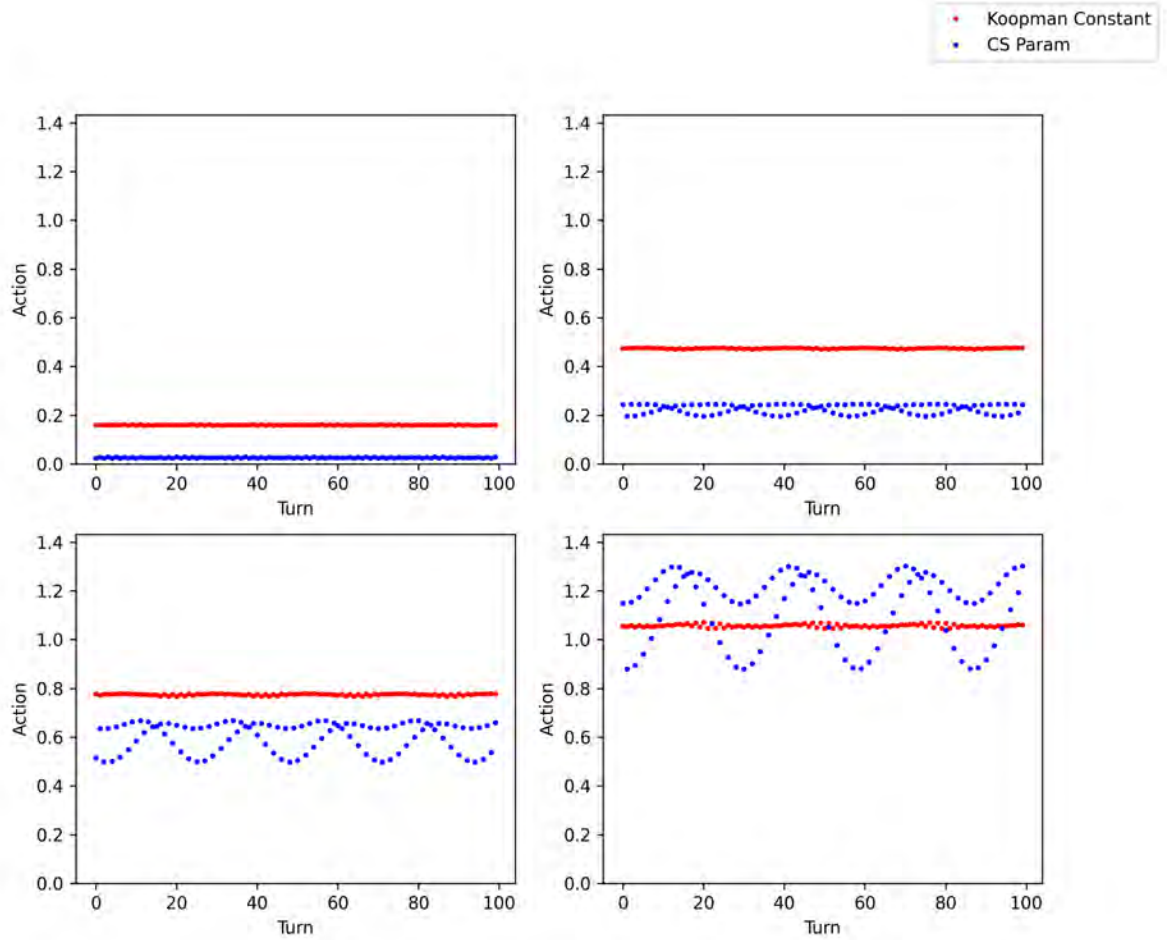


Figure 4.11 The amplitude of the motion of the standard map in the C-S formalism (blue) vs the same orbits in the Koopman subspace (red) plotted by turn.

