Imperfections and Correction

V. Ziemann

Uppsala University
What is this talk about?

- First, you come up with lattice and design optics
  - nice and shiny beta functions
  - high periodicity $\rightarrow$ systematic errors cancel
But then...

• ...the accelerator is built, and..
  – the magnets are not quite where they should be;
  – power supplies have calibration errors;
  – magnets have incorrect shims;
  – the diagnostics might have imperfections, too.
  • BPM
  • Screens
Therefore...

- I talk about
  - things that can go wrong (courtesy of Mrs Murphy...)
    → Imperfections
  - how to figure out what is wrong
    → Diagnostics to use
  - and fix it
    → Corrections
Outline

- Imperfections
- Straight systems
  - Beam lines and Linac
  - Imperfections and their corrections
- Rings
  - Imperfections and their corrections
Part 1: Linear Imperfections

• Spoil the 'nice&shiny™' periodic magnet lattice
  – due to unwanted magnetic fields in the wrong place
• that's where the beam is
  – average: dipole kick
  – gradient: focusing
  – skew gradient: coupling

• Solenoid fields
  – detector
  – electron cooler
Sources of Imperfections

- Anything that is not in the design lattice
- Fringe fields and cross talk between magnets
- Saturation of magnets
- Power supply calibration and read back errors
- Wrong shims
- Earth magnetic field in low-energy beam lines
- Nickel layers in the wrong place
- Solenoids in detectors or coolers
- Weak focusing from wigglers
- Tilt and roll angles of magnets
- Misaligned magnets (or beams)

Figure 1: The LEP dipole chamber and its nickel layer
J. Billan et al., PAC 1993
Alignment

• How do you do it?
  – Magnets on tables
  – Fiducialization to pods
  – Triangulation

• How well can you do it?
  – 0.2-0.3 mm OK
  – <0.1 mm increasingly more difficult
  – more difficult in large installations

• Sub-micron for linear colliders → beam-based
Misaligned Magnets

- Misalignment of linear elements

\[
\begin{pmatrix}
  x_f \\
  x'_f \\
\end{pmatrix} = \begin{pmatrix}
  -d_x \\
  0 \\
\end{pmatrix} + \tilde{R} \left[ \begin{pmatrix}
  d_x \\
  0 \\
\end{pmatrix} + \begin{pmatrix}
  x_i \\
  x'_i \\
\end{pmatrix} \right]
\]

\[
= \tilde{R} \begin{pmatrix}
  x_i \\
  x'_i \\
\end{pmatrix} + \left[ \tilde{R} - 1 \right] \begin{pmatrix}
  d_x \\
  0 \\
\end{pmatrix} = \tilde{q} + \tilde{R} \begin{pmatrix}
  x_i \\
  x'_i \\
\end{pmatrix}
\]

- and for a thin quadrupole…

\[
\tilde{q} = \left[ \tilde{R} - 1 \right] \begin{pmatrix}
  d_x \\
  0 \\
\end{pmatrix} = \begin{pmatrix}
  0 \\
  -\frac{1}{f} \\
\end{pmatrix} \begin{pmatrix}
  d_x \\
  0 \\
\end{pmatrix} = \begin{pmatrix}
  0 \\
  -\frac{d_x}{f} \\
\end{pmatrix}
\]

- An additional dipolar kick appears → feed-down
Misaligned quadupoles focus just as good as centered ones

- Same focal length despite misalignment
- Lower ray is further away from the quad center and bent more
- Upper ray is closer to axis and is bent less
- But they kick the centroid of the beam
Tilted elements

- come in, step right and point left, go through, step right again and point right

\[
\begin{pmatrix}
  x_f \\
  x'_f \\
\end{pmatrix}
 =
\begin{pmatrix}
  -d'_x L/2 \\
  d'_x \\
\end{pmatrix} + \hat{R} \begin{pmatrix}
  -d'_x L/2 \\
  d'_x \\
\end{pmatrix} + \begin{pmatrix}
  x_i \\
  x'_i \\
\end{pmatrix}
 = \hat{R} \begin{pmatrix}
  x_i \\
  x'_i \\
\end{pmatrix} + \left[ \hat{R} \begin{pmatrix}
  1 & 0 \\
  0 & -1 \\
\end{pmatrix} \right] \begin{pmatrix}
  -d'_x L/2 \\
  d'_x \\
\end{pmatrix} = \tilde{q} + \tilde{R} \begin{pmatrix}
  x_i \\
  x'_i \\
\end{pmatrix}
\]

- Again, normal transport and a constant vector
DIY: calculate ‘tilted’ $\vec{q} = \left[ \hat{r} + \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \right] \left( -\frac{d'_x L}{d_x} \right)$

- I use octave-like pseudo-code with notation as in the Octave/Matlab version of the tutorial

```matlab
% calculate_vec_q_for_tilt.m
F=2.5;  % focal length of the quadrupoles
fodo=[ 1, 1, 1.0, 0;  % Drift 1m
      2, 1, 0.0, -F;  % QD
      1, 1, 2.0, 0;  % Drift 2m
      2, 1, 0.0, F;  % QF
      1, 1, 1.0, 0];  % Drift 1m
beamline=[fodo;fodo];
[Racc,spos,nmat,nlines]=calcmat(beamline);
% tilt both quads in first cell, element 2 to 4 (note all repeat codes are 1)
L=spos(4)-spos(2)  % end of fourth element minus second
dprime=1e-3;
% Racc start with unit matrix 'before' the beam line
Rhat=Racc(:,:,5)*inv(Racc(:,:,1))
Rhat(1,1)=Rhat(1,1)+1;
Rhat(2,2)=Rhat(2,2)-1;
qvec=Rhat*[-dprime/L, dprime]
```

https://arxiv.org/abs/1907.10987

a little simplified
NO repeat codes
to avoid book keeping
Longitudinally Shifted Elements

- Add a short positive element on one side and the negative on the other
- Dipole
  - kick on either side
- Quadrupoles
  - thin quadrupoles

How would you implement this in your code?
Incorrectly powered Quadrupoles

- Focal length changes
  - beam matrix differs from the expected
  - beta functions change
  - in rings, the tune changes
Undulators and Wigglers

- $B_y \sim \cos(2\pi s/\lambda_u) \rightarrow$ horizontal oscillations
- $\partial B_y/\partial s = \partial B_s/\partial y \rightarrow$ vertically changing $B_s$
- Focus vertically (only)
- Many Rbends
- Weak effect $(l/\rho)^2$, but
- Changing gap may
  - affect orbit
  - affect tune

“Hilda”
Dispersion

- Effect of magnetic fields on the beam ($\sim B/\rho$) with $\rho = \rho_0 (1+\delta)$ is reduced by $1+\delta$

- Every dipole behaves as a spectrometer
  - separates the particles according to their momentum
  - even dipole correctors contribute

- In planar systems the vertical dispersion is by design zero
  - but rolled dipoles (and quadrupoles) make it non-zero.

Check out hands-on exercises 33 to 38 about how this is done in software!

...and revisit Wolfgang’s slides, section 5
Chromaticity

- Also quadrupolar fields are reduced by $1 + \delta$
  - Longitudinal location of the focal plane depends on momentum and enlarges the beam sizes at the IP
  - Chromaticity $Q' = \frac{dQ}{d\delta}$
  - Tune spread

How would you implement this in your software?

