Modeling the SPS Feedback and Feedforward Systems for Improved Performance

Jake Hargrove

May 24, 2016

Abstract

The Super Proton Synchrotron (SPS) is the last link in the chain of accelerators providing protons to the Large Hadron Collider (LHC). The SPS is currently the limiting factor on the maximum number of protons and thus collisions in the LHC. The SPS upgrade is under way to expand the discovery potential of the LHC. The accelerating system — Radio Frequency (RF) — is being improved. Models of the SPS RF feedback systems were developed. These models could assist with design choices, evaluating the upgraded system performance, and anticipate limitations and issues.

Intro

CERN has the largest and most powerful particle accelerator in the world, the LHC, or Large Hadron Collider (LHC). High-energy particle beams are boosted through a series of accelerators and into the LHC to velocities close to the speed of light, where they then are meant to collide. The collisions achieve energies of 13 TeV at the collision point. Currently, the Super Proton Synchrotron (SPS) is the limiting factor on the maximum number of collisions and is the last accelerator particles are sent through before entering the LHC. Therefore, if the SPS were to be optimized further, the performance of the LHC could be improved possibly achieving more important discoveries in accelerator physics quicker. The SPS uses a feedback system to control the accelerating cavity voltage. The cavity is operating at the Radio Frequency (RF) part of the spectrum. Using a Matlab and Simulink model, the SPS can be tested to extremes in all parameters without danger, and an optimal ratio between RF station stability and beam performance can be found.

The SPS

The SPS is made up of two cavity groups, four 200MHz cavities and four 800MHz cavities. An RF generator launches a wave that propagates the cavity in the same direction as the accelerating particles at the desired voltage frequency. The particles being accelerated are protons, and being moving charges, create a current and their own electric field. This causes perturbations in the RF voltage, in amplitude, phase, and/or frequency. To combat this effect and keep the system stable, the SPS has feedback and feedforward systems.

In an input-output system, where some perturbation or noise has been added to the input and distorted it in some way, a feedback system can be added to make the input and output values match. At its most basic, a feedback system works by finding the error between the output and desired value, i.e. subtracting, and adding that value to the input, making the next output closer to the desired. For example, consider the system in figure 1. A simple sign wave is added to a noise generator causing a distorted wave in the output figure 2. Figure 3 introduces the simple feedback method introduced earlier. The job of this system is to preserve the original sine wave, in other words match the input and output. The success of this system can be seen in figure 4.

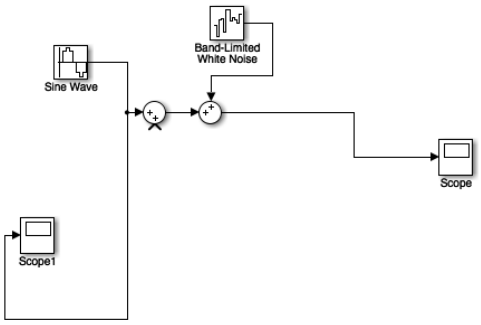


Figure 1: A Simulink system in which a sine wave is added to random noise.

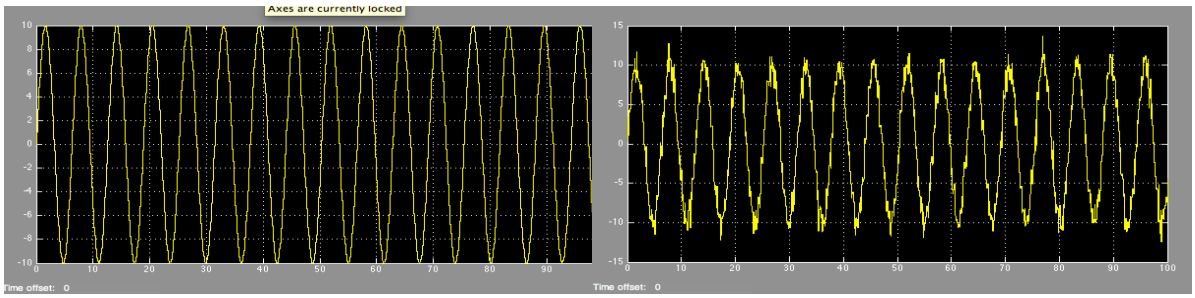


Figure 2:
Left: the inputted sine wave from the system in figure 1. Right: The result of the noise being added to that sine wave.

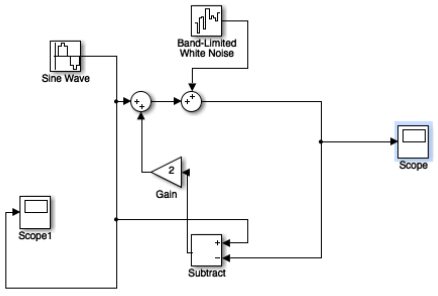


Figure 3: A simple feedback loop attached to the system from figure 1. The output is subtracted from the desired inputted sine wave and added to the input.

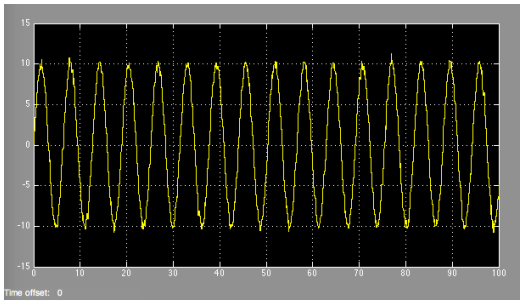
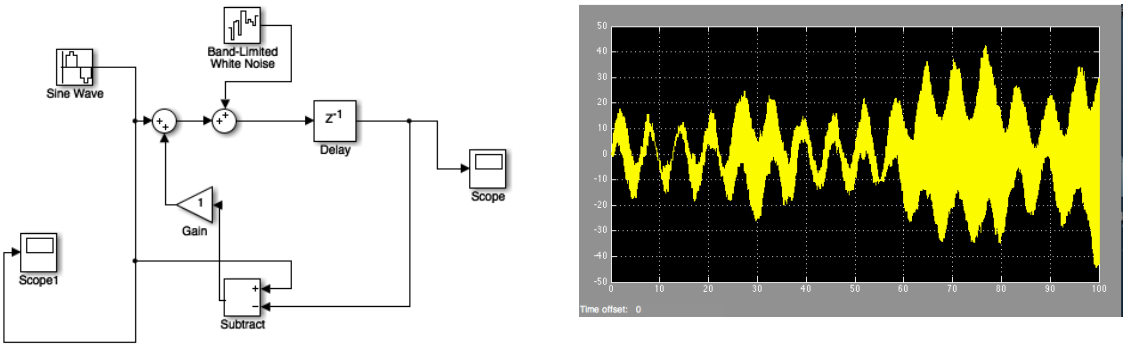


Figure 4: The output from figure 3, note that much of the noise from figure 2 has been eliminated.

