# A Fast Tracking Method using Resonance Basis Hamiltonians

J. Irwin, T. Chen and Y. T. Yan\*

Stanford Linear Accelerator Center, Stanford, CA 94309

**Abstract.** A resonance basis map is an important way of analyzing lattice properties in circular accelerators. A method is developed to do tracking with resonance basis maps. The speed of this method is faster than other map tracking methods, and can be dramatically increased by dropping insignificant resonance terms. This mapping method enables us to simulate and study the interplay between lattice nonlinearities and other effects such as the beam-beam interaction.

# I. INTRODUCTION

Mapping techniques using Lie algebra and truncated power series algebra have revolutionized the study of accelerator lattice nonlinearities in many ways. Simulations with nonlinear one-turn maps can be much faster than element-byelement tracking so that one can study long term stability on large hadron machines<sup>[1]</sup>. On electron machines, the requirement is different. Because of radiation effects, symplecticity is not as important as in hadron machines. However, to study multi-particle effects such as the beam-beam interaction, fast tracking methods are also needed.

Maps may also be used to analyze the lattice. In circular machines, resonances play an important role in most machine physics. With maps, one can obtain the driving terms of each resonance of the lattice<sup>[2]</sup>. It would be very useful to track with an explicit resonance form, so that the interaction between lattice nonlinearities and other effects can be studied in the resonance basis.

In this paper, we present a tracking method of using resonance basis maps.

# **II. RESONANCE BASIS HAMILTONIAN**

The transverse one-turn map with energy deviation  $\delta$  as a parameter can be written as:

<sup>\*</sup> Work supported by the Department of Energy, contract DE-AC03-76SF00515.

 $M = \mathbf{A}\mathbf{R}e^{:-H:}\mathbf{A}^{-1}$ 

where  $\mathbf{A}$  is the similarity transform made of eigenvectors which bring the one-turn matrix into block diagonal form, and  $\mathbf{R}$  is the block diagonal rotation matrix. The Hamiltonian is defined as:

$$-H = \sum_{\substack{\vec{m},\vec{n},n_{\delta}\\\vec{n}-\vec{m}=\text{even}}}^{NO} \{a_{\vec{m},\vec{n},n_{\delta}} \sqrt{2J_{x}}^{n_{x}} \sqrt{2J_{y}}^{n_{y}} \delta^{n_{\delta}} \cos(m_{x}\theta_{x} + m_{y}\theta_{y})$$
$$+b_{\vec{n},\vec{n},n_{\delta}} \sqrt{2J_{x}}^{n_{x}} \sqrt{2J_{y}}^{n_{y}} \delta^{n_{\delta}} \sin(m_{x}\theta_{x} + m_{y}\theta_{y}) \}$$
$$+\sum_{\substack{\vec{n}=\text{even}\\n_{\delta}}}^{NO} c_{\vec{n},n_{\delta}} \sqrt{2J_{x}}^{n_{x}} \sqrt{2J_{y}}^{n_{y}} \delta^{n_{\delta}}$$

where  $n_i \ge m_i$ , (i = x, y) and  $n_i \cdot m_i$  must be even. Once a one-turn map is obtained, it may, by a linear transformation, be expressed in a resonance basis, thereby obtaining the coefficients, a, b, and c.

To evaluate the nonlinear map, we expand  $e^{:-H}$ : into the Poisson bracket series:

$$e^{:-H:}z = z - [H,z] + \frac{1}{2}[H,[H,z]] - \dots$$

where *z* can be an any one of *x*,  $p_x$ , *y*,  $p_y$ . For small amplitude particles, we will show the series may be truncated after the first Poisson bracket, which we call single Poisson bracket (SPB) tracking. When particles have large amplitudes, out to the dynamic aperture, we have found it sufficient to terminate the series at the second term, which we call double Poisson bracket (DPB) tracking.