...again, revisit Wolfgang's Section 5.
Measuring Dispersion and Chromaticity

• Change the beam energy, in rings by changing the RF frequency
  - and look at orbit from BPMs (dispersion)
  - and measure the tune (chromaticity)
• In transfer line or linac change the energy of the injected beam
• Optionally, may scale all magnets with the same factor
  - all beam observables ~B/p
Rolled elements

- Coordinate rotation

\[
\begin{pmatrix}
  x_2' \\
  x_2' \\
  y_2' \\
  y_2'
\end{pmatrix} = 
\begin{pmatrix}
  \cos \phi & 0 & \sin \phi & 0 \\
  0 & \cos \phi & 0 & \sin \phi \\
  -\sin \phi & 0 & \cos \phi & 0 \\
  0 & -\sin \phi & 0 & \cos \phi
\end{pmatrix}
\begin{pmatrix}
  x_1 \\
  x_1' \\
  y_1' \\
  y_1'
\end{pmatrix}
\]

- sandwich roll-left before the element and then roll-right after the element

- example quad to skew-quad (example, thin quad)

\[
Q_s = R(-\pi/4) \begin{pmatrix} Q_f & 0_2 \\ 0_2 & Q_d \end{pmatrix} R(\pi/4) = 
\begin{pmatrix}
  1 & 0 & 0 & 0 \\
  0 & 1 & 1/f & 0 \\
  0 & 0 & 1 & 0 \\
  1/f & 0 & 0 & 1
\end{pmatrix}
\]

- mixes the transverse planes \(\rightarrow\) coupling
Reminder: Multipoles

- Magnet builder's view ($b_m$: upright, $a_m$: skew)

\[ B_y + iB_x = B_0 \sum_{m=1}^{\infty} (b_m + ia_m) \left( \frac{x + iy}{R_0} \right)^{m-1} \]

- How the beam “sees” the fields

\[ \Delta x' - i\Delta y' = \frac{(B_y + iB_x)L}{B\rho} = \sum_{n=0}^{\infty} \frac{k_n L}{n!} (x + iy)^n \]

- Multipole coefficients
  - real part: upright
  - imaginary part: skew

\[ \frac{k_n L}{n!} = \frac{(B_0/R_0^n)L}{B\rho} (b_{n+1} + ia_{n+1}) \]

\[ k_n L = \frac{d^n B/dx^n}{B\rho} L \]
Feed-down from displaced multipoles

- Kick from thin multipole
  \[ \Delta x' - i \Delta y' = \frac{k_n L}{n!} (x + iy)^n \]

- and from a displaced multipole
  \[ \Delta x' - i \Delta y' = \frac{k_n L}{n!} (x + dx + iy)^n \]
  \[ = \frac{k_n L}{n!} (x + iy)^n + \frac{k_n L}{n!} \sum_{k=0}^{n-1} \binom{n}{k} d_x^{n-k} (x + iy)^k \]
  - binomial expansion

- Displaced multipole still works as intended, but also generates all lower multipoles
Feed-down from sextupoles

- Horizontally misaligned
  \[ \Delta x' - i\Delta y' = \frac{k_2 L}{2} \left[ (x + iy)^2 + 2d_x(x + iy) + d_x^2 \right] \]
  - additional quadrupolar and dipolar kicks

- Vertically misaligned
  \[ \Delta x' - i\Delta y' = (x + iy + id_y)^2 = \frac{k_2 L}{2} \left[ (x + iy)^2 + 2id_y(x + iy) - d_y^2 \right] \]
  - additional skew-quadrupolar and dipole kicks
  - vertical displacement in sextupoles causes coupling
Detrimental effects

- Dipole fields cause beam to be in wrong place
  - losses, bad if you have a multi-MJ beam
  - Background in the experiments
- Gradients change the beam size, this spoils
  - Luminosity, if you work on a collider
  - Coherence, if you work on a light source
- Breaks the symmetry of the optics of a ring
  - more resonances
  - reduces dynamic aperture
- Need observables to figure out what's wrong
Beam Position Monitors and their Imperfections

- Transverse offset
- (Longitudinal position)
- Electrical offset
- Scale error

\[ x = k_x \frac{(S_A + S_D) - (S_B + S_C)}{S_A + S_B + S_C + S_D} \]
Find offsets with K-modulation

- BPM+Quadrupole are often mounted on the same support

\[ I = I_0 + \Delta I \cos(\omega_{\text{mod}} t) \]

- Modulate gradient of quadrupole
  - Kick from quadrupole \( \Theta = \frac{dx}{f(\omega)} \) is also modulated
  - Observe on BPM2 and minimize signal by moving beam with a bump \( \rightarrow \) quadrupole center
  - Reading of BPM1 gives BPM1 offset rel. to Quad
Screens et al. and their Bugs

- Transverse position
- Scale errors from the optical system
  - place fiducial marks on the screen
- Looking at an angle
- Depth of focus limitations, especially at large magnification levels
- Burnt-out spots on fluorescent screens
- Non-linear response of screen and saturation
Imperfections and their Correction in Beam Lines or Linacs

- Dipole errors
- Gradient errors
- Skew-gradient errors
- Filamentation
Transfer matrices in linacs

• Just a reminder...

• The beam energy at the location for the kick and the observation point may be different.

• Adiabatic damping
  – transverse momentum $p_x$ is constant
  – longitudinal momentum $p_s$ increases
  – $x' = p_x / p_s$ scales with $p_s = \beta \gamma mc$

• $R_{12}$ then scales with $(\beta \gamma)_{\text{kick}} / (\beta \gamma)_{\text{look}}$
Beam lines: Dipole errors

- Each misaligned element with label $k$ may add a misalignment dipole-kick $\vec{q}_k$

$$\vec{x}_n = R_n \cdots (\vec{q}_{k+1} + R_{k+1})(\vec{q}_k + R_k) \cdots (\vec{q}_1 + R_1)\vec{x}_0$$

$$= R_n \cdots R_1 \vec{x}_0 + \sum_{j=1}^{n-1} (R_n \cdots R_{j+1})\vec{q}_j$$

- Simple interpretation
  - at the look-point (BPM) $n$ all perturbing kicks are added with the transfer matrix from kick to end
DIY: in software

• We need all transfer matrices from the location with errors \((k)\) to the BPM \((n)\)

```matlab
% all_transfer_matrices.m

beamline=.. % define the beamline
[Racc,spos,nmat,nlines]=calcmat(beamline);

table_of_errpos=[..,k,..] % position of errors
n= % position in lattice file of BPM

for k=1:length(table_of_errpos)
    errpos=table_of_errpos(k)-1; % from just upstream of error
    RR(k)=Racc(:,:,n)*inv(Racc(:,:,errpos));
end
```

more details, explanations, and example code at https://www.crcpress.com/9781138589940
Correct with orbit correctors

- small dipole magnet, here for both planes (steerer for CTF3-TBTS)
- affects the beam like any other error

\[
\begin{pmatrix}
  x_1 \\
  x'_1
\end{pmatrix}
= \begin{pmatrix}
  0 \\
  \theta
\end{pmatrix}
+ \begin{pmatrix}
  x_0 \\
  x'_0
\end{pmatrix}
\]

\[
\vec{x}_1 = \vec{q} + \tilde{R}\vec{x}_0
\]

- treat just as additional misalignment
Local trajectory Bumps

- Occasionally a particular displacement or angle of the orbit at a given point might be required.
- Displace orbit at IP to bring beams into collision.
- Or a slight excursion (3-bump).
- Differential changes ('by' not 'to').
Trajectory knob

• Change position and angle at reference point

![Diagram showing trajectory knobs and their positions](image)

- $\Delta x_0, \Delta x_0'$
- $\theta_2$
- $\theta_1$

• Remember that kicks add up with TM from source to observation or reference point

$$
\begin{pmatrix}
\Delta x_0 \\
\Delta x_0'
\end{pmatrix}
=
\begin{pmatrix}
R_{12}^{01} & R_{12}^{02} \\
R_{22}^{01} & R_{22}^{02}
\end{pmatrix}
\begin{pmatrix}
\theta_1 \\
\theta_2
\end{pmatrix}
$$