Real life feedback systems are usually more complicated, the SPS feedback system being one of them. The SPS feedback works to keep the cavity voltage at the constant desired value. The equipment in the system comes with a natural delay meaning the feedback isn't applied immediately as seen before. The presence of delay minimizes the maximum allowed gain. The system gain scales the correction being made to the input of the system. Smaller gain means a smaller correction and thus a longer time to achieve equilibrium. In the simple example from figure 3, gain is multiplied to the difference between output and desired value. A gain of two corrects the signal adequately. However, as seen in figures 5 and 6, the introduction of a delay causes instability even for a gain of one. While higher gain reduces the time it takes a system to reach the desired value, too much gain will cause immediate instability.



Figures 5(left) and 6(right): With an added delay of 1 to the system, the output becomes unstable even with the gain lowered from 2 to 1.

The SPS accomplishes this through an RF control system. An RF feedback system works to stabilize the effect of the beam traveling through the cavity. The beam stability is modeled in the time domain. It corrects the signal by comparing it to a desired value, finding the error and applying a gain and adding the correction back to the input. To keep the system stable, the frequency and phase of that voltage need to stay as close to those original values as possible as the beam passes through. The signal coming into the cavity is not as simple as our previous example, however. It contains a range of different amplitudes, frequencies and phases. Therefore, a single correction will not work for all frequencies that are produced by the beam. Different feedback gain and phase might be required for different frequencies. For our system a relatively low gain is optimal for system stability. Eventually, the corrections will add up and finally reach the desired value. Gain can be increased in small increments to reach the desired voltage quicker, but just as with the simple example above, once over a certain value, the gain will completely destabilize the system.

Along with the RF feedback, the SPS also employs a feedforward correction. The main difference being that the feedforward system does not act on the perturbation, it anticipates it and acts before it arrives. Therefore the feedforward acts once per turn.

After the first turn of the protons, the feedforward system notes the error, and before the next time the bunch comes around, it acts and corrects the system. For example, take a straight horizontal line to be a system. Imagine a periodic disturbance creates an upward spike in the line every 10 seconds. A feedback system would be able to correct this, but only after each spike has occurred. A feedforward system would see the first spike and measure the size of the disturbance. Then in 10 seconds as the next disturbance came around, the feedforward would create a downward spike of equal size; effectively canceling out the disturbance and keeping our line straight. This is how the feedforward in the SPS works. It anticipates the beam and will cause a disturbance in the opposite direction of the beam, lessening the work of the feedback. Since the beam doesn't create the same perturbation every turn, a combination of feedback and feedforward gives the system optimal stability.

The Frequency and Time Domains

The ultimate goal of this project is to find the RF feedback system parameters that will lead to the best possible performance of the SPS. For our purposes performance is measured in two ways, the stability of the beam and the stability of the RF system itself. The stability of the proton beam is best measured in the time domain. A stable beam is characterized by a steady phase of the proton bunches. Meaning the phase of all the bunches are basically the same as they move through the cavity. For our system, stability can be estimated by plotting the beam phase as a function of time, where stability is measured in peak-to-peak phase difference. A smaller difference means a more stable beam and vice versa.

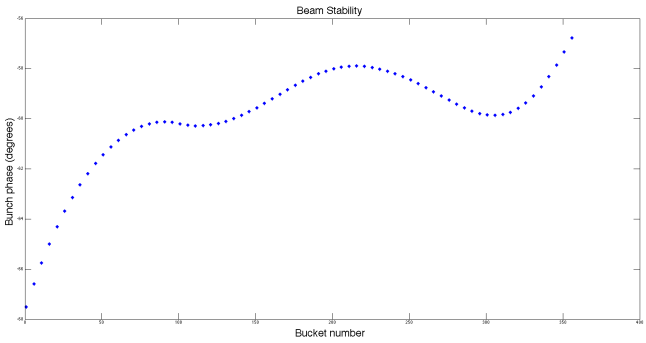


Figure 7: The phase of the proton bunches as they move through the cavity. The peak-to-peak difference is about 10 degrees, which — even though stable — is not optimal.

The stability of the RF system is measured in the frequency domain. In the frequency domain, a system's response is measured for different frequencies, rather than how it responds in time. Matlab models were used to determine the stability of the RF system from its estimated frequency response. The frequency response is the complex ratio between system input and output at all frequencies. These models include the accelerating cavity, high power generators, and the feedback system. The two main tools in Matlab to estimate stability in the frequency domain are Bode and Nyquist plots. Each of these methods quantify stability through the gain and phase margins.

Radio Frequency System Stability

Bode plots are used to represent the frequency response of a system and offer two measures of stability. One measure is known as the gain margin. By plotting amplitude against frequency the response of the signal can be seen over all frequencies. The gain margin refers to the part of the plot where the phase is 180°. If the magnitude is greater than or equal to one at this point, then the system is unstable. In other words, it is stable if the magnitude is less than one at this point. The phase margin refers to the Bode plot of phase against frequency. To check stability in the phase margin, locate the point where the gain is equal to 1. The phase at that point subtracted from 180 gives the phase margin.

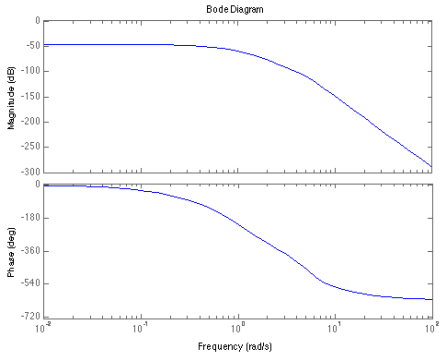


Figure 8: A typical Bode plot, the top is amplitude, or magnitude, against frequency and the bottom is phase against frequency.