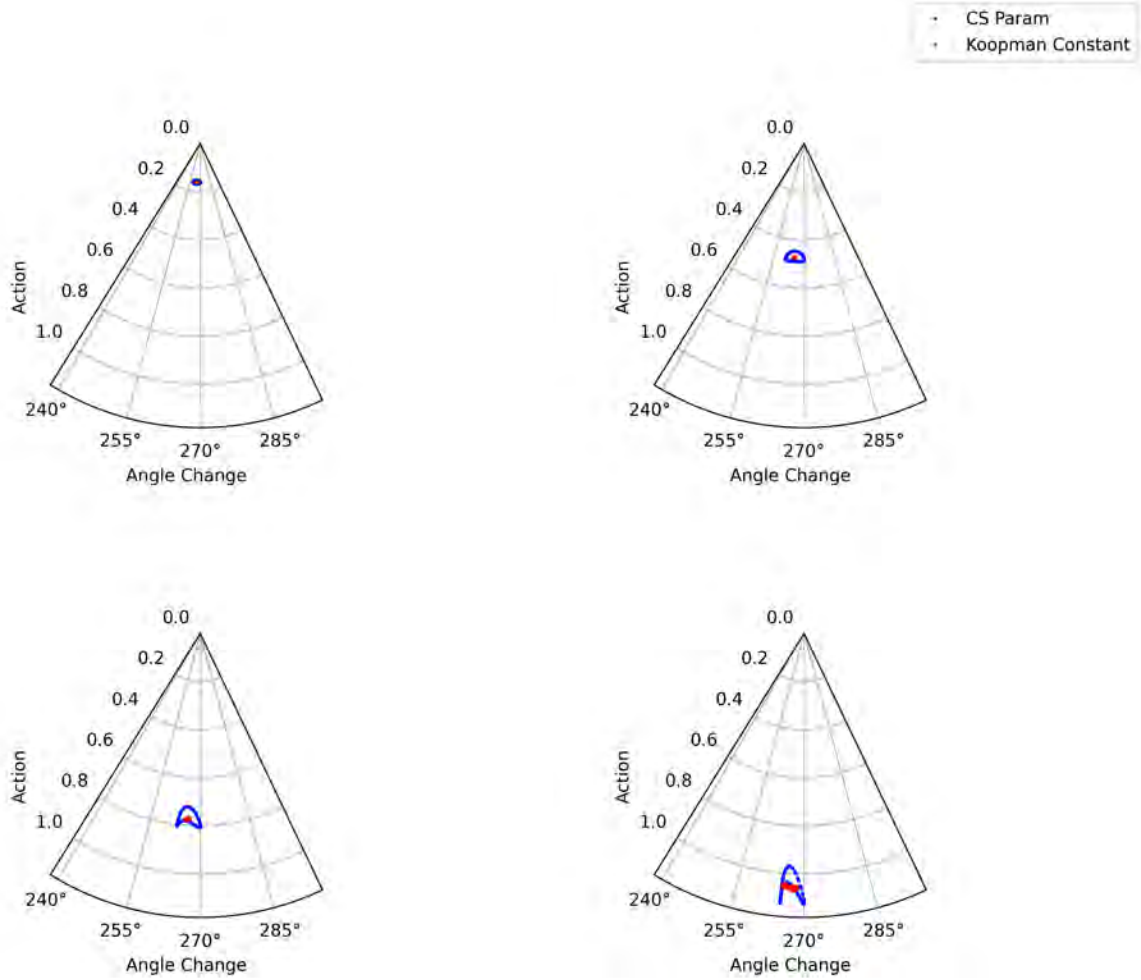


Figure 4.12 Orbits of the standard map plotted by action vs change in phase comparing the C-S formalism (blue) to the Koopman subspace (red).

4-D implementation of this model could find a similar tune map as NAFF.

While the model produces a similar curve and value to the NAFF data when looking at average predictions over a single orbit, it shows increasing variance as the amplitude grows (Figures 4.9 and 4.3). This makes sense in that the nonlinearities increase with amplitude, so should the instabilities in the tune, like we see in FMA. It is also possible that middle amplitudes have the benefit of data on both sides of their amplitude, whereas the larger portions only have data below their orbits. So, one might expect that these variances would decrease if pushed to larger amplitudes with more orbits.

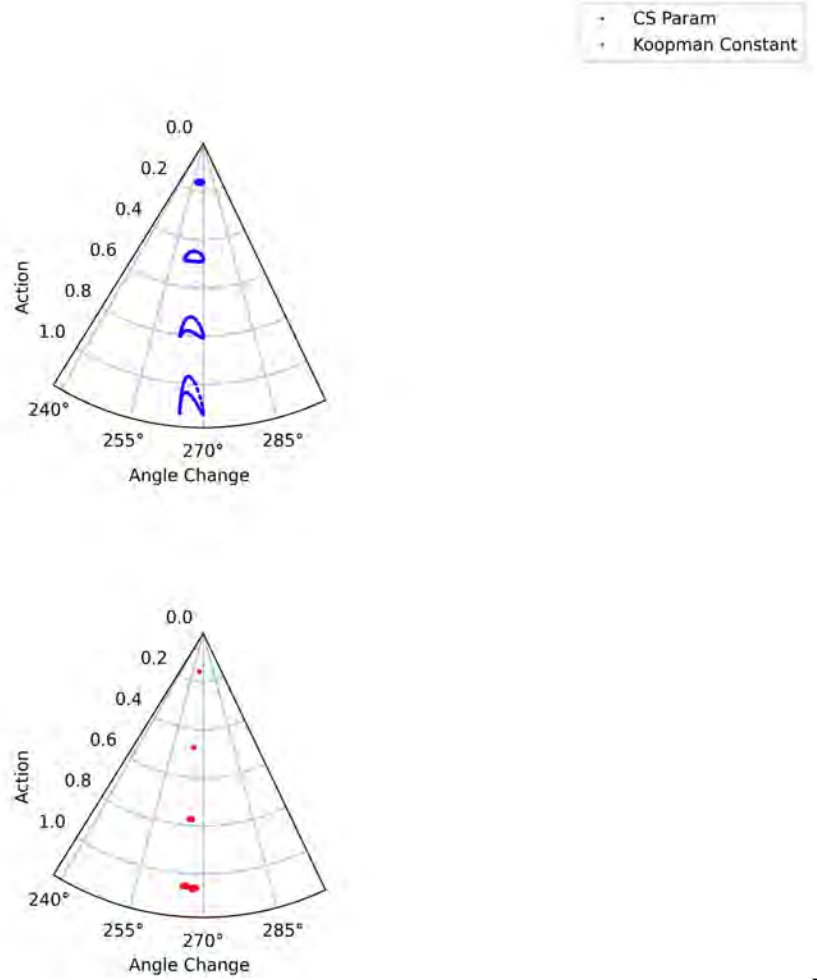


Figure 4.13 Orbits of the standard map plotted by action vs change in phase in the C-S formalism (blue) and the Koopman subspace (red).

#### 4.7 Errors in Tracking

With the data that can be collected from beam position monitors in accelerators, it is natural to ask if this method could withstand errors in the measurements of the particle positions in the phase space. Up to this point, we have used toy models where the accuracy of our tracking is known to machine precision.

To begin to answer this question, we can introduce random errors to our training and test data. Random errors from a Gaussian distribution centered at zero with a standard deviation of 1 were selected, then multiplied by  $10^{-1}$  and added to the data for the standard map before training as previously described.