• and the **columns of the inverse matrix** are the knobs

$$
\begin{pmatrix}
\theta_1 \\
\theta_2
\end{pmatrix}
=
\begin{pmatrix}
R_{12}^{01} & R_{12}^{02} \\
R_{22}^{01} & R_{22}^{02}
\end{pmatrix}^{-1}
\begin{pmatrix}
\Delta x_0 \\
\Delta x_0'
\end{pmatrix}
$$
A trivial example

- Two steering magnets with drift between them

\[
\begin{aligned}
\Delta x_0, \Delta x_0' & & \theta_2 & & \theta_1 \\
R^{02} &= \begin{pmatrix} 1 & L \\ 0 & 1 \end{pmatrix} & R^{01} &= \begin{pmatrix} 1 & 2L \\ 0 & 1 \end{pmatrix}
\end{aligned}
\]

- Response matrix

\[
\begin{pmatrix} \Delta x_0 \\ \Delta x_0' \end{pmatrix} = \begin{pmatrix} R^{01}_{12} & R^{02}_{12} \\ R^{01}_{22} & R^{02}_{22} \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} 2L & L \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}
\]

- Knobs

\[
\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} = \frac{1}{L} \begin{pmatrix} 1 & -L \\ -1 & 2L \end{pmatrix} \begin{pmatrix} \Delta x_0 \\ \Delta x_0' \end{pmatrix} \quad \text{Almost common sense!}
\]

\[
\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} = \frac{1}{L} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \Delta x_0
\]
Remark about Orthogonality

• Knobs are orthogonal
• Optimize one parameter without screwing up the other(s).
  - Faster convergence
  - Enables heuristic optimization
  - Deterministic
• Use physics rather than hardware parameters
Optimality

- How “good” are the knobs?
- Position knob is ill-defined for $L \rightarrow 0$.
- Matrix inversion can fail $\rightarrow$ condition number $\xi = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}}$
  - $\xi=1$: All parameters controlled equally well
- Consider beamline with betas equal and $\alpha=0$ at steerers and observation point
  \[
  \begin{pmatrix}
  \Delta x_0 \\
  \Delta x'_0 \\
  \end{pmatrix} = \begin{pmatrix}
  R_{12}^{01} & R_{12}^{02} \\
  R_{22}^{01} & R_{22}^{02} \\
  \end{pmatrix} \begin{pmatrix}
  \theta_1 \\
  \theta_2 \\
  \end{pmatrix}
  \]
  \[
  \begin{pmatrix}
  \Delta x / \sqrt{\beta} \\
  \sqrt{\beta} \Delta x' \\
  \end{pmatrix} = \begin{pmatrix}
  \sin 2\mu & \sin \mu \\
  \cos 2\mu & \cos \mu \\
  \end{pmatrix} \begin{pmatrix}
  \sqrt{\beta} \theta_1 \\
  \sqrt{\beta} \theta_2 \\
  \end{pmatrix}
  \]
- Generally applicable

$90^\circ$, common sense!
4-Bump

- Use four steerers to adjust angle and position at a center point and then flatten orbit downstream of the last steerer.

\[
\begin{pmatrix}
    x_0 \\
x'_0 \\
x_f = 0 \\
x'_f = 0
\end{pmatrix}
= \begin{pmatrix}
    R_{12}^{01} & R_{12}^{02} & 0 & 0 \\
    R_{22}^{01} & R_{22}^{02} & 0 & 0 \\
    R_{12}^{f1} & R_{12}^{f2} & R_{12}^{f3} & R_{12}^{f4} \\
    R_{22}^{f1} & R_{22}^{f2} & R_{22}^{f3} & R_{22}^{f4}
\end{pmatrix}
\begin{pmatrix}
    \theta_1 \\
    \theta_2 \\
    \theta_3 \\
    \theta_3
\end{pmatrix}
\]

- Invert matrix and express thetas as a function of the constraints \(x_0\) and \(x'_0\)

- Gives the required steering excitations \(\theta_j\) as a function of \(x_0\) and \(x'_0\) → Multiknob
% software_bumps.m

beamline=.. % define the beamline
[Racc, spos, nmat, nlines]=calcmat(beamline);

n=.. % x0, position where I care
cor=[p1,p2,p3,p4]; % table of four corrector positions

R01=Racc(:,,:)
p1-1)*inv(Racc,:,:)
R02=Racc(:,,:)
p2-1)*inv(Racc,:,:)
R41=Racc,:,:
p4)*inv(Racc,:,:)
R42=Racc,:,:
p4)*inv(Racc,:,:)
R43=Racc,:,:
p4)*inv(Racc,:,:)
R44=Racc,:,:
p4-1)*inv(Racc,:,:)

A=[R01(1,2),R02(1,2),0,0;
R01(2,2),R02(2,2),0,0;
R41(1,2),R42(1,2),R43(1,2),R44(1,2);
R41(2,2),R42(2,2),R43(2,2),R44(2,2)];

knobs=inv(A) % invert the 'response'

knob_x0=knobs(:,1) % first column

knob_x0p=knobs(:,2) % second column

- Setup beam line
- Get transfer matrices
- Assemble matrix
- Invert it
- Knobs are columns of the inverse matrix
  - knob(1,1) is the proportionality constant to change $\theta_1$
    when changing only $x_0$
Orbit Correction in Beamline #1

- Observe the orbit on beam-position monitors
- and correct with steering dipoles
- How much do we have to change the steering magnets in order to compensate the observed orbit either to zero or some other 'golden orbit'.
- In beam line the effect of a corrector on the downstream orbit is given by transfer matrix element $R_{12}$
- One-to-one steering
Orbit correction in a Beamline #2

- Observed beam positions $x_1$, $x_2$, and $x_3$
- Only downstream BPM can be affected
- Linear algebra problem to **invert matrix** and find required corrector excitations $\theta_j$ to produce negative of observed $x_i$
- Include BPM errors by left-multiplying the equation with

$$\bar{\Lambda} = \text{diag}\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_n}\right)$$

This weights each BPM measurement by its inverse error. Good BPMs are trusted more!
How to get the response matrix?

• With the computer (MADX or any other code)
  – tables of transfer matrix elements
  – but it is based on a model and somewhat idealized
  – no BPM or COR scale errors known

• Experimentally by measuring difference orbits
  – record reference orbit $\vec{x}_0$
  – change steering magnet $\Delta \theta_j$
  – record changed orbit $\vec{x}_j$
  – Build response matrix one column at a time

\[ A = \left( \frac{\vec{x}_1 - \vec{x}_0}{\Delta \theta_1}, \frac{\vec{x}_2 - \vec{x}_0}{\Delta \theta_2}, \ldots \right) \]
Solving $-x = A\theta$

- $A$ is an $n \times m$ matrix, $n$ BPM and $m$ correctors
- $n=m$ and matrix $A$ is non-degenerate:
  \[ \vec{\theta} = -A^{-1} \vec{x} \]
- $m<n$: too few correctors, least squares 
  \[ \chi^2 = | -\vec{x} - A\vec{\theta}|^2 \]
  \[ \vec{\theta} = -(A^t A)^{-1} A^t \vec{x} \]
- MICADO: pick the most effective, fix orbit, the next effective, fix residual orbit, the next...
  - good for large rings with many BPM and COR
- $m>n$ or degenerate: singular-value dec. (SVD)
Digression on SVD

• Singular Value Decomposition
  – may need to zero-pad
  – $U$ is orthogonal, a coordinate rotation
  – $\Lambda$ is diagonal, it stretches the coordinates by $\lambda_i$
  – $O$ is orthogonal and rotates, but differently

• If $A$ is symmetric $\rightarrow$ eigenvalue decomposition

• Inversion is trivial
  
  
  
  "$A^{-1}$" = $U\Lambda^{-1}O^t$

  – invert only in sub-space where you can if $\lambda \neq 0$
  – and set projection onto degenerate subspace to zero
  “$1/0 = 0$” (see Numerical Recipes for a discussion)
DIY: calculate “SVD-inverse”