The other method of determining stability is through a Nyquist plot. A Nyquist plot represents the frequency response of a feedback system as a contour/polar plot. In polar coordinates the radial component is the gain while the angular component represents the phase. The frequency response of the system usually resembles a cardioid. As with the Bode plots, there are two measures of stability, a gain and phase margin. First, the gain margin refers to the distance between the point the plot crosses the real axis, and -1 on that axis. If the plot encircles -1, it is considered unstable. In other words, gain can be increased and still have a stable system response, until the plot reaches -1. The phase margin deals with the left half of the plot as well and refers to how close the plot comes to the real axis. The angle between the lowest point of the plot and the real axis is known as the phase margin. As long as the plot stays above the axis, the system is considered stable. The higher the gain and phase margins, the more stable the system. The Matlab model of the RF system shows significant gain and phase margins.

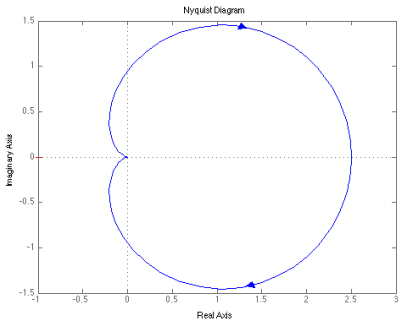


Figure 9: A typical Nyquist plot.

Creating the Simulink model

A Simulink model of the 200MHz feedback system was created. As seen in the figure 9, Simulink is used to model the SPS as a block diagram. The cavity voltage V_0 is first fed into the RF block, the feedback system, then through the inductive output tube (IOT), which is a high power generator, and finally the cavity. This gives the behavior of the cavity with no beam present. The lower portion of the model introduces the beam into the system. It samples the output of the cavity and its resulting voltage perturbation is then added to the V_0 input. The beam is made up of packets of protons that circle the SPS. The proton packets pass through and are detected by the cavity periodically. This allows the feedback system to act on the cavity every turn, by predicting the arrival of the protons. The feedback system includes low and high pass filters. The simulation will run for a predetermined number of turns. The stability and performance of the simulation can be checked through the block labeled scope, or by plotting the data outputted as simout.

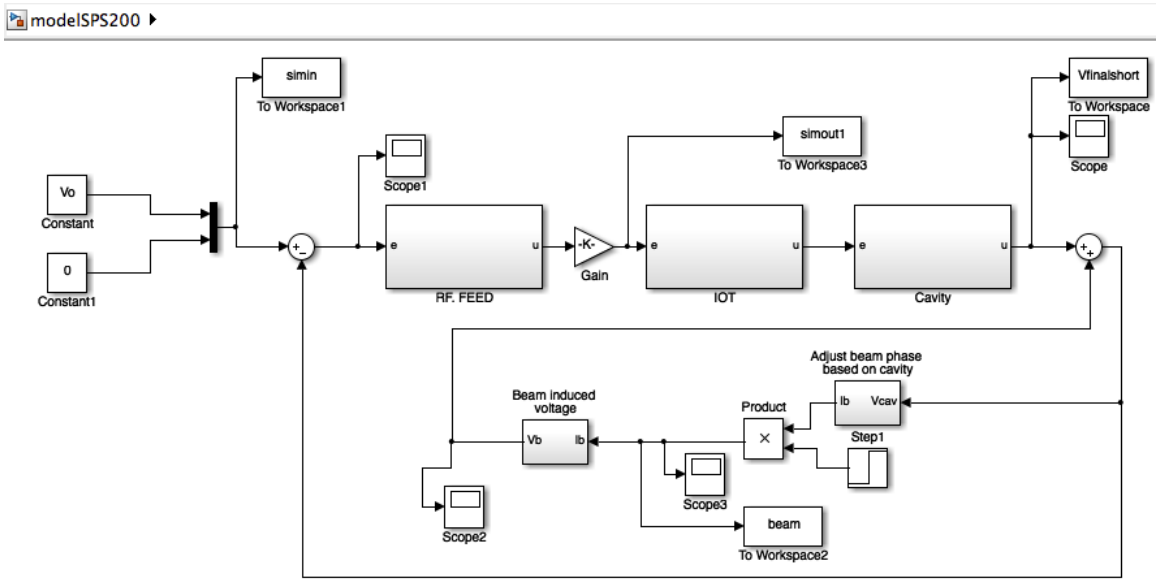


Figure 10: The complete Simulink block diagram of the short cavity of the 200MHz feedback system.

The 200MHz RF system consists of 4 cavities, two short cavities and two long cavities. The figure above is the Simulink model for the short cavity. Each pair of cavities is modeled by a "macrocavity" with the same characteristics but twice the voltage. It is beneficial for verification purposes to build each model (short and long cavities) separately, to be combined later. The Simulink model does not work on its own, it requires a Matlab script to describe the variables. The initialization file can be constructed using the same principle but requires much more finesse. The file is split into sections corresponding to the block diagram: Feedback, IOT, and Beam, and was constructed in pieces. There are also a few parameters from the SPS, and parameters for

the cavity that need to be considered. Using the new initialization file and Simulink model together, the parameters were adjusted to get a stable response on the scope, completing the 200MHz short cavity model.

The Long cavity was built through the same process. All of the blocks needed to be modified to fit the long cavities parameters. After creating a variable for the new length in the initialization file, the IOT, RF feedback, and Cavity sections each needed their own 'Long' versions. Namely, changing the length of the cavity changes the way the system behaves, which needed to be represented. This was accomplished by creating a new set of parameters to be fed into the transfer functions of each block. For example, the cavity block has a transfer function that accepts the variables CavNum and CavDen that act as the numerator and denominator of the transfer function. For the long cavity, CavNumLong and CavDenLong were defined in the initialization file, and fed into the cavity block for the Long model. Adding long parameters to the initialization file for the RF and IOT in the same fashion as the Cavity completes the Long model. Now both the Short and Long cavity models can be run from the same initialization file. The beam runs through both cavities every turn, however, meaning to match the real system, our model needs to run both cavities simultaneously. This led to the full model of the SPS 200MHz system with the feedforward off. Both short and long systems are summed with the induced voltage of the beam caused by each cavity and summed with their respective initial voltages. Now the output, labeled Vtotal, accurately represents that of the SPS with the feedforward off.