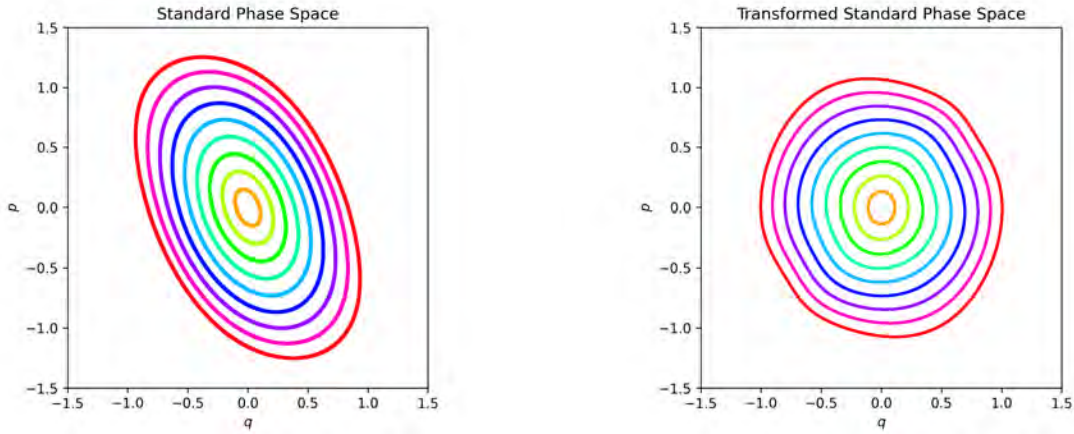


Figure 4.14 The phase space of the Standard map (a) and the same phase space after being passed through the encoder to the Koopman subspace in the noisy model (b).

Unsurprisingly, this leads to larger variances in the constant as shown in Figure 4.16. However, these are still more linear than the C-S parameters, which we can see in Figures 4.17, 4.18, and 4.19. Figure 4.15 shows the main issue: the model tune shift no longer matches the curve from NAFF. However, it mostly doesn't match at the ends of the curve, so this could be the middle of the curve, benefiting from having data on both sides of its amplitudes, like we discussed before. So this could be mitigated with more data at larger amplitudes. An important note to make for this study, though, is that while the data is noisy, the C-S parameters are essentially known. Errors when calculating the C-S parameters from BPM data could increase the errors. Nevertheless, this seems to show that the solution gained from the model can be robust to errors and more likely to converge on some fundamental properties of the system.

## 4.8 Discussion

What we have shown is that eigenfunctions of the Koopman operator can be calculated from nonlinear dynamical systems using symplectic neural networks. The use of SympNets in the neural networks reduces the number of parameters needed, which reduces the time and computations required to train the network. The eigenvalues of these eigenfunctions contain the betatron tunes of the dynamical systems, which show similar tune shifts with amplitude as NAFF does. Future work needs to be done to find how many parameters need to be used to cover a larger area, as well as account for amplitude dependencies in the transformation. It appears that while this number of

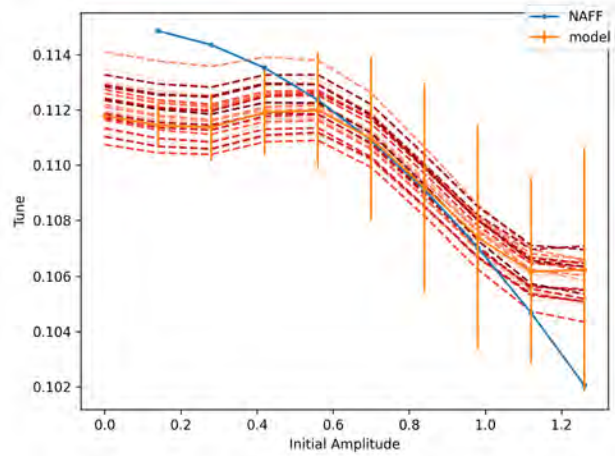


Figure 4.15 The tune shift with amplitude calculated from NAFF (blue) and the model (orange). In red are the noisy model's c

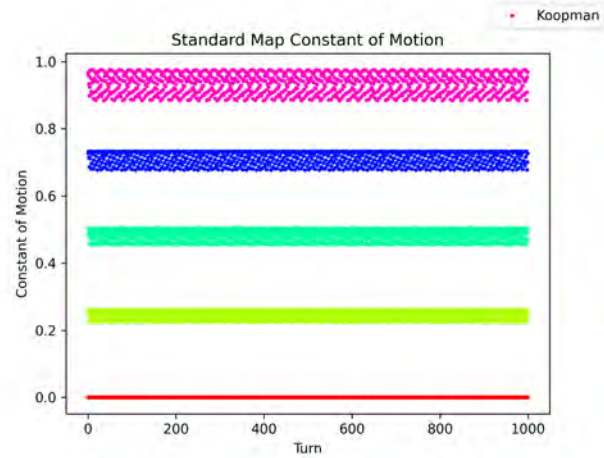


Figure 4.16 The amplitude of the motion of the standard map in the noisy Koopman subspace.

parameters may be adequate to predict a small range of amplitudes, a more complicated network would be needed to cover a larger area of amplitudes.