- Consider 2x2 matrix
  - and invert it as $z \rightarrow 0$

```matlab
% svd_example.m
clear all;
A=@(z)[1,2;2+z,4];
Atest=A(1); % or A(0)
[0,L,V]=svd(Atest) % A=O*L*V'
n=size(L,1); % dimension of matrix
Linv=zeros(n,n); % create matrix with all zeros
for k=1:size(L,1) % invert where you can
    if abs(L(k,k)) > 1e-8 Linv(k,k)=1/L(k,k);
end
Ainv=V*Linv*V'; % calculate inverse
check=Atest*Ainv % should be unit matrix, unless...
```
Comment on Matrix Inversion

• Many correction problems can be brought into a generic form, if you
  – pretend you know the excitation of all controllers (think correctors)
  – determine the response matrix (expt. or numerically)
    \[ C_{ij} = \frac{\partial \text{Observable}_i}{\partial \text{Controller}_j} \]
  – to predict the changes of the observables (think BPM)

• Then invert the response matrix to determine the controller values required to change the observable by some value.
Effect of gradient errors

Eight 90° FODO cells, first quad 10% too low

Unperturbed lattice

Nice and repetitive beta functions

Repeats after 2 cells or 2 x 90°

Beta-function “beats”

Injection into following beam line or ring is compromised
Beam lines: Gradient errors

• Gradient errors cause the beam matrix or beta functions $\beta$ to differ from their design values $\hat{\beta}$

• Downstream beam size

$$\bar{\sigma}_x^2 = \varepsilon \hat{\beta} \left[ B_{mag} + \sqrt{B_{mag}^2 - 1 \cos(2\mu - \varphi)} \right]$$

  - enlarged effective emittance, beta-beat oscillations with twice the betatron phase advance $\mu$

• This is called mismatch and is quantified by

$$B_{mag} = \frac{1}{2} \left[ \left( \frac{\hat{\beta}}{\beta} + \frac{\beta}{\hat{\beta}} \right) + \beta \hat{\beta} \left( \frac{\alpha}{\beta} - \frac{\hat{\alpha}}{\hat{\beta}} \right)^2 \right]$$

• For a single thin quad we have

$$B_{mag} = 1 + \frac{\hat{\beta}^2}{2f^2}$$
Filamentation #1

- What happens when we inject a mismatched beam into a ring with chromaticity $Q'$?

\[ \sigma_n^2 = \varepsilon \tilde{\beta} \left[ B_{mag} + \sqrt{B_{mag}^2 - 1} \cos(4\pi n (Q + Q' \delta) - \varphi) \right] \]

- with momentum distribution

\[ \psi(\delta) = \frac{1}{\sqrt{2\pi\sigma_\delta}} e^{-\delta^2/2\sigma_\delta^2} \]

- Averaging over $\delta$ gives

\[ \sigma_n^2 = \varepsilon \tilde{\beta} \left[ B_{mag} + e^{-2(2\pi Q' \delta)^2 n^2} \sqrt{B_{mag}^2 - 1} \cos(4\pi n Q - \varphi) \right] \]

- Oscillates with $2 \times Q$, 'damps' with $\exp(-n^2)$, and leaves an increased beam size (by $B_{mag}$).
Filamentation #2

Matched beam in the ring
Injected beam

Smeared-out black spot

Final distribution is not Gaussian

Bmag=3

Injecting with transverse offset also leads to filamentation

You've seen it before...
Measuring Beam Matrices

V. Ziemann: Imperfections and Correction

\[ \bar{\sigma} = R(f) \sigma R(f)^t \]

Vary quadrupole and observe changes on a screen, usually one plane at a time

- Beam size on screen depends on quad setting

\[ \bar{\sigma}_x^2 = \bar{\sigma}_{11} = R_{11}^2 \sigma_{11} + 2R_{11}R_{12}\sigma_{12} + R_{12}^2 \sigma_{22} \]

- where \( R=R(f) \), use several measurement and solve for the three sigma matrix elements

\[ \varepsilon_x^2 = \sigma_{11}\sigma_{22} - \sigma_{12}^2 \quad \beta_x = \sigma_{11}/\varepsilon_x \quad \alpha_x = -\sigma_{12}/\varepsilon_x \]
A worked example: Quad scan

- Transfer matrix

\[ R = \begin{pmatrix} 1 & l \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -1/f & 1 \end{pmatrix} = \begin{pmatrix} 1 - l/f & l \\ -1/f & 1 \end{pmatrix} \]

- Relate unknown beam matrix to measurements

\[
\bar{\sigma}_x^2 = R_{11}^2 \sigma_{11} + 2R_{11}R_{12}\sigma_{12} + R_{12}^2 \sigma_{22} \\
= (1 - l/f)^2 \sigma_{11} + 2l(1 - l/f)\sigma_{12} + l^2 \sigma_{22} \\
= \left(\frac{l}{f}\right)^2 \sigma_{11} - \left(\frac{l}{f}\right) (2\sigma_{11} + 2l\sigma_{12}) + (\sigma_{11} + 2l\sigma_{12} + l^2 \sigma_{22})
\]

- Indeed a parabola in \( l/f \)
Quad scan #2

- Build matrix of the type \( y = Ax \)
  - and with error bars \( \Sigma_k = 2\sigma_k \Delta \sigma_k \)

\[
\begin{pmatrix}
\sigma_{x,1}^2 \\
\sigma_{x,2}^2 \\
\sigma_{x,3}^2 \\
\sigma_{x,4}^2 \\
\sigma_{x,5}^2
\end{pmatrix} = \begin{pmatrix}
(1 - l/f_1)^2 & 2l(1 - l/f_1) & l^2 \\
(1 - l/f_2)^2 & 2l(1 - l/f_2) & l^2 \\
(1 - l/f_3)^2 & 2l(1 - l/f_3) & l^2 \\
(1 - l/f_4)^2 & 2l(1 - l/f_4) & l^2 \\
(1 - l/f_5)^2 & 2l(1 - l/f_5) & l^2
\end{pmatrix} \begin{pmatrix}
\sigma_{11} \\
\sigma_{12} \\
\sigma_{22}
\end{pmatrix} = \begin{pmatrix}
\sigma_{x,1}^2 \\
\sigma_{x,2}^2 \\
\sigma_{x,3}^2 \\
\sigma_{x,4}^2 \\
\sigma_{x,5}^2
\end{pmatrix} = \begin{pmatrix}
\Sigma_1 \\
\Sigma_2 \\
\Sigma_3 \\
\Sigma_4 \\
\Sigma_5
\end{pmatrix} = \begin{pmatrix}
\Sigma_1 \\
\Sigma_2 \\
\Sigma_3 \\
\Sigma_4 \\
\Sigma_5
\end{pmatrix} \begin{pmatrix}
\sigma_{11} \\
\sigma_{12} \\
\sigma_{22}
\end{pmatrix}
\]

- Solve by least-squares pseudo-inverse
  \( x = (A^tA)^{-1}A^t y \)

- with the covariance matrix \( \text{Cov} = (A^tA)^{-1} \)
  - diagonal elements are square of error bars of fit parameter \( x \)
Or use several wire scanners

- \((A^t A)^{-1}A^t\) - gymnastics with error bar estimates
- Derive emittance in same way, once \(\sigma_{ij}\) is found by inversion
- Can use several more wire scanners which allows \(\chi^2\) calculation for goodness-of-fit estimate

\[
\begin{align*}
\chi^2 &= (R^t)^t (R^t) \sigma_{11} + 2R^t (R^t) \sigma_{12} + (R^t)^t \sigma_{22} \\
\chi^2 &= (R^t)^t \sigma_{11} + 2R^t (R^t) \sigma_{12} + (R^t)^t \sigma_{22} \\
\chi^2 &= (R^t)^t \sigma_{11} + 2R^t (R^t) \sigma_{12} + (R^t)^t \sigma_{22}
\end{align*}
\]
DIY: emittance from three wire scanners in software