The truncation of the Poisson bracket series results in the loss of symplecticity, but this is not an important issue in electron storage rings. The evaluation of DPB tracking is about twice as fast as truncated Taylor series tracking to the same order, if it contains all possible resonance terms. However the huge advantage of this method occurs when H does not contain all resonances. When studying dynamics for any particular working point in tune space, the range of significant tunes arising from tune-shift-with-amplitude or tune-shift-withenergy terms in the Hamiltonian (the third sum), including beam-beam, defines a known area in tune space. Resonances which do not intersect this area may be dropped from the Hamiltonian with savings of up to a factor of 10 in map evaluation time.

# **III. CALCULATING POISSON BRACKETS**

#### 1. Single Poisson Bracket

The calculation is simpler in action-angle variable space than  $x-p_x$  space, because the derivatives are easy to get and it is congruent with the choice of resonance basis.

In order to minimize CPU time a specific procedure is followed. The tracking starts from normal coordinates, *i.e.*, *x*, *p<sub>x</sub>*, *y*, *p<sub>y</sub>*,  $\tau$ ,  $\delta$ . The first step is to sum up the coefficients as a polynomial of energy deviation  $\delta$  to eliminate the index  $n_{\delta}$ :

$$a_{\vec{m},\vec{n}} = \sum_{n_{\delta}} a_{\vec{m},\vec{n},n_{\delta}} \delta^{n_{\delta}}, b_{\vec{m},\vec{n}} = \sum_{n_{\delta}} b_{\vec{m},\vec{n},n_{\delta}} \delta^{n_{\delta}}, c_{\vec{n}} = \sum_{n_{\delta}} c_{\vec{n},n_{\delta}} \delta^{n_{\delta}}.$$

Then, we define:

$$C_{\vec{m},\vec{n}} = \sqrt{2J_x}^{n_x} \sqrt{2J_y}^{n_y} \{a_{\vec{m},\vec{n}} \cos(m_x \theta_x + m_y \theta_y) + b_{\vec{m},\vec{n}} \sin(m_x \theta_x + m_y \theta_y)\}$$
  

$$S_{\vec{m},\vec{n}} = \sqrt{2J_x}^{n_x} \sqrt{2J_y}^{n_y} \{a_{\vec{m},\vec{n}} \sin(m_x \theta_x + m_y \theta_y) - b_{\vec{m},\vec{n}} \cos(m_x \theta_x + m_y \theta_y)\}$$
  

$$\tilde{J}_{\vec{n}} = \sqrt{2J_x}^{n_x} \sqrt{2J_y}^{n_y}$$
(for *n* even)

So, the Hamiltonian can be written as:

$$-H = \sum_{\substack{\vec{m},\vec{n}\\\vec{n}-\vec{m}=\text{even}}}^{NO} C_{\vec{m},\vec{n}} + \sum_{\vec{n}=\text{even}}^{NO} c_{\vec{n}} \tilde{J}_{\vec{n}}$$

It would be very time consuming to convert the *x*-*p*<sub>*X*</sub> coordinates into actionangle variables and calculate all the sine and cosine functions. To get around this, we can do the following: first, the coordinates are normalized by matrix **A**, and rotated by matrix **R**. After the **A** normalization, we know that  $x = \sqrt{2J_x} \cos \theta_x$ ,  $p_x = -\sqrt{2J_x} \sin \theta_x$ , and similar for *y*. One can then calculate all terms of  $\sqrt{2J_x}^{m_x} \sqrt{2J_y}^{m_y} \begin{cases} \sin \\ \cos \end{cases} (m_x \theta_x + m_y \theta_y)$  with add and multiplies by using trigonometric formulas. The actions may be calculated by  $2J_x = x^2 + p_x^2$ , and  $2J_y = y^2 + p_y^2$ . S in c e  $S_{\vec{m},\vec{n}}$  and  $C_{\vec{m},\vec{n}}$  c on s is t of  $(2J_x)^{\frac{n_x - m_x}{2}} (2J_y)^{\frac{n_y - m_y}{2}} \sqrt{2J_x}^{m_x} \sqrt{2J_y}^{m_y} \begin{cases} \sin \\ \cos \end{cases} (m_x \theta_x + m_y \theta_y)$ , and  $n_{x,y} - m_{x,y}$  are always even numbers, only simple multiplies of  $J_x$  and/or  $J_y$  are needed to get

always even numbers, only simple multiplies of  $J_x$  and/or  $J_y$  are needed to get these terms.