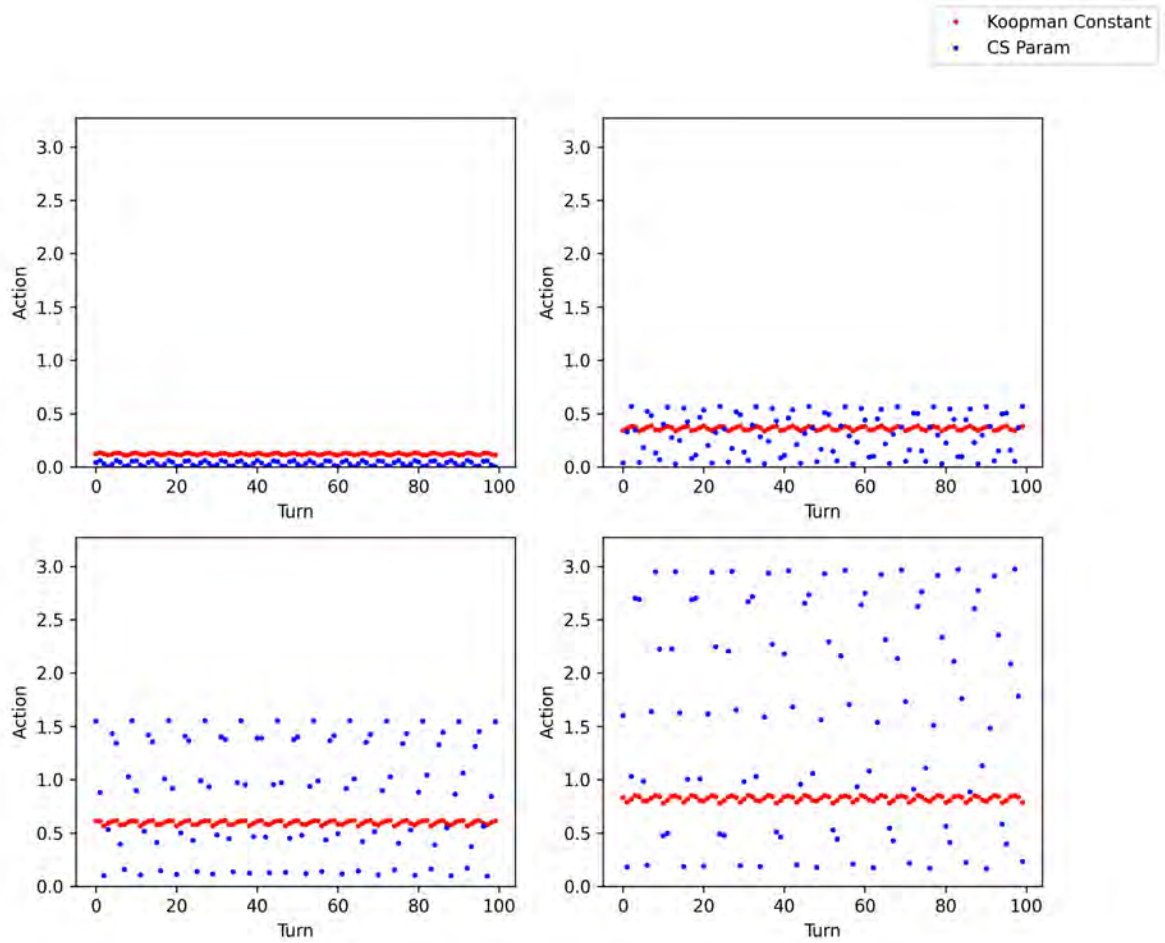


Figure 4.17 The amplitude of the motion of the standard map in the C-S formalism (blue) vs the same orbits in the noisy Koopman subspace (red) plotted by turn.

