SVD Orbit Correction for ALPHA

T. Satogata

JLAB-TN-14-030

November 10, 2014

This paper describes an orbit correction algorithm using singular value decomposition (SVD), and applications of this correction algorithm to orbit correction in the ALPHA ring at Indiana University.

1 Theory

Consider a beam traveling on a closed orbit in a synchrotron that contains n_b beam position monitors (BPMs) and n_c dipole correctors in a single plane. In the linear approximation, the closed orbit (or bunch centroid displacement) displacement Δx_m at BPM m from a dipole kick change $\Delta x'_k$ at corrector k is given by a response matrix element A_{mk}:

$$A_{mk} \equiv \frac{\Delta x_m}{\Delta x'_k} \tag{1.1}$$

If the Twiss parameters (beta functions $\beta_{m,k}$ and phases $\phi_{m,k}$) at each element and the tune Q in this plane are known accurately, A_{mk} can be calculated:

$$A_{mk} = \frac{\sqrt{\beta_m \beta_k}}{2\sin(Q/4\pi)} \cos(|\phi_m - \phi_k| - Q/4\pi) \tag{1.2}$$

The entire orbit response matrix (ORM) is given by

$$\begin{pmatrix} \Delta x_0 \\ \Delta x_1 \\ \dots \\ \Delta x_{n_b} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n_c} \\ A_{21} & A_{22} & \dots & A_{2n_c} \\ \dots & \dots & \dots & \dots \\ A_{n_b1} & A_{n_b2} & \dots & A_{n_bn_c} \end{pmatrix} \begin{pmatrix} \Delta x'_0 \\ \Delta x'_1 \\ \dots \\ \Delta x'_{n_c} \end{pmatrix} = \mathbf{A} \begin{pmatrix} \Delta x'_0 \\ \Delta x'_1 \\ \dots \\ \Delta x'_{n_c} \end{pmatrix} \tag{1.3}$$

Note that the x' variables here do not have to be the dipole corrector kicks in units of mrad; they can be any variable (such as current in those correctors) that is linearly proportional to each dipole kick. This is particularly useful when calculating an orbit correction scheme based on measured orbit response, where the independent variables for the kicks are currents in the correctors.

Linear orbit correction reduces to inverting the (usually non-square) $\mathbf{A}$ matrix, or solving this system of linear equations with a linear least-squares method. One then has a set of equations that give the set of corrector changes necessary to produce a given orbit change on all BPMs. Putting in the negative of the measured BPM positions for this given orbit change will give a set of corrector strengths that make all BPM positions zero and thus correct the orbit.

Traditional linear algebra courses teach methods to create square matrix inverses, but don't teach the more general approach to finding an optimal solution for a non-square linear problem.

Singular value decomposition (SVD) is the most common approach to producing a "pseudo-inverse" for $\mathbf{A}$. Background information is in the corresponding Wikipedia article [1] but the approach is fairly straightforward and well-supported by standard linear algebra environments like Mathematica, Matlab, Octave, Perl PDL, and Python numpy.

The process is:

1. Simulate or measure all elements of the $\mathbf{A}$ response matrix. They can be in any reasonable units (mm/mrad, mm/A, etc) as long as the orbit response to the corrector strength is linear.

2. Use a linear algebra package to calculate the three matrices of the singular value decomposition of $\mathbf{A}$: $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\star$

3. $\mathbf{U}$ and $\mathbf{V}$ will each orthogonal matrices, while $\mathbf{\Sigma}$ will be a diagonal matrix with non-negative real numbers on the diagonal with descending magnitude.

4. If $\mathbf{A}$ was near degenerate, $\mathbf{\Sigma}$ will have some diagonal terms that are “near zero”. (Judgment of how close is “near zero” is the only judgement call in SVD tuning.) Set the near zero terms to zero before the next step.

5. Calculate $\mathbf{\Sigma}^+$, the pseudoinverse of $\mathbf{\Sigma}$, by replacing all non-zero diagonal terms s_i of $\mathbf{\Sigma}$ with their inverses, $1/s_i$. The previous step eliminated very small s_i that would produce very large $1/s_i$ in this step.

6. Construct the pseudoinverse of $\mathbf{A}$: $\mathbf{A}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^\star$. This matrix depends on the focusing (quadrupole strengths) of the machine but does not depend on the orbit to be corrected or absolute corrector strengths.