In the program, tables are created for each particle on each turn. The first table contains all the  $\sqrt{2J_x}^{m_x}\sqrt{2J_y}^{m_y}\begin{cases}\sin\\\cos\\cos\end{cases}(m_x\theta_x+m_y\theta_y)$  terms. They are generated directly from x,  $p_x$  and y,  $p_y$ . Another small table has  $(2J_x)^{lx}(2J_y)^{ly}$ , where  $l_{x,y}=(n_{x,y} - m_{x,y})/2$  are integers. With these three tables generated, it is quite efficient to calculate the nonlinear kicks in action-angle space.

Usually, we prefer tracking in the regular x- $p_x$  space. We can easily convert the nonlinear kicks from action-angle space by the normal canonical transformation using the following formula

$$[-H,x] = -\frac{\partial H}{\partial \theta_x} \frac{\partial x}{\partial J_x} + \frac{\partial H}{\partial J_x} \frac{\partial x}{\partial \theta_x} = -\frac{\partial H}{\partial \theta_x} \frac{x}{2J_x} + \frac{\partial H}{\partial J_x} p_x$$
$$[-H,p_x] = -\frac{\partial H}{\partial \theta_x} \frac{\partial p_x}{\partial J_x} + \frac{\partial H}{\partial J_x} \frac{\partial p_x}{\partial \theta_x} = -\frac{\partial H}{\partial \theta_x} \frac{p_x}{2J_x} - \frac{\partial H}{\partial J_x} x$$

and the same thing for y. In which

$$-\frac{\partial H}{\partial \theta_x} = \sum_{\vec{m},\vec{n}} m_x S_{\vec{m},\vec{n}}$$
$$-\frac{\partial H}{\partial \theta_y} = \sum_{\vec{m},\vec{n}} m_y S_{\vec{m},\vec{n}}$$

and

$$\frac{\partial H}{\partial J_x} = -(2J_x)^{-1} \left[ \sum_{\vec{m},\vec{n}} n_x C_{\vec{m},\vec{n}} + \sum_{\vec{n}=\text{even}} n_x c_{\vec{n}} \tilde{J}_{\vec{n}} \right]$$
$$\frac{\partial H}{\partial J_y} = -(2J_y)^{-1} \left[ \sum_{\vec{m},\vec{n}} n_y C_{\vec{m},\vec{n}} + \sum_{\vec{n}=\text{even}} n_y c_{\vec{n}} \tilde{J}_{\vec{n}} \right]$$

In the whole process, including calculating the tables, only simple add and multiply operations are need.

## 2. Longitudinal Motion

To do beam-beam studies, we need the one turn map evaluated at the interaction point. If the RF cavity were located at the IP, we could use the map to also calculate the one-turn  $\Delta \tau$  to evaluate the RF kick. Locating the RF at the IP, although physically impossible, in our experience seems to yield the same physics as the cavity located elsewhere in lattice. The time-of-flight is calculated as a linear function of the transverse coordinates and a high order nonlinear function of  $\delta$ . We do not include the effect of the nonlinear transverse motion on the time-of-flight. It is a very small effect.

When the RF cavity is located elsewhere, we could do two things: 1) use two nonlinear maps. One is from IP to the cavity, and the other is from the cavity to the IP through the rest of the ring. If we do this, two nonlinear maps are needed and the CPU time is doubled. 2) use one nonlinear map of the whole ring, from IP to IP, plus two time-of-flights and two linear  $\delta$ -dependent maps. One of the timeof-flights and one linear map is from the IP to the RF cavity, where the energy deviation  $\delta$  is updated by the RF cavity sinusoidal kick depending only on  $\tau$ . After updating  $\delta$ , we proceed through the rest of the ring with the other time-offlight and linear map. In most cases, the location of the RF cavities is not important to the particle dynamics.