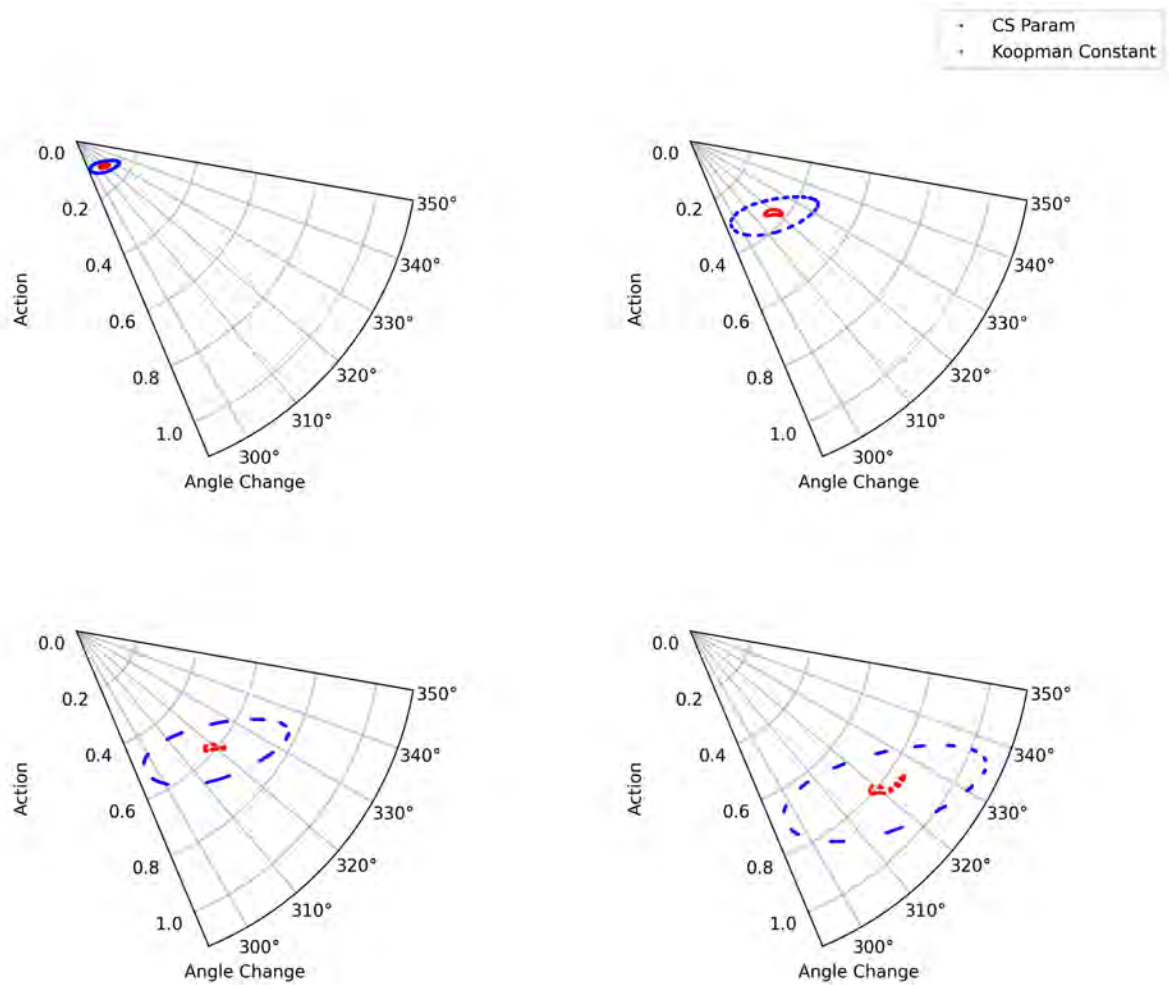


Figure 4.18 Orbits of the standard map plotted by action vs change in phase comparing the C-S formalism (blue) to the noisy Koopman subspace (red).

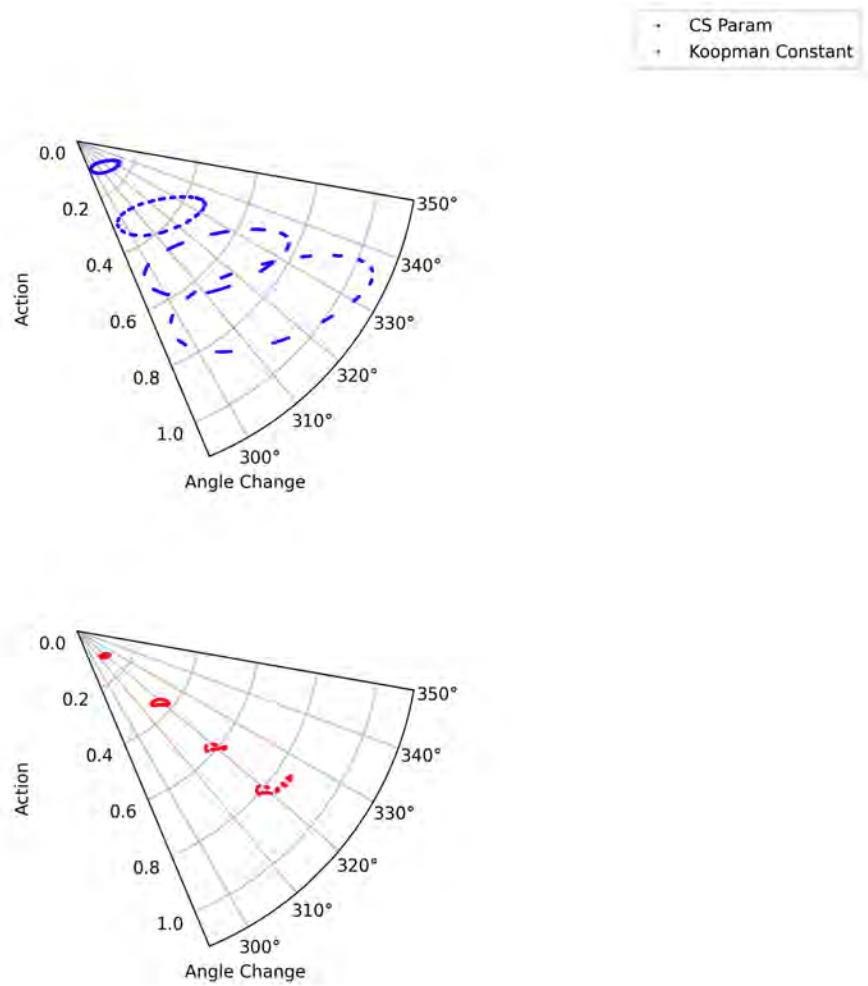


Figure 4.19 Orbits of the standard map plotted by action vs change in phase in the C-S formalism (blue) and the noisy Koopman subspace (red).

## CHAPTER 5

### CONCLUSIONS AND FUTURE WORK

#### 5.1 Problem Summary

The purpose of this thesis was to study the long-term dynamics of nonlinear systems in particle accelerators. For 2-D Linear Hamiltonian systems, we can find action-angle variables through Courant-Snyder parameters. In the 4-D case, we begin to encounter linear coupling, which can be treated through Edward-Teng parameterization. In accelerators, these describe systems that use dipole magnets and quadrupole magnets. When we introduce sextupole magnets, we enter nonlinear dynamics. These nonlinear dynamics introduce resonance and chaotic effects that can cause instability and beam loss during the operation of circular accelerators.

We can attempt to describe the system's dynamics using Poincaré maps and perform brute-force, turn-by-turn particle tracking to find these instabilities. This can work for beams with strong damping from synchrotron radiation, such as electron storage rings. But systems without that, such as hadron rings, can be lost after millions of turns or longer. This creates an unfeasible amount of computations needed to calculate the long position. It is therefore necessary that we develop methods to analyze the long-term information from these nonlinear systems.