7. Measure the orbit at each BPM, $x_0, \ldots, x_{n_b}$.

8. The least-squares solution for the orbit corrector strengths is given by using $\mathbf{A}^+$ for the inverse of $\mathbf{A}$ in (1.3) to calculate the corrector changes necessary to remove the measured orbit offsets.

$$\begin{pmatrix} \Delta x'_0 \\ \Delta x'_1 \\ \cdots \\ \Delta x'_{n_c} \end{pmatrix} = \mathbf{A}^+ \begin{pmatrix} -x_0 \\ -x_1 \\ \cdots \\ -x_{n_b} \end{pmatrix} \tag{1.4}$$

2 Example: Overconstrained Vertical Orbit Correction

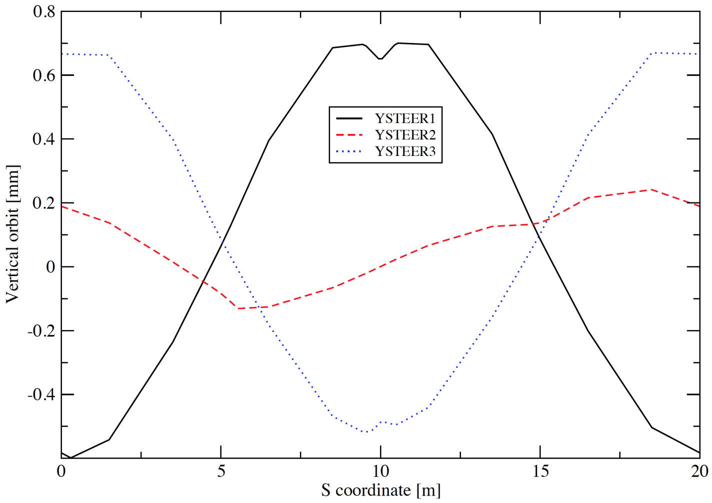

Figure 1: Model vertical orbit distortion in ALPHA ring for 0.1 mrad vertical steerer kicks.

Using the `alpha_18-Aug-2014.lte` lattice with `RHOW=1`, `Qstrength13=-1.5`, and `Qstrength2=3`, I calculated the vertical response of all eight BPMs to the three vertical steerers YSTEER1 through YSTEER3.

These results are:

$$\begin{pmatrix}\Delta x_0\\ \Delta x_1\\ \dots\\ \Delta x_8\end{pmatrix} = \begin{pmatrix} -5.546918 & 1.467595 & 6.632193\\ -2.151922 & 0.0801527 & 3.771348\\ 3.726690 & -1.263585 & -1.644588\\ 6.867813 & -0.6156072 & -4.735577\\ 6.964935 & 0.6230723 & -4.461127\\ 3.926187 & 1.267806 & -1.402687\\ -1.819361 & 2.103730 & 3.915262\\ -5.174599 & 2.321665 & 6.692933 \end{pmatrix} \times 10^{-3} \begin{pmatrix}\Delta x'_0\\ \Delta x'_1\\ \dots\\ \Delta x'_8\end{pmatrix} = \mathbf{A}\begin{pmatrix}\Delta x'_0\\ \Delta x'_1\\ \Delta x'_2\end{pmatrix} \tag{2.5}$$

where the units of the A_{mk} are m/rad or mm/mrad. Orbits for 0.1 mrad kicks are shown in Fig. 1.

The SVD decomposition for this response matrix is

$$\mathbf{U} = \begin{pmatrix} -0.46789397 & 0.23102221 & 0.1683598\\ -0.22191568 & 0.1603133 & 0.56501912\\ 0.21384283 & 0.12221096 & 0.71791789\\ 0.44483362 & 0.2551266 & 0.21327959\\ 0.4304547 & 0.45675924 & -0.18519678\\ 0.19610979 & 0.49888193 & 0.17817977\\ -0.22752731 & 0.47762338 & -0.13407184\\ -0.46119842 & 0.39403935 & -0.086561938 \end{pmatrix} \tag{2.6}$$

$$\mathbf{\Sigma} = \begin{pmatrix} 18.5826 & 0.000 & 0.000\\ 0.000 & 4.7360 & 0.000\\ 0.000 & 0.000 & 2.0011 \end{pmatrix} \times 10^{-3} \Rightarrow \mathbf{\Sigma}^{+} = \begin{pmatrix} 0.05381 & 0.000 & 0.000\\ 0.000 & 0.21115 & 0.000\\ 0.000 & 0.000 & 0.4997 \end{pmatrix} \times 10^{3} \tag{2.7}$$

$$\mathbf{V} = \begin{pmatrix} 0.72613099 & 0.59400829 & 0.34624837\\ -0.122754 & 0.60750378 & -0.78477424\\ -0.6765096 & 0.52734552 & 0.51404422 \end{pmatrix} \tag{2.8}$$

Note that no cuts of the $\mathbf{\Sigma}$ matrix were necessary; the correctors are acceptably orthogonal. Here the orbit correction is also overconstrained, so the "corrected" orbit will not be zero in all the BPMS but will be minimized in a least squares sense. Weighting can be added to this scheme in a straightforward way if one desires stronger orbit correction at some BPMs than others. Perl code using PDL for the SVD calculation is in Appendix 1.