% three_wire_scanners.m

beamline=..  % define the beamline
[Racc,spos,nmat,nlines]=calcmat(beamline);
p0=..        % Reference position
cor=[w1,w2,w3];  % table of three wire scanners
R1=Racc(:,,:,w1)*inv(Racc(:,,:,p0-1));
R2=Racc(:,,:,w2)*inv(Racc(:,,:,p0-1));
R3=Racc(:,,:,w3)*inv(Racc(:,,:,p0-1));
A=[R1(1,1)^2, 2*R1(1,1)*R1(1,2), R1(1,2)^2;  
   R2(1,1)^2, 2*R2(1,1)*R2(1,2), R2(1,2)^2;  
   R3(1,1)^2, 2*R3(1,1)*R3(1,2), R3(1,2)^2]

sigx=[sig1;sig2;sig3];  % measurements
sigma0=inv(A)*(sigx.^2)  % sigma/beam matrix at p0

eps=sqrt(sigma0(1)*sigma0(3)-sigma0(2)^2);
beta=sigma0(1)/eps;
alfa=-sigma0(2)/eps;

\[ \varepsilon_x^2 = \sigma_{11}\sigma_{22} - \sigma_{12}^2 \]
\[ \beta_x = \sigma_{11}/\varepsilon_x \]
\[ \alpha_x = -\sigma_{12}/\varepsilon_x \]
Fix beam matrix a.k.a. Beta match

- Uncoupled beam matrix
  \[ \varepsilon_x \begin{pmatrix} \beta_x & -\alpha_x \\ -\alpha_x & \gamma_x \end{pmatrix} \]
  - need four quadrupoles to adjust \( \alpha_x, \beta_x, \alpha_y, \) and \( \beta_y \)
  - non-linear optimizer (MADX matching module)

This is a good time to re-visit hands-on exercise 27 and 28
Waist knob

• Finding quad-excitations to match beta functions (or sigma matrix) is a non-linear problem

• and depends on the incoming beam matrix.

• Tricky, but one sometimes can build knobs, based on the design optics, to correct some observable
  – conceptually: linearizing around a working point

• Example:
  – IP-waist knob
  – $d\alpha_x/dQ_{1,2}$ and $d\alpha_y/dQ_{1,2}$
Beam lines: Skew-gradient errors

- Transfer matrix for a skew-quadrupole

$$s = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1/f & 0 \\ 0 & 0 & 1 & 0 \\ 1/f & 0 & 0 & 1 \end{pmatrix}$$

- Vertical part of the sigma-matrix after skew quad

$$\begin{pmatrix} \hat{\sigma}_{33} & \hat{\sigma}_{34} \\ \hat{\sigma}_{34} & \hat{\sigma}_{44} \end{pmatrix} = \begin{pmatrix} \sigma_{33} & \sigma_{34} \\ \sigma_{34} & \sigma_{44} + \sigma_{11}/f^2 \end{pmatrix}$$

- Projected emittance after skew quadrupole

$$\hat{\varepsilon}_y^2 = \varepsilon_y^2 + \frac{\sigma_{11}\sigma_{33}}{f^2} = \varepsilon_y^2 \left(1 + \frac{\varepsilon_x \beta_x \beta_y}{\varepsilon_y \frac{f^2}{2}}\right)$$

  - Problem with flat beams. Increases with ratio $\varepsilon_x / \varepsilon_y$ and is proportional to both beta functions.

- Problem in Final-Focus Systems with flat beams. Solenoid fields need compensation.

Exercise 44: add skew-quadrupoles to the code and play around with it.
Imperfections in a Ring

- Effect of a localized kick on orbit
- Effect of a localized gradient error
- Effect of a skew gradient error
- Stop-bands and resonances
Dipole errors in a Ring

- Beam bites its tail → periodic boundary conditions → closed orbit

- Orbit after perturbation at $j$
  \[ \vec{x}_j = R^{jj} \vec{x}_j + \vec{q}_j \]
  \[ \vec{x}_j = (1 - R^{jj})^{-1} \vec{q}_j \]

- Propagate to BPM i
  \[ \vec{x}_i = R^{ij} \vec{x}_j = R^{ij} (1 - R^{jj})^{-1} \vec{q}_j = C^{ij} \vec{q}_j \]

- Response coefficients
  \[ C^{ij} = R^{ij} (1 - R^{jj})^{-1} \]
  just like transfer matrix in beam line, but with built-in closed-orbit constraint.
DIY: response coefficients

Depends how matrices in Racc are organized:
\[ R_{jj} = (\text{Start2COR}) \times (\text{Start2End}) \times \text{inv}(\text{Start2COR}) \]
and
\[ R_{ij} = \text{depends on whether BPM}(i) \text{ is upstream or downstream of corrector}(j) \]

% CC.m, BPM-COR response matrix in ring
function out=CC(ipos,jpos,Racc)
    Rjj=Racc(:,:,jpos)*Racc(:,:,end)*inv(Racc(:,:,jpos));
    if ipos > jpos
        Rij=Racc(:,:,ipos)*inv(Racc(:,:,jpos));
    else
        Rij=Racc(:,:,ipos)*Racc(:,:,end)*inv(Racc(:,:,jpos));
    end
    out=Rij*(inv(eye(4)-Rjj));

\[ C^{ij} = R_{ij} (1 - R_{jj})^{-1} \]
Response coefficients with beta functions

- Express transfer-matrices through beta functions

\[
\begin{pmatrix}
x \\
x'
\end{pmatrix}
_j =
\begin{pmatrix}
\cos(2\pi Q) & \beta_j \sin(2\pi Q) \\
-\sin(2\pi Q)/\beta_j & \cos(2\pi Q)
\end{pmatrix}
\begin{pmatrix}
x \\
x'
\end{pmatrix}
_j +
\begin{pmatrix}
0 \\
\theta
\end{pmatrix}
\]

- Solve for closed orbit

\[
\begin{pmatrix}
x \\
x'
\end{pmatrix}
_j = \frac{\theta}{2} \begin{pmatrix}
\beta_j \cot(\pi Q) \\
1
\end{pmatrix}
\]

- Transfer matrix to BPM i

\[
R_{ij} = \begin{pmatrix}
\sqrt{\beta_i} & 0 \\
-\alpha_i/\sqrt{\beta_i} & 1/\sqrt{\beta_i}
\end{pmatrix}
\begin{pmatrix}
\cos \mu_{ij} & \sin \mu_{ij} \\
-\sin \mu_{ij} & \cos \mu_{ij}
\end{pmatrix}
\begin{pmatrix}
1/\sqrt{\beta_j} & 0 \\
0 & \sqrt{\beta_j}
\end{pmatrix}
\]

- Response coefficient

\[
x_i = \left[ \frac{\sqrt{\beta_i \beta_j}}{2 \sin(\pi Q)} \cos(\mu_{ij} - \pi Q) \right] \theta
\]

\[
C_{12}^{ij} = \frac{\partial BPM_i(x)}{\partial COR_j(x')}
\]

You’ve seen it before, check out Wolfgang’s slide 88
Quadrupole alignment amplification factor

- Consider randomly displaced quadrupoles

\[ \theta_j = \frac{d_j}{f} \quad \langle d_j \rangle = 0 \quad \langle d_j d_k \rangle = \sigma_d^2 \delta_{jk} \]