We discussed some treatments for these, such as perturbation theory, Lie operators, and normal form. These perturbative methods can be helpful but require tedious order-by-order calculations, and any changes to the lattice would require them to be redone. We can instead leverage some turn-by-turn tracking with frequency map analysis (FMA). This method uses a number of turns to find the frequencies of particles at the beginning and end of the tracking using the numerical analysis of fundamental frequencies algorithm (NAFF). KAM theory tells us that for linear Hamiltonian systems, particle motion will be bounded to the surface of a torus and have stable frequencies that trace them out. But when perturbations are applied to that system, some of the tori will be destroyed and create chaotic regions, but others will be warped and survive. We can then use the stabilities of the frequencies calculated to gain insight into the long-term stability of the particle; unstable frequencies could imply unstable orbits. The FMA maps are helpful to find resonances

in the phase space and chaotic regions, but they still rely on turn-by-turn tracking, which can have compounding errors if the map is inaccurate and possibly falsely identify stable regions as unstable. It is worthwhile to identify other methods to gain this long-term information from a system.

## **5.2 Conclusions and Future Work**

### **5.2.1 Iterative Method**

The iterative method developed provides such a possible method [2]. The method starts with a sample of particles that form a circle in phase space. We assume a transformation that can change these coordinates from the real phase space into one where the dynamics are that of rigid rotations by constant frequencies (Equations 3.38 and 3.39). That transformation is calculated by examining how that transformation would change after one turn (Equations 3.40 and 3.41) and relating each of those terms to the map (Equations 3.44 and 3.45). Then the initial condition determines the 0th-order component. This can be repeated until it converges to a solution. The convergence is then used as an indicator to create convergence maps similar to FMA, see Figures 3.9 and 3.10. This thesis described how we expanded this method from 2-D phase spaces to 4-D phase spaces. This needed to take into account how the transformations now had multiple variables and the coordinates were coupled.

We used action angle variables from the Square Matrix Method (SMM) to smooth the map on the RHS of equations 3.40 and 3.41. We can see, comparing this to Figure 3.5 to Figure 3.6, that this is a great improvement from using the square matrix on its own. We can see from Figure 3.8 that the SMM also improves the area of convergence for the iterative method.

We can see that by comparing the convergence maps to their FMA counterparts, we see similar resonance structures. We also see that the convergence method can be sensitive to resonances because of the small denominator problem seen in Equations 3.44 and 3.45. This shows that the convergence map is a practical tool that can be used to identify resonances and phase space for dynamic aperture calculations like the frequency method.

Future work would be to expand this method to a full 6-D phase space. This would introduce new challenges to include the longitudinal dynamics, as we would encounter more synchro betatron

resonances. Further work could also be done applying the method to more practical maps to further describe what phase space features can be found with the method.

### 5.2.2 Machine Learning Methods

With beam position monitors in accelerators, it is possible to collect particle tracking data from accelerators. NAFF can be used to find frequencies in this data and create frequency maps, but we could also leverage this data in machine learning models to find frequencies and constants of motion. One approach is the idea of the Koopman operator. It was shown by Koopman in 1931 that observables of a nonlinear Hamiltonian dynamical system can be described by the evolution of a linear operator, the Koopman operator, in an infinite-dimensional phase space. It is trading the finite nonlinear dynamical system for a linear infinite-dimensional one. The difficulty is finding a transformation to this space and describing it in a useful but truncated form. An idea to do that is to find eigenfunctions of the Koopman operator. The nature of eigenfunctions would mean that the evolution of the eigenfunctions would remain in a finite subspace of the Koopman operator phase space and evolve by multiplying a constant eigenvalue. We then aim in our systems to find an eigenfunction whose eigenvalue corresponds to a rigid rotation in the Koopman operator phase space. The amplitude of these functions would be an invariant, and the frequencies would correspond to the fundamental frequencies of our system. Not all systems have invariants, however, as we have shown, approximate invariants can be found in local regions of the phase space of these systems in some cases.

The transformation to this eigenfunction is described as an encoder; the loss functions are structured to capture a few properties. One being that the transformation is symplectic, meaning it preserves area and Hamiltonian equations of motion. This would also ensure that any invariants are preserved in the transformation. They also must ensure that the dynamics after the encoder are linear. Figure 4.1 shows an example of the structure of our network for this case.

Current approaches also require the models to train separate parameters for the decoder that inverts the encoder. What we have shown is that by using symplectic neural networks (SympNets) we can improve the efficiency when training these networks. The SympNets naturally preserve the

symplectic condition, and their inverse is exactly known. By using these in our encoder, we remove the need for a separate decoder to train, we remove the need for the symplectic condition to be included in the loss function, and we require fewer parameters than a traditional network. These factors improve the efficiency of training the networks.

We have shown that this method is able to find transformations to approximately linear spaces in the in an integrable system, the Octupole McMellan map, see Figure 4.8 as well as find a similar action-dependent tune shift as NAFF, see Figure 4.9. We have also shown that this method is able to find transformations to approximately linear spaces in a non-integrable system, the standard map, see Figure 4.2, as well as find a similar action-dependent tune shift as NAFF, see Figure 4.3. Future work is needed to expand the study to 4-D and 6-D phase spaces. There could also be more work done in finding the minimum number of parameters needed to approximate these eigenfunctions.