The pseudoinverse of $\mathbf{A}$ is

$$\mathbf{A}^{+} = \mathbf{V}\mathbf{\Sigma}^{+}\mathbf{U}^{\star} \tag{2.9}$$

and is a 3×8 matrix. Multiplying this by $\mathbf{A}$ does not give a unit matrix since the problem is overconstrained; hence the term "pseudoinverse".

For a test, I calculated corrector settings necessary to change all vertical BPMs by +1.0 mm. These corrector settings are VSTEER1=0.509 mrad, VSTEER2=-0.090 mrad, and VSTEER3=0.570 mrad. Inserting these into the elegant lattice produces the results shown in Fig. 2, where the eight BPMs have an orbit of (0.989±0.109) mm. One can calculate the Hessian or second order response of the RMS to each corrector strength change to confirm that this fit is the closest fit possible with these three correctors in a least-squares sense.

3 Example: Actual Horizontal Orbit Correction

Horizontal orbit response data was acquired from Chandra Romel on Sep 12, 2014, and analyzed by Alex Dzierba and Tom RInckel. Alex shared analysis data on Sep 23 that was later corrected in email on October 23. The raw horizontal response data from the `TrimpRampData.xls` file included in the Oct 23 email

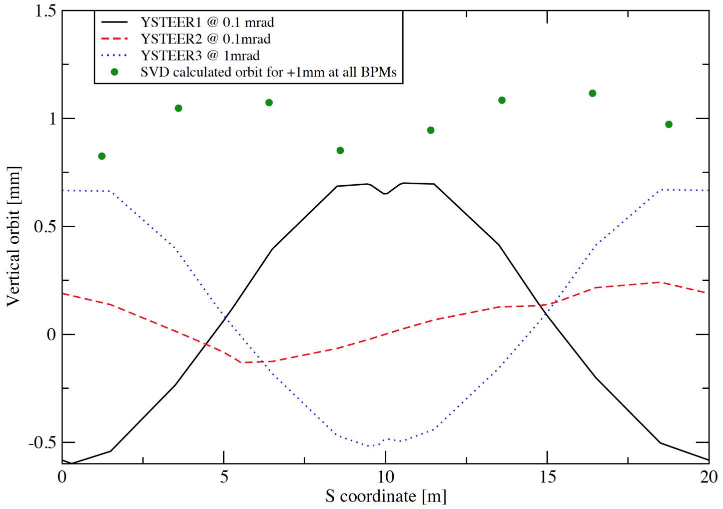

Figure 2: Model vertical orbit distortion in ALPHA ring for 0.1 mrad vertical steerer kicks, and SVD calculated orbit for desired correction to +1.0 mm at all BPMs

constitutes a measured horizontal response matrix, with units of mm/A:

$$\begin{pmatrix} \Delta x_0 \\ \Delta x_1 \\ \dots \\ \Delta x_8 \end{pmatrix} = \begin{pmatrix} -0.860064 & 2.17828 & -4.09971 & 2.7196 \\ -0.992726 & 3.81669 & -5.10908 & 2.25453 \\ 4.0169 & -1.40878 & 2.57946 & -4.99278 \\ 2.59835 & -1.10153 & 3.78762 & -4.81224 \\ -5.28003 & 3.86681 & -1.35513 & 2.7202 \\ -5.12252 & 2.50716 & -1.51811 & 4.07757 \\ 2.22105 & -4.97991 & 4.02543 & -0.910331 \\ 3.03123 & -4.16271 & 2.42819 & -0.752286 \end{pmatrix} \begin{pmatrix} \Delta I_1 \\ \Delta I_2 \\ \Delta I_3 \\ \Delta I_4 \end{pmatrix} \tag{3.10}$$

