Homework #3 for Beam-based Diagnostics
USPAS Knoxville, TN, January 2019

Christoph Steier, LBNL/ALS-U

James Safranek, SLAC/SSRL

Xiaobiao Huang, SLAC/SSRL

Computer Homework #3: Vertical Orbit Correction

Given the following middle layer commands:

```
y = getam('BPMy');          % Vertical orbit at BPMs (result is vector of length num_bpmy)
Ry = getrespmat('BPMy', 'VCM');      % Get the vertical BPM response matrix
                    % Matrix size is num_bpmy*num_vcm
num_bpmy = length(getam('BPMy')); % get the number of vert. BPMs
num_vcm = length(getam('VCM'));    % get the number of vert. correctors
vcm = getsp('VCM');                 % Get the vertical corrector setpoints
setsp('VCM', vcm);                  % Set the vertical correctors (absolute)
stepsp('VCM', deltavcm);            % Increment the vertical correctors
f1=figure;f2=figure                 % opens two figure windows
figure(f1)                          % switches to figure window f1 for plot commands
subplot(2,1,1)                      % subdivides figure window into 2 plots and plots
to the first one
```

Computer Homework #3: Vertical Orbit Correction

To solve the following homework problem, please also consult the computer class from Monday: monday_orbit.mlx, cells 9-10 (if you cannot find it, you can get it at http://als.lbl.gov/als_physics/csteier/uspas19/compuclass/), as well as the orbit correction lecture from Tuesday. They include examples similar to this homework (main commands summarized briefly below):

```
[U, S, V] = svd(Ry, 0);          % SVD decomposition of response matrix Ry
Ivec = 1:55;                     % we used 55 singular values in Monday example
                                 % in this homework we will use 1-74 (=#VCMs)


% Find the corrector changes to correct back to reference orbit (with # SVs in Ivec)


DeltaVCM = -V(:,Ivec) * S(Ivec,Ivec)^-1 * U(:,Ivec)' *  (y-y0);   % y0 is the reference orbit
```

Homework #3 – SVD orbit correction

Important: At the time the Matlab Middle Layer setup you are using was copied to a memory drive a few years ago, 3 of the BPMs in ALS were 'broken'. So they are not included in the response matrix that getrespmat returns (instead you get NaN – i.e. not a number entries). This would break the Matrix inversion we do for orbit correction. So you first have to disable them (in x and y):

```
>> setpv('BPMx','Status',0,[3 3;4 2;9 8]); % Setting Status=0 excludes device
>> setpv('BPMy','Status',0,[3 3;4 2;9 8]);
```

Write a Matlab script (for the ALS, i.e. use setpathals first) **to**:

1. Create a random orbit error with the following commands (before starting set the correctors to zero and take a reference orbit, which you later correct to):

   vcm_err = 0.1 * randn(num_vcm,1);   % num_vcm vertical correctors at the ALS

   stepsp('VCM', vcm_err);   % rms 0.1 A kick (powersupplies are 40-120 A)

For the following parts, you can use 'for' loops in Matlab (>>help for), as well as the possibility to add new data to a vector, like yrms(loop)=std(getam('BPMy')-y0) with loop being incremented inside the for loop.

Homework #3 – Dependence on SVD

2. Correct the orbit back to the reference orbit you took without errors (there are BPM offsets, so do not correct to 0, but to the starting orbit withouterrors) using various numbers of SVs (Ivec in the monday_orbit.mlx example – Start with one up to the maximum number possible, which is given by the number of VCM, i.e. 74):

  a. Plot the singular values (i.e. the diagonal elements of the matrix S after the SVD inversion) of the response matrix (using semilogy – which works similar to plot, just with log y-scale)

  b. Plot the remaining RMS and Max orbit error (std and max(abs)) vs. # of the singular values used in the orbit correction.

    i.e., plot the orbit error when 1 sv is used, 1:2 sv, 1:3 sv, 1:4 sv, etc. (up to num_vcm)

    Important: Do not forget to remove previous correction before applying the new one!

  c. Plot the total corrector strength change (sum(abs(DeltaVCM))) as well as total strength plus initial error (sum(abs(DeltaVCM+vcm_err))) vs.

    # of the singular values used.

  d. How many singular values would you recommend using?

3. Now remove the random orbit error (removing what you did in 1.) and only put one big error in: Set any one of the VCMs for example to 1 A.

4. Repeat step 2. a-d above for this single error. Does the 'optimum' number of SVs change?