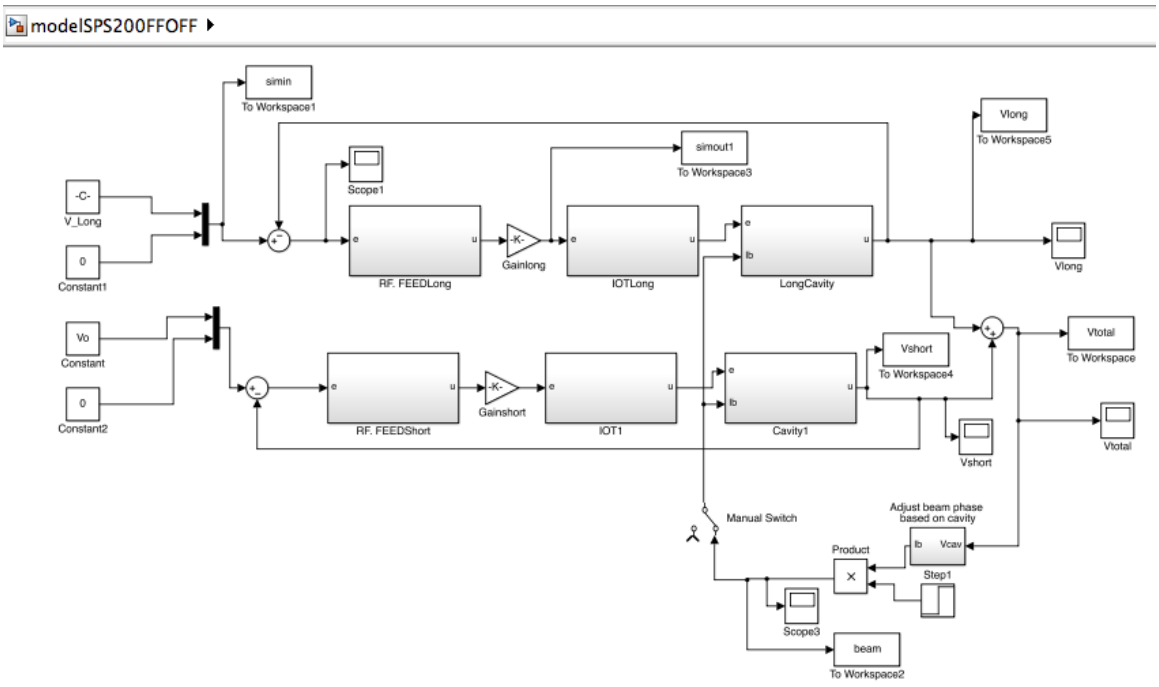


Figure 11: The full Simulink model for the 200MHz model with the feedforward off. Note that the effect of the beam has been brought inside the cavity blocks.

Matching Data

Now that a working model of the SPS system with the feedforward off has been constructed with wide margins of stability for both the beam and the RF system, it can be used for analysis of the performance. First, the current conditions of the SPS need to be replicated in the model as the starting point for any possible modifications. Data taken from the long and short cavities, and a combination of the two, at the SPS facility at CERN can be used to compare to the response of the model. As expected with a real system, the data contains a lot of noise. Therefore the model will never perfectly match the real data, but it should be possible to replicate the overall response. This requires the creation of a new Matlab script.

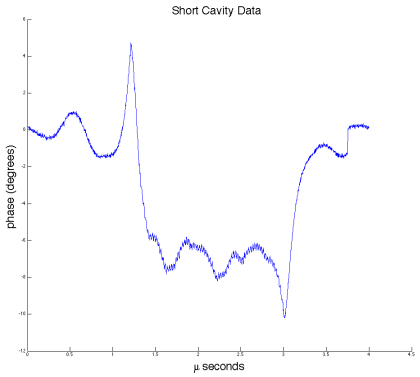


Figure 12: The response of the short cavity with the feedforward off from the SPS at CERN.

The data can be brought into files through the load function, and with a few modifications can be plotted against time. The same can be done through our simulation by putting blocks in the Simulink model to send data to the workspace. Essentially, creating a 'send to workspace' block at the end of the short cavity, for example, labelled Vshort, saves the data from the short cavity in the simulation as the variable Vshort. From there it can be plotted just like data from the SPS. As it will take many runs to get an adequate match with the simulation, the save function will prove to be very handy. There are many parameters to adjust that have some effect on the shape of the response. Each cavity has a loop gain, which is an overall gain, a high pass gain, a low pass gain, a loop delay, and finally a delay between low and high pass filters. That makes five variables per cavity, or ten overall. It is much easier to analyze one cavity at a time, and to focus on one group of parameters at a time, gain and delay.

As it is not possible to analytically determine the optimal parameters, it is helpful to use a systematic approach. Through adjusting each parameter while holding the others constant, it can be shown that gain has a much more dramatic effect on the system output than the delay. Therefore, it is easier to start with gain, and move to the more minute changes in delay later. More specifically the overall gain has the biggest effect. Starting with the short cavity, first the overall gain, or loop gain, was analyzed. The loop gain determines the amplitude of the signal. In other words a higher gain will result in flatter response with less dramatic oscillations. Matching the peak to peak phase difference of the SPS data to the simulation is the first step to finding a match. First, the values need to

be found of the highest or lowest gain where the system response is still stable. This defines a range to stay within while trying to accurately represent the data. Taking small steps in gain, 15 simulations were run and saved over the gain range. The runs were then plotted and measured for their peak to peak phase difference. Comparing this to the peak to peak phase difference of the data narrowed down the desired gain to a much narrower window. From that point small tweaks were made to get the best match. This process was then repeated for the long cavity.

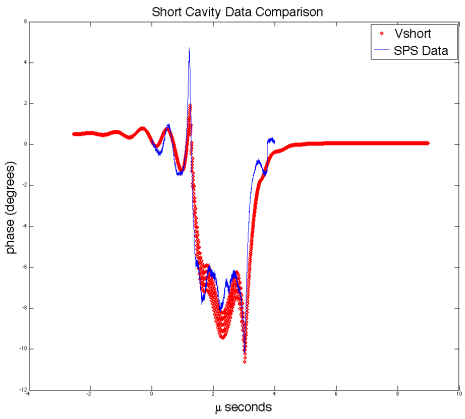


Figure 13: Example of matching the SPS data with the Simulink model for the short cavity. Blue is the SPS data while red is the output of the model, Vshort from figure 11.