## 3. Double Poisson Bracket Tracking

At large amplitude, *e. g.*, at the vicinity of dynamic aperture, the single Poisson bracket may not be able to achieve acceptable accuracy. Therefore, higher order terms in the expansion have to be included. The next term in the expansion is [H, [H, z]]. Tracking based on the truncation after this term is called Double Poisson Bracket (or DPB) tracking.

The second order Poisson bracket is expanded as:

$$[H,[H,x]] = \frac{\partial H}{\partial \theta_x} \frac{\partial [H,x]}{\partial J_x} - \frac{\partial H}{\partial J_x} \frac{\partial [H,x]}{\partial \theta_x} + \frac{\partial H}{\partial \theta_y} \frac{\partial [H,x]}{\partial J_y} - \frac{\partial H}{\partial J_y} \frac{\partial [H,x]}{\partial \theta_y}$$

and

$$[H, [H, p_x]] = \frac{\partial H}{\partial \theta_x} \frac{\partial [H, p_x]}{\partial J_x} - \frac{\partial H}{\partial J_x} \frac{\partial [H, p_x]}{\partial \theta_x} + \frac{\partial H}{\partial \theta_y} \frac{\partial [H, p_x]}{\partial J_y} - \frac{\partial H}{\partial J_y} \frac{\partial [H, p_x]}{\partial \theta_y}$$

Similar expressions are valid for the *y* derivatives. Expanding the above formulas, one can find that most of the quantities, such as  $\partial H/\partial J_x$  and  $\partial H/\partial \theta_x$ , have been calculated in SPB. The additional quantities which need to be calculated are the 10 second order derivatives:

$$\frac{\partial^2 H}{\partial J_x^2} = -(2J_x)^{-2} \left[ \sum_{\vec{m},\vec{n}} n_x (n_x - 2) C_{\vec{m},\vec{n}} + \sum_{\vec{n} = \text{even}} n_x (n_x - 2) c_{\vec{n}} \tilde{J}_{\vec{n}} \right]$$
$$\frac{\partial^2 H}{\partial \theta_x^2} = \sum_{\vec{m},\vec{n}} m_x^2 C_{\vec{m},\vec{n}}$$
$$\frac{\partial^2 H}{\partial J_x \partial \theta_x} = (2J_x)^{-1} \sum_{\vec{m},\vec{n}} n_x m_x S_{\vec{m},\vec{n}}$$

and the same for *y*, plus cross terms:

$$\frac{\partial^2 H}{\partial J_x \partial J_y} = -(2J_x \times 2J_y)^{-1} \left[ \sum_{\vec{m},\vec{n}} n_x n_y C_{\vec{m},\vec{n}} + \sum_{\vec{n} = \text{even}} n_x n_y c_{\vec{n}} \tilde{J}_{\vec{n}} \right]$$
$$\frac{\partial^2 H}{\partial J_x \partial \theta_y} = (2J_x)^{-1} \sum_{\vec{m},\vec{n}} n_x m_y S_{\vec{m},\vec{n}}$$
$$\frac{\partial^2 H}{\partial \theta_x \partial J_y} = (2J_y)^{-1} \sum_{\vec{m},\vec{n}} n_y m_x S_{\vec{m},\vec{n}}$$
$$\frac{\partial^2 H}{\partial \theta_x \partial \theta_y} = \sum_{\vec{m},\vec{n}} m_x m_y C_{\vec{m},\vec{n}}$$

One can see that the formula structure is quite simple and similar to those of  $\partial H/\partial J$  and  $\partial H/\partial \theta$ . Thus tables generated in SPB may be used to calculate the second order derivatives. As the result, DPB consumes typically about 20% more CPU time than SPB.

#### 4. Extend to nPB

Even in most cases the double PB is sufficient to represent the lattice, it is natural to extend to higher PB's in more demanding cases, even for hadron machines. As we have seen in the double PB section, we can build up higher PB's based on the results from the single PB. However, the formulas get more complicated for higher PB's. Here, we introduce a new notation for the formulas.