## BIBLIOGRAPHY

- [1] P. Cincotta, “Arnold diffusion: An overview through dynamical astronomy,” *New Astronomy Reviews*, vol. 46, pp. 13–39, Jan. 2002. doi: 10.1016/S1387-6473(01)00153-1.
- [2] L. H. Yu, Y. Hidaka, V. Smaluk, K. Anderson, and Y. Hao, “Convergence map with action-angle variables based on square matrix for nonlinear lattice optimization,” *Phys. Rev. Accel. Beams*, vol. 26, no. 5, p. 054 002, May 2023, Publisher: American Physical Society. doi: 10.1103/PhysRevAccelBeams.26.054002. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevAccelBeams.26.054002>.
- [3] B. Lusch, J. N. Kutz, and S. L. Brunton, “Deep learning for universal linear embeddings of nonlinear dynamics,” *Nature Communications*, vol. 9, no. 1, Nov. 2018, Publisher: Springer Science and Business Media LLC, ISSN: 2041-1723. doi: 10.1038/s41467-018-07210-0. [Online]. Available: <http://dx.doi.org/10.1038/s41467-018-07210-0>.
- [4] P. Jin, Z. Zhang, A. Zhu, Y. Tang, and G. E. Karniadakis, “SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems,” *Neural Networks*, vol. 132, pp. 166–179, 2020, ISSN: 0893-6080. doi: <https://doi.org/10.1016/j.neunet.2020.08.017>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608020303063>.
- [5] S. Navas *et al.*, “Review of particle physics,” *Phys. Rev. D*, vol. 110, no. 3, p. 030 001, 2024. doi: 10.1103/PhysRevD.110.030001.
- [6] S. Lee, *Accelerator Physics (Fourth Edition)*. World Scientific Publishing Company, 2018, ISBN: 978-981-327-469-3. [Online]. Available: <https://books.google.com/books?id=UvJ9DwAAQBAJ>.
- [7] J. Pöschel, *A lecture on the classical KAM theorem*, \_eprint: 0908.2234, 2009. [Online]. Available: <https://arxiv.org/abs/0908.2234>.
- [8] G. Jackson, “The Fermilab Recycler Ring Technical Design Report: Rev. 1.2,” Fermi National Accelerator Laboratory (FNAL), Batavia, IL (United States), Tech. Rep., Oct. 1996. doi: 10.2172/16029. [Online]. Available: <https://www.osti.gov/biblio/16029>.
- [9] E. D. Courant and H. S. Snyder, “Theory of the Alternating-Gradient Synchrotron,” *Annals of Physics*, vol. 281, no. 1, pp. 360–408, 2000, ISSN: 0003-4916. doi: <https://doi.org/10.1006/aphy.2000.6012>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0003491600960123>.
- [10] D. A. Edwards and L. C. Teng, “Parametrization of Linear Coupled Motion in Periodic Systems,” *IEEE Transactions on Nuclear Science*, vol. 20, no. 3, pp. 885–888, 1973. doi: 10.1109/TNS.1973.4327279.



- [11] I. Stewart, *Chapter 5: Perturbation Theory*, Published: Lecture notes for 8.09 Classical Mechanics III, Fall 2014, 2014. [Online]. Available: [https://ocw.mit.edu/courses/8-09-classical-mechanics-iii-fall-2014/resources/mit8\\_09f14\\_chapter\\_5/](https://ocw.mit.edu/courses/8-09-classical-mechanics-iii-fall-2014/resources/mit8_09f14_chapter_5/).
- [12] D. Rothman, *Lectures 15–16: Poincare Sections*, Published: Lecture notes for 12.006J Nonlinear Dynamics: Chaos, Fall 2022, 2022. [Online]. Available: [https://ocw.mit.edu/courses/12-006j-nonlinear-dynamics-chaos-fall-2022/resources/mit12\\_006jf22\\_lec15-16/](https://ocw.mit.edu/courses/12-006j-nonlinear-dynamics-chaos-fall-2022/resources/mit12_006jf22_lec15-16/).
- [13] W. Commons, *File:Poincare map.svg — Wikimedia Commons, the free media repository*, 2025. [Online]. Available: [https://commons.wikimedia.org/w/index.php?title=File:Poincare\\_map.svg&oldid=1005183475](https://commons.wikimedia.org/w/index.php?title=File:Poincare_map.svg&oldid=1005183475).
- [14] M. Hénon, “A two-dimensional mapping with a strange attractor,” *Communications in Mathematical Physics*, vol. 50, no. 1, pp. 69–77, 1976, Publisher: Springer.
- [15] J. Laskar, “The chaotic motion of the solar system: A numerical estimate of the size of the chaotic zones,” *Icarus*, vol. 88, no. 2, pp. 266–291, 1990, ISSN: 0019-1035. DOI: [https://doi.org/10.1016/0019-1035\(90\)90084-M](https://doi.org/10.1016/0019-1035(90)90084-M). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/001910359090084M>.
- [16] J. Laskar, C. Froeschlé, and A. Celletti, “The measure of chaos by the numerical analysis of the fundamental frequencies. Application to the standard mapping,” *Physica D: Nonlinear Phenomena*, vol. 56, no. 2, pp. 253–269, 1992, ISSN: 0167-2789. DOI: [https://doi.org/10.1016/0167-2789\(92\)90028-L](https://doi.org/10.1016/0167-2789(92)90028-L). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/016727899290028L>.
- [17] J. Laskar, “Introduction to Frequency Map Analysis,” in *Hamiltonian Systems with Three or More Degrees of Freedom*, C. Simó, Ed., Dordrecht: Springer Netherlands, 1999, pp. 134–150, ISBN: 978-94-011-4673-9. DOI: 10.1007/978-94-011-4673-9\_13. [Online]. Available: [https://doi.org/10.1007/978-94-011-4673-9\\_13](https://doi.org/10.1007/978-94-011-4673-9_13).
- [18] Y. Papaphilippou, “Detecting chaos in particle accelerators through the frequency map analysis method,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 24, no. 2, p. 024412, Jun. 2014, ISSN: 1054-1500. DOI: 10.1063/1.4884495. [Online]. Available: <https://doi.org/10.1063/1.4884495>.
- [19] A. W. Chao, “Lecture Notes on Topics in Accelerator Physics,” Stanford Linear Accelerator Center, Menlo Park, CA (US), Tech. Rep., Nov. 2002. DOI: 10.2172/812598. [Online]. Available: <https://www.osti.gov/biblio/812598>.
- [20] K. J. Anderson, Y. Hao, and L. H. Yu, “Study of Nonlinear Dynamics in the 4-D Hénon Map Using the Square Matrix Method and Iterative Methods,” in *Proc. 5th Int. Particle Accel. Conf. (NAPAC’22)*, ser. International Particle Accelerator Conference, ISSN: 2673-7000 Issue: 5, JACoW Publishing, Geneva, Switzerland, Oct. 2022, pp. 232–235, ISBN: 978-3-95450-232-5. DOI: 10.18429/JACoW-NAPAC2022-MOPA81. [Online]. Available:

<https://jacow.org/napac2022/papers/mopa81.pdf>.