Next, to match the shape of the perturbations, the high pass and low pass gains need to be adjusted. The ratio between high pass and low pass gives the signal its shape. To find the optimal gains for the short cavity, the ratio was adjusted and plotted against the short cavity SPS data. Plotting the simulation on top of the data, close matches can be found, an example of this process is seen in figure 12. As always the same was carried out for the long cavity. The best fits for the short cavity for high and low pass gains were used from earlier analysis with good agreement. The long cavity had less success in gain analysis which will be remedied when the delay is analyzed. The data, for both cavities, won't fully match until the delay is dealt with. Using a similar process with the delay, and a few more tweaks of the gain, the simulation will come closer to attaining an accurate replication of SPS.

Sensitivity Analysis

To match the Matlab simulation to the SPS data, there are five parameters that need to be considered. Each has a unique effect on the output of the system. Some of the parameters effects overlap with each other, making the understanding of each effect important for accurate data matching. The following sections show the effects of varying each of the parameters on the short cavity. In each figure the run labelled Vshort2 is the current best match of the SPS data. This sort of analysis was also carried out for the long cavity.

Loop Delay

The loop delay describes the delay of the feedback system every turn. Varying the loop delay is useful in matching the initial response of the cavity to the beam.

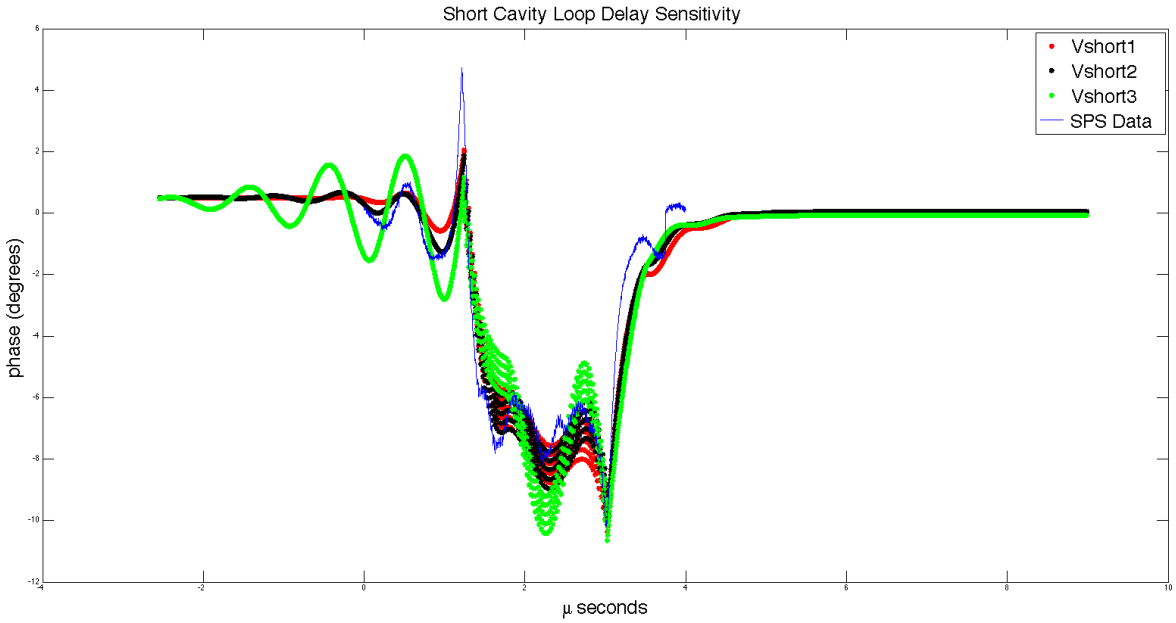


Figure 14: Increasing the loop delay increases the amplitude of the initial perturbations of the response.

High and Low Pass Delay

There is also a delay that acts between the high and low pass gain in the RF feedback system. Like the loop delay, it is best used for matching the initial spikes and end shape of the response.

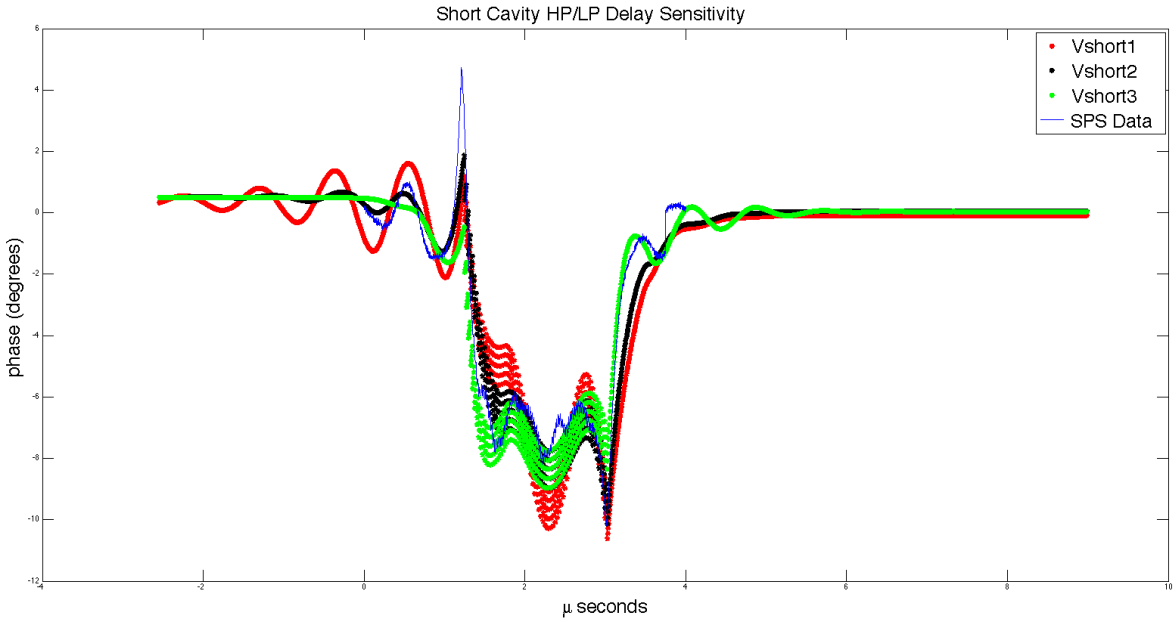


Figure 15: Increasing the delay decreases the amplitude of the initial perturbation, while increasing the disturbances at the end.