The pseudoinverse of this matrix can be computed using the same approach as the previous section. The Σ matrix elements range from 17.1 down to 0.28, indicating that there is mild but not excessive degeneracy in the four horizontal trims in correcting the orbit at the eight BPMs. I did not eliminate any eigenvalues in the following calculation of the pseudoinverse:

$$\mathbf{A}^{+}[\mathrm{A/mm}] = \begin{pmatrix} 0.46163 & 0.796076 & 0.0819633 & 0.890064 & 0.452940 & 0.348161 & 0.240058 & 1.05145 \\ 0.41982 & 0.796165 & 0.107418 & 1.00444 & 0.594896 & 0.424611 & 0.18407 & 0.995438 \\ 0.343008 & 0.656336 & 0.0586843 & 0.940606 & 0.579074 & 0.448257 & 0.271754 & 0.995296 \\ 0.449313 & 0.752229 & -0.006822 & 0.848581 & 0.518853 & 0.445454 & 0.303719 & 1.089499 \end{pmatrix} \tag{3.11}$$

This matrix should be multiplied by a column vector with the desired changes in BPM positions (e.g. -1 times the measured position to correct to zero position) to get required corrector setting changes in Amps. Perl code used to calculate this pseudoinverse, that can also be used to calculate the corrector currents, is in Appendix 2.

References

[1] “Singular Value Decomposition”, Wikipedia article, `http://en.wikipedia.org/wiki/Singular_value_decomposition`.

4 Appendix 1: Vertical SVD Calculation

```
#!/usr/bin/env perl
######################################################################
use PDL;

######################################################################
# BPM response in mm/mrad or m/rad from elegant simulations
######################################################################
$a = pdl [[-5.546918, 1.467595, 6.632193 ],
          [-2.151922, 0.0801527, 3.771348 ],
          [3.726690, -1.263585, -1.644588 ],
          [6.867813, -0.6156072,-4.735577 ],
          [6.964935, 0.6230723, -4.461127 ],
          [3.926187, 1.267806,-1.402687 ],
          [-1.819361, 2.103730, 3.915262 ],
          [-5.174599, 2.321665, 6.692933 ] ];

######################################################################
# SVD is U.S.V*, pseudoinverse is V.S+.U*
######################################################################
($u,$s,$v) = svd($a);

print "Initial response matrix: a=$a\n";
print "SVD u=$u\n";
print "SVD s=$s\n";
print "SVD v=$v\n";

$ut = transpose($u);

$sstar = pdl [[1/$s->at(0,0),0,0], [0,1/$s->at(1,0),0], [0,0,1/$s->at(2,0)]];

######################################################################
# Calculate pseudoinverse
######################################################################
$pinv = $v x $sstar x $ut;

######################################################################
# Target orbit deviation is +1mm at all BPMs
######################################################################
$dx = pdl [[1,1,1,1,1,1,1,1]];

$corrs = $pinv x transpose($dx);

print "corrector strengths [mrad]: $corrs\n";
```

5 Appendix 2: Horizontal SVD Calculation

```
#!/usr/bin/env perl
######################################################################
use PDL;

######################################################################
# Measured mm/A response from Sep 12 2014 ALPHA measurements
######################################################################
$a = pdl [[-0.860064,2.17828,-4.09971,2.7196],
          [-0.992726,3.81669,-5.10908,2.2545],
          [4.0169,-1.40878,2.57946,-4.99278 ],
          [2.59835,-1.10153,3.78762,-4.81224],
          [-5.28003,3.86681,-1.35513,2.7202 ],
          [-5.12252,2.50716,-1.51811,4.07757],
          [2.22105,-4.97991,4.02543,-0.91033],
          [3.03123,-4.16271,2.42819,-0.75228]];

######################################################################
# SVD is U.S.V*, pseudoinverse is V.S+.U*
######################################################################
($u,$s,$v) = svd($a);

print "a=$a\n";
print "u=$u\n";
print "s=$s\n";
print "v=$v\n";

$ut = transpose($u);

$sstar = pdl [[1/$s->at(0,0),0,0,0],[0,1/$s->at(1,0),0,0],
              [0,0,1/$s->at(2,0),0],[0,0,0,1/$s->at(3,0)]];

print "sstar=$sstar\n";

$pinv = $v x $sstar x $ut;

print "pinv=$pinv\n";

######################################################################
# Target orbit deviation in mm at all BPMs
######################################################################
$dx = pdl [[1,1,1,1,1,1,1,1]];

$corrs = $pinv x transpose($dx);

print "corrector strengths [A]: $corrs\n";
```