- Incoherently (RMS) add all contributions

\[
\langle x_i^2 \rangle = \left\langle \left[ \sum_j \frac{\sqrt{\beta_i \beta_j}}{2 \sin \pi Q} \cos(\mu_{ij} - \pi Q) \frac{d_j}{f_j} \right] \left[ \sum_k \frac{\sqrt{\beta_i \beta_k}}{2 \sin \pi Q} \cos(\mu_{ik} - \pi Q) \frac{d_k}{f_k} \right] \right\rangle \\
= \sum_j \frac{\beta_i \beta_j}{(2 \sin \pi Q)^2} \cos^2(\mu_{ij} - \pi Q) \frac{\sigma_d^2}{f_j^2}
\]

- Misalignment amplification factor

\[ \sqrt{\langle x_i^2 \rangle} \approx \sqrt{N_q} \frac{\bar{\beta}/\bar{f}}{2\sqrt{2} \sin \pi Q} \sigma_d \]

- large rings with large \( N_q \) are sensitive
- such as LHC and FCC
Response Coefficients with RF

- Radio-frequency system constrains the revolution time
  \[
  \frac{\Delta T}{T} = \frac{\Delta C}{C} - \frac{\Delta v}{v} = \left(\alpha - \frac{1}{\gamma^2}\right) \delta
  \]

- but a horizontal kick causes a horizontal closed orbit distortion which causes the circumference to change by \(\Delta C = D_x \theta_x\)
  (6x6 TM is symplectic, and if uncoupled: \(R_{16} = R_{52}\))

- Since RF fixes the revolution frequency the momentum of the particle has to adjust to \(\delta = -D_j \theta / \eta C\)

- ...and the particle moves on a dispersion trajectory

- Complete response coefficient between BPM\(_i\) and dipole error or COR\(_j\)
  \[
  C_{12}^{i,j} = \left[ \frac{\sqrt{\beta_i \beta_j}}{2 \sin(\pi Q)} \cos(\mu_{i,j} - \pi Q) - \frac{D_i D_j}{\eta C} \right]
  \]

190913, CAS High Tatras

V. Ziemann: Imperfections and Correction
Orbit Correction in a Ring

• Every steering magnet affects every BPM
  – orbit response coefficients and matrix \( C^{ij} = R^{ij} (1 - R^{ij})^{-1} \)

• Compensate measured positions \( x_i \) by inverting

\[
\begin{pmatrix}
-x_1 \\
-x_2 \\
\vdots \\
-x_m
\end{pmatrix} = \begin{pmatrix}
C^{11}_{12} & C^{12}_{12} & \cdots & C^{1n}_{12} \\
C^{21}_{12} & C^{22}_{12} & \cdots & C^{2n}_{12} \\
\vdots & \vdots & \ddots & \vdots \\
C^{m1}_{12} & C^{m2}_{12} & \cdots & C^{mn}_{12}
\end{pmatrix}
\begin{pmatrix}
\theta_1 \\
\theta_2 \\
\vdots \\
\theta_n
\end{pmatrix}
\]

• and also in the vertical plane

• left-multiply with diagonal BPM error matrix \( \bar{\Lambda} = \text{diag} \left( \frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_n} \right) \)

• use either calculated or measured response matrix

• inversion with pseudo-inverse, MICADO, or SVD
DIY: BPM-COR matrix

- Receive table of BPM and corrector positions as input. Is based on Racc.
- Loops over all combinations and returns Cx and Cy (uncoupled)

```matlab
function [Cx,Cy]=response_coefficients(bpmpos,corpos,Racc)
    nbpm=length(bpmpos); ncor=length(corpos);
    Cx=zeros(nbpm,ncor); Cy=Cx;
    for ibpm=1:nbpm
        for icor=1:ncor
            C=CC(bpmpos(ibpm),corpos(icor),Racc);
            Cx(ibpm,icor)=C(1,2);
            Cy(ibpm,icor)=C(3,4);
        end
    end
end
```
Example: orbit correction

Vertical orbit in LHC, before and after correction

J. Wenninger
CAS, 2018
Steering synchrotron beam lines

- steer synchrotron light beam onto experiment
- fix angle at source point
- incorporate in orbit correction by $+L, v_{BPM}, -L$
Dispersion-free steering

- Steering magnets are small dipoles and also affect the dispersion (in ring and linac) besides the orbit.

- Take into account with dispersion response matrix $S_{ij} = \frac{dD_i}{d\theta_j} = \frac{d^2 x_i}{d\delta d\theta_j}$ $(D_i = dx_i/d\delta)$
  - Either numerically or from measurements

- Simultaneously correct orbit and dispersion
  - weight with $\Sigma$s
  - more constraints
  - same number of correctors

\[
\begin{pmatrix}
\vdots \\
\frac{x_i}{\Sigma_i} \\
\vdots \\
\frac{D_i}{\hat{\Sigma}_i}
\end{pmatrix}
= 
\begin{pmatrix}
C_{ij}/\Sigma_i \\
S_{ij}/\hat{\Sigma}_i
\end{pmatrix}
\begin{pmatrix}
\vdots \\
\theta_j \\
\vdots
\end{pmatrix}
\]
Gradient Errors in a Ring

- Add a gradient error (modeled as a thin quad) to a ring with $\mu=2\pi Q$

$$R_Q R = \begin{pmatrix} 1 & 0 \\ -1/f & 1 \end{pmatrix} \begin{pmatrix} \cos \mu + \alpha \sin \mu & \beta \sin \mu \\ -\frac{1+\alpha^2}{\beta} \sin \mu & \cos \mu - \alpha \sin \mu \end{pmatrix}$$

$$= \begin{pmatrix} \cos \mu + \alpha \sin \mu & \beta \sin \mu \\ -\frac{(\cos \mu + \alpha \sin \mu)/f + \gamma \sin \mu}{\cos \mu - \alpha \sin \mu - (\beta/f) \sin \mu} \end{pmatrix}$$

- Trace gives the perturbed tune $\tilde{Q} = Q + \Delta Q$

$$2 \cos(2\pi(Q + \Delta Q)) = 2 \cos(2\pi Q) - \frac{\beta}{f} \sin(2\pi Q)$$

- and if $\beta/f$ is small: the tune-shift is $\Delta Q \approx \frac{\beta}{4\pi f}$

- Gradient errors change the tune!
Changes of the beta function and stop bands

• From $R_{12}$ get the change in the beta function

$$\bar{\beta} = \frac{\beta \sin(2\pi Q)}{\sin(2\pi(Q + \Delta Q))} \approx \beta \left[1 + 2\pi \Delta Q \cot(2\pi Q)\right]$$

$$\frac{\Delta \beta}{\beta} = 2\pi \Delta Q \cot(2\pi Q) \approx \frac{\beta}{2f} \cot(2\pi Q)$$

• Divergences at half-integer values of the tune

• Stability requires

$$\left|\cos(2\pi Q) - \frac{\beta}{2f} \sin(2\pi Q)\right| \leq 1$$

− stop-band width

You’ve seen it before, check out Wolfgang’s slide 96
DIY: preparing the plot

- Choose $d\mu = \beta / 2f$ of the additional quadrupole
- Loop over all tunes $Q_x$ and $Q_y$
- and check whether
  \[
  \left| \cos(2\pi Q) - \frac{\beta}{2f} \sin(2\pi Q) \right| \leq 1
  \]
  in both planes.
- Draw plus sign if the condition is violated.

```matlab
% stopband_quad.m
clear all; close all
dmu=0.1; % beta/2f
hold on
for Qx=-0.05:0.01:1.05;
  tx=abs(cos(2*pi*Qx)+dmu*sin(2*pi*Qx));
  for Qy=-0.05:0.01:1.05
    ty=abs(cos(2*pi*Qy)-dmu*sin(2*pi*Qy));
    if (tx>1) plot(Qx,Qy,'k+'); end
    if (ty>1) plot(Qx,Qy,'k+'); end
  end
pause(0.001)
end
xlim([-0.05,1.05]); ylim([-0.05,1.05]);
xlabel('Q_x'); ylabel('Q_y')
```
Measuring the Tune