Loop Gain

The loop gain is the overall gain of the system. The loop gain is useful in controlling the peak-to-peak of the response, or the magnitude of the disturbance caused by the beam.

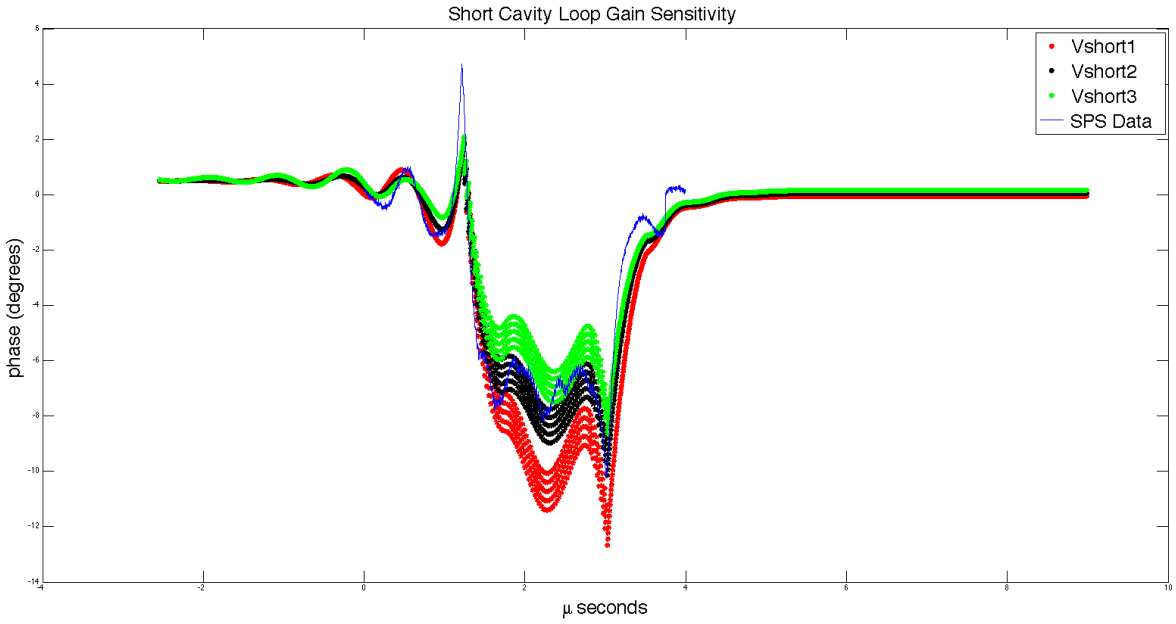


Figure 16: Increasing the loop gain brings the response up closer to the initial voltage phase. This parameter is used to match the position of the beam disturbance on the plot.

High Pass/ Low Pass Gain

The high pass and low pass gain have the biggest effect on the shape of the response to the beam. The most important factor in matching the shape of the disturbance is finding the correct ratio between these two gains. The low pass gain also comes with a similar effect to the loop gain in moving the vertical position of the disturbance on the response. Once a proper ratio is found between high and low pass, any displacement caused can be fixed by readjusting the loop gain.

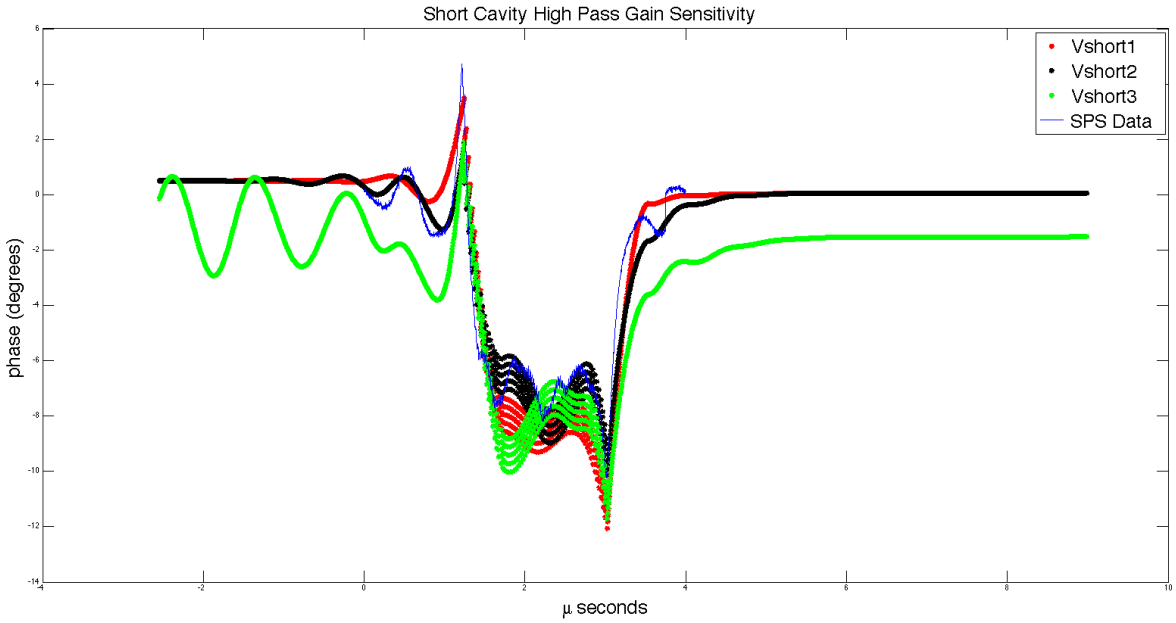


Figure 17: The high pass gain changes the shape of the response to the beam. Too much variation can begin to cause instability as seen in Vshort3.