A Poisson bracket can be written as:

$$[A,B] = \frac{\partial A}{\partial \zeta_i} J_{ij} \frac{\partial B}{\partial \zeta_j}$$
  
where  $\mathbf{J} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}$ , and  $\zeta_i = \{\theta_x, J_x, \theta_y, J_y\}$ . Now, define:  
 $A_i \equiv \frac{\partial A}{\partial \zeta_i}, A_i \equiv \frac{\partial A}{\partial \zeta_j} J_{ji}.$ 

So, we can write single PB as:

 $[H, z] = H_i z_i$ , where  $z = \{x, p_x, y, p_y\}$ , and the sum over *i* is understood.

For double PB, we have

$$[H,[H,z]] = H_{\overline{i}} (H_{\overline{j}} z_j)_i = H_{\overline{i}} H_{\overline{j}i} z_j + H_{\overline{i}} H_{\overline{j}} z_{ij}$$

The double subscripts indicate the second order derivatives. The derivatives commute, *i. e.*,  $A_{ij} = A_{ji}$  etc. In double PB, we find  $z_i$  and  $H_{\bar{i}}$  has been calculated in SPB. The new quantities to be calculated are  $z_{ij}$  and  $H_{\bar{i}}H_{\bar{j}i} \equiv H_{\bar{j}}^{(2)}$ . Given the  $z(\zeta_i)$ , it is not difficult to figure out  $\mathbf{z}_{ij}$  in terms of  $\zeta_i$ . Thus, major calculations to get DPB is to calculate  $H_{\bar{j}}^{(2)}$ . This is identical to what was described in previous section.

Now, we can go on to the triple PB:

$$[H, [H, [H, z]]] = H_{\bar{k}} \Big( H_{\bar{j}}^{(2)} z_j + H_{\bar{i}} H_{\bar{j}} z_{ij} \Big)_{k}$$
  
$$= H_{\bar{k}} H_{\bar{j}k}^{(2)} z_j + H_{\bar{k}} H_{\bar{j}}^{(2)} z_{jk} + H_{\bar{k}} H_{\bar{i}k} H_{\bar{j}} z_{ij}$$
  
$$+ H_{\bar{k}} H_{\bar{i}} H_{\bar{j}k} z_{ij} + H_{\bar{k}} H_{\bar{i}} H_{\bar{j}} z_{ijk}$$
  
$$= H_{\bar{k}} H_{\bar{j}k}^{(2)} z_j + 3 H_{\bar{k}} H_{\bar{j}}^{(2)} z_{jk} + H_{\bar{k}} H_{\bar{i}} H_{\bar{j}} z_{ijk}$$

in which  $H_{jk}^{(2)} = \frac{\partial}{\partial \zeta_k} H_j^{(2)}$ . Again, we find that the quantities we need to calculate are  $H_{j}^{(3)} \equiv H_{\bar{k}} H_{jk}^{(2)}$  and  $z_{ijk}$ .

Similarly, we can get 4PB and 5PB, etc., without too much difficulties. High order derivatives are easy to calculate in the action-angle space, as we have seen in the previous section. However, it is obvious that many more terms will appear in higher PB calculation. The order of PB should be chosen according to the requirement of the physics problem and the speed. In addition, since the Poison brackets provide higher order accuracy for high order Hamiltonian terms, we believe that only lower order resonance terms need to go through higher PB calculations.

## **IV. DISCUSSION**

This resonance basis SPB or DPB tracking has been checked against Taylor map tracking and element-by-element tracking. Figure 1 shows an example of PEP-II dynamic aperture calculated by the resonance basis maps. This is the high energy ring of PEP-II<sup>[3]</sup>, with fractional tunes of 0.57/0.64, and  $\beta_y^*=2cm$ . The dynamic apertures are generated by SPB and DPB tracking with a 10th order Hamiltonian, and by the element-by-element tracking, for 1000 turns. The circle of the 10 $\sigma$  fully coupled beam size is plotted as a reference. Radiation effects are included. The damping time is 5000 turns. One can see that both SPB and DPB tracking give about the same dynamic aperture as element-by-element tracking. However, in many cases, DPB gives better agreement than SPB and is typically as good as truncated power series tracking.