• Kick beam and look at BPM difference-signal on spectrum analyzer
  – and divide frequency by revolution frequency gives fractional part of the tune

• Turn by turn BPM recordings and FFT
  – is it Q or 1-Q?
  – change QF and see which way the tune moves

• PLL in LHC: Beam is band-pass, tickle it, and detect synchronously
Tune Correction

- Use a variable quadrupole with $1/f = \Delta k_1 l$

- Changes both $Q_x$ and $Q_y$
  \[ \Delta Q_x = \frac{\beta_{1x}}{4\pi f_1} \quad \text{and} \quad \Delta Q_y = -\frac{\beta_{1y}}{4\pi f_1} \]

- Use two independent quadrupoles
  \[
  \begin{align*}
  \Delta Q_x &= \frac{\beta_{1x}}{4\pi f_1} + \frac{\beta_{2x}}{4\pi f_2} \\
  \Delta Q_y &= -\frac{\beta_{1y}}{4\pi f_1} - \frac{\beta_{2y}}{4\pi f_2}
  \end{align*}
  \]

- Solve by inversion
  \[
  \begin{pmatrix}
  1/f_1 \\
  1/f_2
  \end{pmatrix}
  = \frac{-4\pi}{\beta_{1x}\beta_{2y} - \beta_{2x}\beta_{1y}} \begin{pmatrix}
  -\beta_{2y} & -\beta_{2x} \\
  \beta_{1y} & \beta_{1x}
  \end{pmatrix}
  \begin{pmatrix}
  \Delta Q_x \\
  \Delta Q_y
  \end{pmatrix}
  \]

- Quads on same power supply → sum of betas
Measuring beta functions

• Change quadrupole and observe tune variation

\[
\Delta Q_x = \frac{\beta_{1x}}{4\pi f_1} \quad \text{and} \quad \Delta Q_y = -\frac{\beta_{1y}}{4\pi f_1}
\]

• Need independent power supplies
  – or piggy-back boost supply
  – or a shunt resistor

• May get sums of betas in quads-on-the-same-power-supply.
Model Calibration #1

- Compare measured $\hat{C}^{ij}$ orbit response matrix to computer model $C^{ij}$
  - enormous amount of data $2 \times N_{\text{bpm}} \times N_{\text{cor}}$
- and blame the difference on quad gradients $g_k$ or other parameters $p_l$
  - much fewer fit-parameters $N_{\text{quad}}$ and $N_{\text{para}}$

$$\hat{C}^{ij} - C^{ij} = \sum_k \frac{\partial C^{ij}}{\partial g_k} \Delta g_k + \sum_l \frac{\partial C^{ij}}{\partial p_l} \Delta p_l$$

- First used in SPEAR and later perfected in NSLS → LOCO
Model Calibration #2

• Normally the parameters $p_i$ are BPM and corrector scale errors
  – fit for $N_{\text{quad}}$ gradients and $2 \times (N_{\text{bpm}} + N_{\text{cor}})$ scales

$$\hat{C}^{ij} - C^{ij} = \sum_k \frac{\partial C^{ij}}{\partial g_k} \Delta g_k + C^{ij} \Delta x^i - C^{ij} \Delta y^j$$

• Determine derivatives $\frac{\partial C^{ij}}{\partial g_k}$ numerically by changing a gradient and re-calculating all response coefficients, then taking differences

• BPM-cor degeneracy $\rightarrow$ SVD needed to invert

• Converges, if $\chi^2$/DOF is close to unity
micro-LOCO

- 2 Quads, 2 BPM, 2 COR, only horizontal “C_{12}"
  - ill-defined, but useful to see the structure of matrix
  - gradient errors Δg, BPM scales Δx, corrector scales Δy
- Blame difference on Δg,Δx,Δy

\[
C^{i,j} = R^{i,j} (1 - R^{j,j})^{-1}
\]

\[
\begin{pmatrix}
\hat{C}^{11} - C^{11} \\
\hat{C}^{21} - C^{21} \\
\hat{C}^{12} - C^{12} \\
\hat{C}^{22} - C^{22}
\end{pmatrix}
= \begin{pmatrix}
\frac{\partial C^{11}}{\partial g_{1}} & \frac{\partial C^{11}}{\partial g_{2}} & C^{11} & 0 & -C^{11} & 0 \\
\frac{\partial C^{21}}{\partial g_{1}} & \frac{\partial C^{21}}{\partial g_{2}} & 0 & C^{21} & -C^{21} & 0 \\
\frac{\partial C^{12}}{\partial g_{1}} & \frac{\partial C^{12}}{\partial g_{2}} & C^{12} & 0 & 0 & -C^{12} \\
\frac{\partial C^{22}}{\partial g_{1}} & \frac{\partial C^{22}}{\partial g_{2}} & 0 & C^{22} & 0 & -C^{22} \\
\end{pmatrix}
\begin{pmatrix}
\Delta g_{1} \\
\Delta g_{2} \\
\Delta x_{1} \\
\Delta x_{2} \\
\Delta y_{1} \\
\Delta y_{2}
\end{pmatrix}
\]
DIY: Response matrix analysis

Here: fit for gradients only!
No BPM or corrector scale factors.

The response coefficients of the ideal model
and we mess them up a bit to simulate real data (for example from difference orbits.
reshape brings them in form of a column vector.

Here we numerically calculate the derivatives and store them ‘reshaped’ one column at a time.

Finally we solve the system (SVD is implicitly used in ‘\' if needed)

```matlab
% micro_loco_example.m, fit for gradients only
%
%............ unperturbed response coefficients
[Cx0,Cy0]=response_coefficients(bpmpos,corpos,Racc);

%............ artificially perturb response coefficients
beamline(2,4)=beamline(2,4)+0.2;  % first quad, QF
[Racc,spos,nmat,nlines]=calcmat(beamline); Rend=Racc(:,end);
[Cxhat,Cyhat]=response_coefficients(bpmpos,corpos,Racc);
beamline(2,4)=quadf(1);       % undo perturbation
dCx=reshape(Cxhat-Cx0,[],1);  % store 'measured' as column vector

%.............. calculate the derivatives dC/dg
dcdg=zeros(nbpm+ncor,nquad);
dquad=0.1;       % perturbation
for iquad=1:nquad
    beamline(qbl(iquad),4)=quadf(iquad)+dquad;  % perturb
    [Racc,spos,nmat,nlines]=calcmat(beamline);
    [Cx,Cy]=response_coefficients(bpmpos,corpos,Racc);
dcdg(:,iquad)=reshape(((Cx-Cx0)/dquad,[],1); % column vector
    beamline(:,iquad)=reshape(Cxhat-Cx0, [],1);  % undo
end

%............ find the perturbed quad values
% (should recover the value of 0.2 from above)
quad_changes=dCdgsdCx
```

190913, CAS High Tatras
V. Ziemann: Imperfections and Correction
Experience

- SPEAR: could explain measured tunes to within $4 \times 10^{-3}$ from quadrupole settings which had percent errors (J. Corbett, M. Lee, VZ, PAC93).