- [21] L. H. Yu, “Analysis of nonlinear dynamics by square matrix method,” *Phys. Rev. Accel. Beams*, vol. 20, no. 3, p. 034 001, Mar. 2017, Publisher: American Physical Society. doi: 10.1103/PhysRevAccelBeams.20.034001. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevAccelBeams.20.034001>.
- [22] A. Meurer, C. P. Smith, M. Paprocki, *et al.*, “SymPy: Symbolic computing in Python,” *PeerJ Computer Science*, vol. 3, e103, Jan. 2017, issn: 2376-5992. doi: 10.7717/peerj-cs.103. [Online]. Available: <https://doi.org/10.7717/peerj-cs.103>.
- [23] Y. Hao, K. J. Anderson, and L. H. Yu, “Revisit of Nonlinear Dynamics in Hénon Map Using Square Matrix Method,” in *Proc. IPAC’21*, ser. International Particle Accelerator Conference, ISSN: 2673-5490 Issue: 12, JACoW Publishing, Geneva, Switzerland, Aug. 2021, pp. 3788–3791, isbn: 978-3-95450-214-1. doi: 10.18429/JACoW-IPAC2021-THP AB016. [Online]. Available: <https://jacow.org/ipac2021/papers/thpab016.pdf>.
- [24] K. Anderson and Y. Hao, “Dynamics study of the crab crossing at the electron ion collider using square matrix and iterative methods,” English, in *Proc. IPAC’24*, ser. IPAC’24 - 15th International Particle Accelerator Conference, ISSN: 2673-5490 Issue: 15, JACoW Publishing, Geneva, Switzerland, May 2024, pp. 717–720, isbn: 978-3-95450-247-9. doi: 10.18429/JACoW-IPAC2024-MOPS11. [Online]. Available: <https://indico.jacow.org/event/63/contributions/3626>.
- [25] *NSLS-II Accelerator Parameters*. [Online]. Available: <https://www.bnl.gov/nsls2/accelerator/docs/accelerator-parameters.pdf> (visited on 05/01/2025).
- [26] K. Anderson and Y. Hao, “Koopman operator method for nonlinear dynamics analysis using symplectic neural networks,” English, in *Proc. IPAC’24*, ser. IPAC’24 - 15th International Particle Accelerator Conference, ISSN: 2673-5490 Issue: 15, JACoW Publishing, Geneva, Switzerland, May 2024, pp. 713–716, isbn: 978-3-95450-247-9. doi: 10.18429/JACoW-IPAC2024-MOPS10. [Online]. Available: <https://indico.jacow.org/event/63/contributions/3638>.
- [27] R. B. Palmer, “Energy scaling, crab crossing and the pair problem,” Stanford Linear Accelerator Center, Menlo Park, CA (USA), Nov. 1988. [Online]. Available: <https://www.osti.gov/biblio/6522191>.
- [28] D. Xu, Y. Hao, Y. Luo, and J. Qiang, “Synchrotron resonance of crab crossing scheme with large crossing angle and finite bunch length,” *Phys. Rev. Accel. Beams*, vol. 24, no. 4, p. 041 002, Apr. 2021, Publisher: American Physical Society. doi: 10.1103/PhysRevAccelBeams.24.041002. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevAccelBeams.24.041002>.
- [29] S. Kostoglou, N. Karastathis, Y. Papaphilippou, D. Pellegrini, and P. Zisopoulos, “Develop-

- ment of Computational Tools for Noise Studies in the LHC,” in *Proc. of International Particle Accelerator Conference (IPAC’17), Copenhagen, Denmark, 14 - 19 May, 2017*, ser. International Particle Accelerator Conference, Issue: 8, Geneva, Switzerland: JACoW, May 2017, pp. 3807–3810, ISBN: 978-3-95450-182-3. DOI: <https://doi.org/10.18429/JACoW-IPAC2017-THPAB044>. [Online]. Available: <http://jacow.org/ipac2017/papers/thpab044.pdf>.
- [30] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, *Modern Koopman Theory for Dynamical Systems*, \_eprint: 2102.12086, 2021. [Online]. Available: <https://arxiv.org/abs/2102.12086>.
  - [31] B. O. Koopman, “Hamiltonian Systems and Transformation in Hilbert Space,” *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931, \_eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.17.5.315>. DOI: 10.1073/pnas.17.5.315. [Online]. Available: <https://www.pnas.org/doi/abs/10.1073/pnas.17.5.315>.
  - [32] T. Zolkin, S. Nagaitsev, and I. Morozov, *McMillan map and nonlinear Twiss parameters*, \_eprint: 2204.12691, 2022. [Online]. Available: <https://arxiv.org/abs/2204.12691>.
  - [33] A. Paszke, S. Gross, F. Massa, *et al.*, *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, \_eprint: 1912.01703, 2019. [Online]. Available: <https://arxiv.org/abs/1912.01703>.