The agreement may be explained by Figure 2. The difference of one turn amplitude kicks between SPB and DPB is plotted versus amplitudes. The amplitude changes due to radiation damping and quantum fluctuation are also plotted in Figure 2. Considering the DPB to be a relatively accurate result, the inaccuracy of SPB vertical kick exceeds the vertical radiation noise at  $10.5\sigma$ . Therefore, we expect the SPB may not be accurate enough to find the dynamic aperture. As for beam tail studies, we have found that the DPB, when compared to element-by-element tracking, has provided good results in all cases we have explored. The SPB is adequate for most cases, however, there are examples that show problems with SPB tracking.



Figure 1. Comparing dynamic aperture by element-by-element tracking, SPB and DPB tracking.

The speed is tested on a IBM RS/6000/750 UNIX work station, which is 22~27 Mflops per second. The CPU time for tracking one particle for 1 million turns is listed in the following table:

CPU time	7th order Hamiltonian		10th order Hamiltonian	
(min.)	Bare lattice	Full error	Bare lattice	Full error
SPB	5.0	8.6	14.3	26.8
DPB	5.8	10.3	16.8	31.8
Taylor map	8.6		40	

Two lattices are tested. The bare lattice includes high order components but no misalignment errors. The full error lattice includes all kinds of errors. For a 7th order Hamiltonian, the bare lattice has 188 resonance terms and 32 tune-shift terms, while the full error lattice has 376 resonance terms. For 10th order, the bare lattice has 728 resonance terms and 80 tune-shift terms, while the full error lattice has 1456 resonance terms. In the tracking, radiation effects are included. As a reference, it takes 592 minutes for element-by-element tracking using DESPOT for 1 million turns.

Figure 3 shows the CPU time change with the number of resonances retained in the Hamiltonian. One can see the advantage of dropping resonance terms. Assuming the resonances that do not intersect the tune spread area in tune space do not contribute to the dynamics, we may be able to reduce the number of resonances to 50. Then, the CPU time will be reduced to 2 to 3 minutes for SPB or DPB tracking, which is comparable with the time spent to calculate the 5-slice beam-beam kick. In addition, the CPU times in fig. 3 also include the transformation of RF cavity location. If the cavity location is not important, we can save more CPU time.





Figure 2. The amplitude kick difference between SPB and DPB plotted as a function of amplitude. One  $\sigma$  refers to A=2.2×10<sup>-4</sup> .



Figure 3. Comparing CUP time for tracking 1 million turns.

# **VI. CONCLUSIONS**

This method was originally designed to use in combination with our simulation method for beam-beam tails. Since the tail simulation requires about 100 million particle-turns tracking, speed is essential. We have found that since the resonances are the dominant factor in tail formation, the resonance basis Hamiltonian is the most suitable expression for studying the effects of lattice nonlinearities on tail formation. The CPU time is also acceptable.

This tracking method also can be a powerful tool in lattice design. Once a map is achieved, it only takes a few minutes to calculate a dynamic aperture. Parameters in the map such as working tune may be easily changed to carry out tune-space explorations. Additionally, many effects, such as beam-beam interaction and radiation, can easily be included. These methods are now being applied in PEP-II design work.

The nPB tracking methods have been tested against element-by-element tracking. Good agreement has been achieved. Inclusion of two Poisson brackets is adequate for calculation of dynamic apertures.

## **VI. REFERENCES**

- [1]. R. Kleiss, F. Schmidt and F. Zimmermann, *Particle Accelerators*, 41, 117,(1993)
- [2]. E. Forest, "Normal form algorithm on nonlinear symplectic maps", SSC-29, (1985)
- [3]. <u>PEP-II Conceptual Design Report</u>, SLAC-418, 1993