- NSLS: LOCO, $\Delta \beta/\beta = 10^{-3}$, dispersion fixed, emittance factor 2 improved (J. Safranek, NIMA 388, 1997)

- and practically every light source since then uses it
Skew-gradient stop bands

• Why are skew-gradient errors bad?
  – they also add stop bands along the diagonals

• Ring with single skew
  – with strength $\sqrt{\beta_x \beta_y / f} = 0.2$

• Calculate the eigentunes
  – Edwards-Teng algorithm

• for each pair $Q_x, Q_y$

• make cross if unstable
  – complex or NAN in Matlab
Measuring Coupling

- BPM turn-by-turn data cross talk, beating

- Closest tune
  - try to make the tunes equal with an upright quad
  - measure tunes
  - coupling 'repels' the tunes
Coupling: mechanical analogy

- Two weakly coupled oscillators: simple to find

  equations of motion

  \[ 0 = m \ddot{x} + (k_x + c)x - cy \]
  \[ 0 = m \ddot{y} + (k_y + c)y - cx \]

- and eigen-frequencies

  \[ \omega^2 = \frac{k_x + k_y + 2c}{2m} \pm \sqrt{\left(\frac{k_x - k_y}{2m}\right)^2 + \frac{c^2}{m^2}} \]

- Minimum tune separation

- Excite one mass, get beating

Translation for accelerator physicists:

- \( x \rightarrow \) horiz. betatr. osc.
- \( y \rightarrow \) vert. betatr. osc.
- \( \frac{k_x}{m} \rightarrow Q_x^2 \)
- \( \frac{k_y}{m} \rightarrow Q_y^2 \) (adj.)
- \( \frac{c}{m} \rightarrow \) coupling source
Coupling correction

- Use a single skew-quad if that is all you have to minimize the closest tune.

- Otherwise build knobs for the four resonance-driving terms with normalized skew gradients

\[ \begin{pmatrix} \text{Re}(F_-) \\ \text{Im}(F_-) \\ \text{Re}(F_+) \\ \text{Im}(F_+) \end{pmatrix} = \begin{pmatrix} \cos(\mu_{x1} - \mu_{y1}) & \cdots & \cos(\mu_{x4} - \mu_{y4}) \\ \sin(\mu_{x1} - \mu_{y1}) & \cdots & \sin(\mu_{x4} - \mu_{y4}) \\ \cos(\mu_{x1} + \mu_{y1}) & \cdots & \cos(\mu_{x4} + \mu_{y4}) \\ \sin(\mu_{x1} + \mu_{y1}) & \cdots & \sin(\mu_{x4} + \mu_{y4}) \end{pmatrix} \begin{pmatrix} \kappa_1 \\ \kappa_2 \\ \kappa_3 \\ \kappa_4 \end{pmatrix} \]

- and empirically minimize each RDT,
  - often $F_-$ (if tunes are close) is sufficient

- Choose phases $\mu$ to make the condition number of the matrix as close to unity as possible.

\[ F_{\pm} = \sum_j \frac{\beta_{x_j}\beta_{y_j}}{2f_j} e^{i(\mu_{x_j} \pm \mu_{y_j})} \]

\[ \kappa_i = \sqrt{\beta_{x_i}\beta_{y_i}/2f_i} \]
Measuring Chromaticity \( Q' \)

- Chromaticity is the momentum-dependence of the tunes: \( Q = Q_0 + Q' \delta \)

- Force the momentum to change by changing the RF frequency. The beam follows, because synchrotron oscillations are stable.

\[
- \frac{\Delta f_{rf}}{f_{rf}} = \frac{\Delta T}{T} = \eta \delta = \left( \alpha - \frac{1}{\gamma^2} \right) \delta \quad \rightarrow \quad \delta = - \frac{\eta \Delta f_{rf}}{f_{rf}}
\]

- Plot tune change \( \Delta Q \) versus \( \Delta f_{rf}/f_{rf} \). The slope is proportional to \( (1/\text{chromaticity } Q') \) [can also use PLL]

\[
Q' = \frac{\Delta Q}{\delta} = - \eta \frac{\Delta Q}{\Delta f_{rf}/f_{rf}}
\]
Chromaticity correction

• Need **controllable and momentum-dependent quadrupole** to compensate or at least change the natural chromaticity $Q' = dQ/d\delta$.

• Momentum dependent feed-down: Use sextupole with dispersion, replace $d_x$ by $D_x \delta$

$$\Delta x' - i\Delta y' = \frac{k_2 L}{2} \left[ (x + iy)^2 + 2D_x \delta (x + iy) + D_x^2 \delta^2 \right]$$

• Linear (quadrupolar) term with effective focal length that is momentum dependent

$$\frac{1}{f_\delta} = k_2 L D_x \delta$$
Chromaticity correction #2

- Momentum-dependent tune shifts

\[
\Delta Q_x = \frac{k_2 L D_x \beta_x}{4\pi} \delta \\
\Delta Q_y = -\frac{k_2 L D_x \beta_y}{4\pi} \delta
\]

- Build correction matrix in the same way as for the tune correction for \( \Delta Q' = \Delta Q/\delta \)

\[
\begin{pmatrix}
\Delta Q'_x \\
\Delta Q'_y
\end{pmatrix} = \frac{1}{4\pi} \begin{pmatrix}
D_{1x} \beta_{1x} & D_{2x} \beta_{2x} \\
-D_{1x} \beta_{1y} & -D_{2x} \beta_{2y}
\end{pmatrix} \begin{pmatrix}
(k_2 L)_1 \\
(k_2 L)_2
\end{pmatrix}
\]

- Invert to find sextupole excitations \( k_2 L \) that add chromaticities to partially compensate the natural
Winding down

• We looked at the sources of all evil, the imperfections,
• and how they affect
  – the orbit
  – the optics
• and figured out how to fix it.
• but there are a few bugs that need special attention...
Bloopers

- LEP vacuum pipe soldering
- Beer bottle in LEP
- Stand-up metal-piece in magnet
- Shielding in SLC DR

- These non-standard 'imperfections' are very difficult to identify, but it is good to keep in mind that even such odd-balls occur.
Extra slides
Skew-gradient Errors in a Ring

• Consider a single skew-quad in a ring

\[ \tilde{S} = \begin{pmatrix} 1 & \tilde{Q} \\ \tilde{Q} & 1 \end{pmatrix} \quad \tilde{Q} = \begin{pmatrix} 0 & 0 \\ \sqrt{\beta_x \beta_y / f} & 0 \end{pmatrix} \]

• and move the perturbation to a reference point

\[ \hat{R} = R_b \tilde{S} R_a = \left( R_b \tilde{S} R_b^{-1} \right) (R_b R_a) \]
Skew-gradient in Ring #2

- Transfer matrix in normalized phase space is rotation, calculate the moved TM

\[
R_b \tilde{S} R_b^{-1} = \begin{pmatrix} O_x & 0 \\ 0 & O_y \end{pmatrix} \begin{pmatrix} 1 & \tilde{Q} \\ \tilde{Q} & 1 \end{pmatrix} \begin{pmatrix} O^t_x & 0 \\ 0 & O^t_y \end{pmatrix} = \begin{pmatrix} 1 & \frac{1}{2} O_x \tilde{Q} O_y^{-1} \\ \frac{1}{2} O_y \tilde{Q} O_x^{-1} & 1 \end{pmatrix}
\]

\[
O_x \tilde{Q} O_y^{-1} = \frac{\sqrt{\beta_x \beta_y}}{f} \begin{pmatrix} \sin \mu_x \cos \mu_y & -\sin \mu_x \sin \mu_y \\ \cos \mu_x \cos \mu_y & -\cos \mu_x \sin \mu_y \end{pmatrix} = \frac{\sqrt{\beta_x \beta_y}}{2f} \begin{pmatrix} \sin(\mu_x - \mu_y) + \sin(\mu_x + \mu_y) & -\cos(\mu_x - \mu_y) + \sin(\mu_x + \mu_y) \\ \cos(\mu_x - \mu_y) + \cos(\mu_x + \mu_y) & \sin(\mu_x - \mu_y) - \sin(\mu_x + \mu_y) \end{pmatrix}
\]

- If many weak skew quads, combine their effect

\[
\hat{R} = (1 + \tilde{P}_1)(1 + \tilde{P}_2) \cdots R_0 \approx (1 + \tilde{P}_1 + \tilde{P}_2 + \cdots) R_0
\]

- Combinations of sines and cosines may add coherently

\[
F_\pm = \sum_j \frac{\beta_{x,j} \beta_{y,j}}{2f_j} e^{i(\mu_{x,j} \pm \mu_{y,j})}
\]