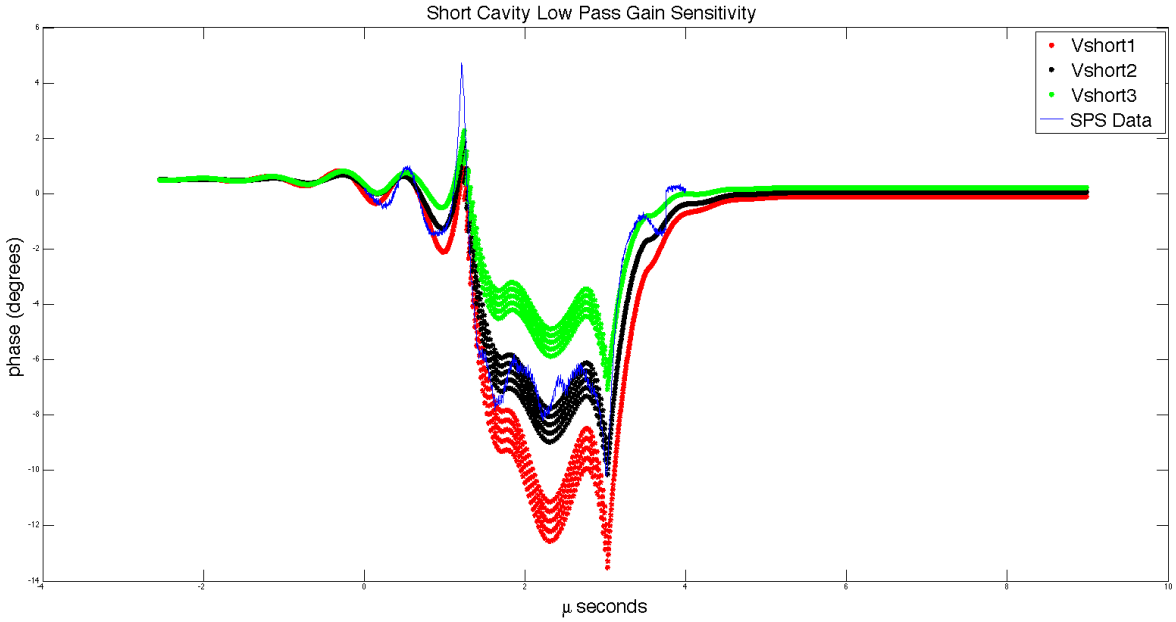


Figure 18: The low pass gain changes the shape of the response to the beam. Lower gain will create a steeper, more pronounced disturbance. Its best to analyze this in conjunction with the high pass gain to find the correct ratio.

Conclusion

The Super-Proton Synchrotron was successful modeled in both the frequency and time domains. Using the frequency domain model, significant gain and phase margins were estimated for the upgraded system.

A time domain model was also created with Matlab and Simulink and has been validated with SPS data. Once the data is fully replicated with the simulation, these two models can be used in conjunction as a tool for optimizing the SPS facility, providing useful information to the system designers, and estimating the limitations of the upgraded system before it is even operational.

The time domain model was also used to check beam stability as a function of various operational parameters. The initial analysis presented here, seems to indicate that better beam performance can be attained with a different set of parameters, in particular with adjustments to the low pass gain.

References

Baudrenghien, Ph. Reducing the Impedance of the Travelling Wave Cavities: Feed-forward and One Turn Feed-back (n.d.): n. pag. Web. 23 May 2016.

Cian O'Luanaigh, Cian. "The Super Proton Synchrotron." Http://home.cern/about/accelerators/super-proton-synchrotron. CERN, 23 Jan. 2012. Web. 24 May 2016.

Appendix

Initialization file:

initSPS200Long.m

```
clear all; close all;

%% Adjustable parameters
Vo = 2.2e6;
V_long = 3.1016e6;

loopGain = 3.0e-4;
loopGainLong = .55e-4;
hpGain = 40;
lpGain = .8;
hpGainLong = 350;
lpGainLong = 2;

frf = 200.222e6;
loopPhase = 40;
loopPhaseLong = 0;
% extraDelayTaps = -30;
turnBeamOn = 1e5; % Turn beam on after system stabilizes
steps = 3e5;

%% SPS parameters
h = 4620;
frev = frf/h;
fs = frf;
turnDelay = h*fs/frf;

%% Simulation parameters
Ts=1/frf;  % 200 MHz clock
stopTime = steps*Ts;

%% LIMIT SAVED DATA POINTS (scope)
ldp=100*h; % last 100 turns

%% 200 MHz CAVITY

% Short Parameters
L = 43*.374;
R2 = 27.1e3;
Zo = 50;

scaling = L*sqrt(Zo*R2/2);

cavNum = scaling*ones(1,124)/124/4;
cavDen = zeros(1,124); cavDen(1)=1;
cavInitial = [zeros(1,123); zeros(1,123)]';
% cavInitial = [750e3*ones(1,123); zeros(1,123)]';

% Long parameters
L_long = 54*.374;

scalingLong = L_long*sqrt(Zo*R2/2);

cavNumLong = scalingLong*ones(1,156)/156/4;
cavDenLong = zeros(1,156); cavDenLong(1)=1;
cavInitialLong = [242*ones(1,155); zeros(1,155)]';


%% IOT

w_bw = 2*pi*1.5e6;
att_bw = 3;
att_cut = 15;

% Short -- Siemens
w_cut = 2*pi*2.5e6;
[n,wb] = buttord(w_bw, w_cut, att_bw, att_cut, 's');
[B,A] = butter(n,wb,'s');

Amp = tf(B, A);
iotDis = c2d(Amp, Ts);
iotDisNum = iotDis.num{1};
iotDisDen = iotDis.den{1};
iotInitial = [232*ones(1,2); 0*ones(1,2)]';

% Long -- Philipps
```

```
w_cut = 2*pi*4e6;
[n,wb] = buttord(w_bw, w_cut, att_bw, att_cut, 's');
[B,A] = butter(n,wb,'s');

Amp = tf(B, A);
iotDis = c2d(Amp, Ts);
iotDisNumLong = iotDis.num{1};
iotDisDenLong = iotDis.den{1};
iotInitialLong = [232*ones(1,2); 0*ones(1,2)]';


%% Feedback Common
Rdev=5;          %Decimation rate in order to reduce Fifo size
Fdec=fs/Rdev;    %Decimation frequency (filter rate)
Ntap=Fdec/frev;
Rsim = Rdev;

% Hcheb
HchebNum = [0.000192357 -0.0110391 -0.0125555 -0.0161118 -0.016202 ...
    -0.0104059 -0.00399355 0.00896196 0.0235302 0.0399136 0.0587599 ...
    0.0748135 0.0907296 0.102548 0.109541 0.113132 0.109541 ...
    0.102548 0.0907276 0.0748135 0.0587599 0.0399136 0.0235302 ...
    0.00896196 -0.00399355 -0.0104059 -0.016202 -0.0161118 ...
    -0.0125555 -0.0110391 0.000192357];

% Comb
aopt=15/16;
G=1;
combFrevDen = [1 zeros(1, Ntap-1) -aopt];
combFrevNum = G*(1-aopt);
combFrevIC = [3.7e6*ones(1,Ntap); zeros(1,Ntap)]';

% LP delay
lpDelay = 43-2;
lpDelayLong = 43-5;
lpDelayIC = [2.3e6*ones(1, lpDelay); zeros(1, lpDelay)];
lpDelayICLong = [2.3e6*ones(1, lpDelayLong); zeros(1, lpDelayLong)];


%% Feedback (short)
% Low-pass
M = 24;
lpNum = [1 zeros(1, M) -1];
% lpNum = [zeros(1,16) 1 zeros(1, M) -1];
lpDen = (M+1)*[1 -1 zeros(1, length(lpNum)-2)];
lpIC = [2.3e5*ones(1, length(lpNum)-1); zeros(1, length(lpNum)-1)]';

% High-pass
b = 1/8;
hpNum = -(1+b)/4*[1 -1 zeros(1, M-1) -1 +1];
hpDen = [1 zeros(1, M) -b zeros(1, length(hpNum)-2)];
hpIC = [zeros(1, length(hpDen)-1); zeros(1, length(hpDen)-1)]';

% Delay
fbDelay = Ntap - 1 - (length(lpNum)+lpDelay+1)/2-43-5;
fbDelayIC = [2.3e6*ones(1, fbDelay); zeros(1, fbDelay)];

%% Feedback (long)
% Low-pass
M = 30;
lpNumLong = [1 zeros(1, M) -1];
lpDenLong = (M+1)*[1 -1 zeros(1, length(lpNumLong)-2)];
lpICLong = [2.3e5*ones(1, length(lpNumLong)-1); zeros(1, length(lpNumLong)-1)]';

% High-pass
b = 1/8;
hpNumLong = -(1+b)/4*[1 -1 zeros(1, M-1) -1 +1];
hpDenLong = [1 zeros(1, M) -b zeros(1, length(hpNumLong)-2)];
hpICLong = [zeros(1, length(hpDenLong)-1); zeros(1, length(hpDenLong)-1)]';

% Delay
fbDelayLong = Ntap - 1 - (length(lpNumLong)+lpDelay+1)/2-43-5;
fbDelayICLong = [2.3e6*ones(1, fbDelayLong); zeros(1, fbDelayLong)];


%% BEAM
batch = [ones(1,72); zeros(4,72)]; batch = batch(:);
beamPattern = [batch; zeros(h-length(batch),1)];
% beamPattern = [batch; zeros(45,1); batch; zeros(45,1); batch; zeros(45,1); ...
%     batch; zeros(45,1); zeros(3e3,1)]; % at 200 MHz (5 ns spacings)

Np = 1.3*1e11;     % Protons per bunch
fb = 1;
q = 1.6e-19;     % Proton charge
Tspacing = 25e-9;
```

```
161
162 -    numberBunches = sum(beamPattern);
163
164      % Instantaneous and average current
165 -    Idc = numberBunches*Np*frev*q;     % DC current
166 -    Irf = 2*fb*Idc;          % RF component of dc current
167 -    Ipeak = 2*fb*Np*q/Tspacing;    % Rf component of peak current
168
169      % phiInitial = zeros(1, h);%
170
171      % Beam Phase
172 -    Ib.signals.values = Ipeak*beamPattern;
173 -    Ib.time = [];
174
175      % Short cavity parameters
176 -    vg = 0.0946;
177 -    c = 300e6;  %speed of light
178 -    K = (L/vg/c)*(1-vg);
179
180      % beam response given by Zt1+Z3
181 -    num1 = 1;
182 -    den1 = [1 zeros(1,102)];
183 -    Z2 = tf(num1,den1,1/fs);
184
185 -    num = 1; den = [K^2 0 0];
186 -    Z1c = tf(num,den);
187 -    Z1 = c2d(Z1c,1/fs,'impulse');
188 -    Zbt2 = -(L*L*R2/4)*(Z1*Z2);
189 -    Z1Num = Zbt2.num{1};
190 -    Z1Den = Zbt2.den{1};
191
192 -    Z3_t = tf([-K 1],[K^2 0 0]);
193 -    Z3_t2 = c2d(Z3_t,1/fs,'impulse');
194 -    Z3_t3 = (L*L*R2/4)*Z3_t2;
195 -    Z3Num = Z3_t3.num{1};
196 -    Z3Den = Z3_t3.den{1};
197
198      % Long cavity parameters
199 -    K = (L_long/vg/c)*(1-vg);
200
201 -    n = 12;
202 -    num1 = 1;
203 -    den1 = [1 zeros(1,128)];
204 -    Z2 = tf(num1,den1,1/fs);
205
206 -    num = 1; den = [K^2 0 0];
207 -    Z1c = tf(num,den);
208 -    Z1 = c2d(Z1c,1/fs,'impulse');
209 -    Zbt2 = -(L*L*R2/4)*(Z1*Z2);
210 -    Z1NumLong = Zbt2.num{1};
211 -    Z1DenLong = Zbt2.den{1};
212
213 -    Z3_t = tf([-K 1],[K^2 0 0]);
214 -    Z3_t2 = c2d(Z3_t,1/fs,'impulse');
215 -    Z3_t3 = (L*L*R2/4)*Z3_t2;
216 -    Z3NumLong = Z3_t3.num{1};
217 -    Z3DenLong = Z3_t3.den{1};
218
219
```