Table of Contents

Part 2: Creating the frequency response

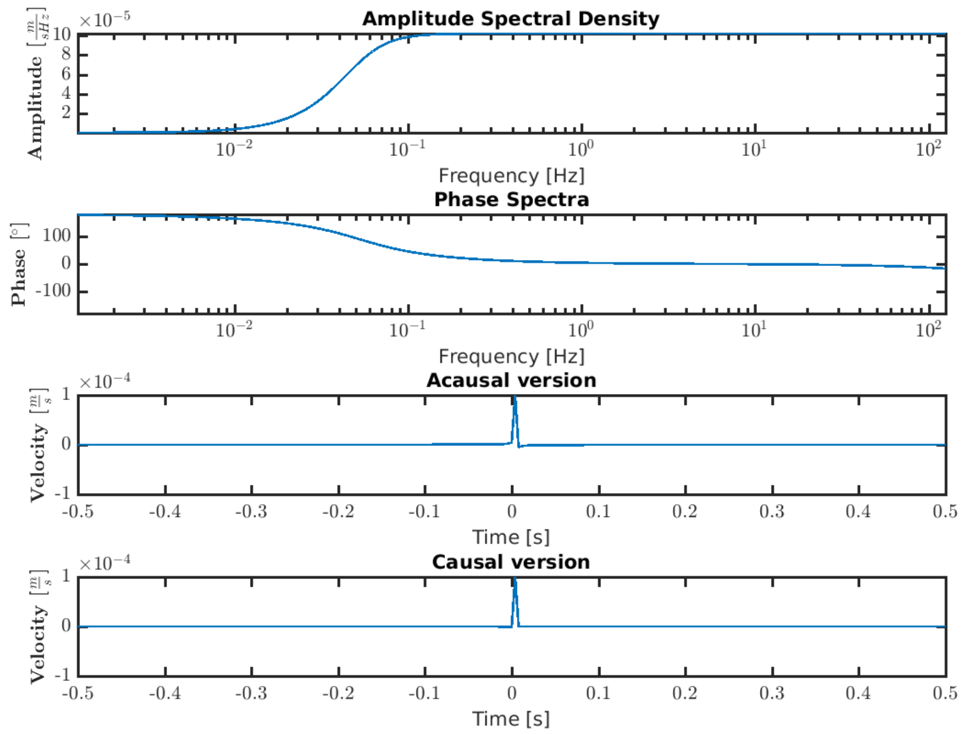
```
addpath '/home/rflee/Documents/MATLAB/mseed/'
f = '../../Data/XX.BSM7.HHZ_MC-PH1_0426_20170112_180000.miniseed';
Meta = rdmseed(f);

sf= Meta.SampleRate; %sampling frequeny in Hz
ny= .5*sf; % nyquist frequency
n = 10e4+1; % number of points for frequency vector
dt = 1/sf;

% create frequency array
frequencies= linspace(0,ny,n);
w = frequencies*2*pi;

A0=4.344928e17; % conversion from counts to m/s for this instrument

zeros = [ % the zeros in 1/sec
    0.000000E+00;
    0.000000E+00;
    -3.920000E+02;
    -1.960000E+03;
    -1.490000E+03 + 1i*1.740000E+03;
    -1.490000E+03 + 1i*-1.740000E+03
    ];

poles = [ % the poles in 1/sec
    -3.691000E-02 + 1i* 3.702000E-02;
    -3.691000E-02 + 1i*-3.702000E-02;
    -3.430000E+02 + 1i* 0.000000E+00;
    -3.700000E+02 + 1i* 4.670000E+02;
    -3.700000E+02 + 1i*-4.670000E+02;
    -8.360000E+02 + 1i* 1.522000E+03;
    -8.360000E+02 + 1i*-1.522000E+03;
    -4.900000E+03 + 1i* 4.700000E+03;
    -4.900000E+03 + 1i*-4.700000E+03;
    -6.900000E+03 + 1i* 0.000000E+00;
    -1.500000E+04 + 1i* 0.000000E+00
```

```
    ];

% get coefficients for polynomials
num = poly(zeros*2*pi); %convert poles and zeros to rad/s for function
denom= poly(poles*2*pi);

%get the frequency response
fResponse=A0*freqs(num,denom,w);

% convert frequency response to polar (magnitude and phase)
mag =abs(fResponse);
phase = angle(fResponse);
phase = rad2deg(phase);

% get time series amplitudes
Vacausal = ifft(fResponse);
Vacausal = ifftshift(Vacausal);

% create times
t1 = 0:dt:dt*((n-1)/2); % positive times
t2= -dt*((n-1)/2):dt:-dt; % negative times
t = [t2, t1];

% get times
df = frequencies(2)-frequencies(1);

% Apply Haney method to enforce causality
Vcausal = real(fResponse) -1i*(hilbert(real(fResponse)));
Vcausal = real(ifftshift(ifft(Vcausal)));
```

Plots

```
fsize=16;

scrsz = get(0,'ScreenSize');
HH=figure('Position',[scrsz(3) scrsz(4) scrsz(3) scrsz(4)]);
subplot(4,1,1)
semilogx(frequencies,mag,'LineWidth',2);
axis tight
xlabel('Frequency [Hz]')
ylabel('{\bf Amplitude [$\frac{m}{s Hz}$]}','Interpreter','latex')
title('Amplitude Spectral Density')


subplot(4,1,2)
semilogx(frequencies,phase,'Linewidth',2)
axis tight
ylim([-180 180])
xlabel('Frequency [Hz]')
ylabel('{\bf Phase [$^{\circ}$]}','Interpreter','latex')
title('Phase Spectra')
```

```
subplot(4,1,3)
plot(t,real(Vacausal),'Linewidth',2)
ylim([-1e-4 1e-4])
xlim([-.5 .5])
ylabel('{\bf Velocity [$\frac{m}{s}$]}','Interpreter','latex')
xlabel('Time [s]')
title('Acausal version')


subplot(4,1,4)
plot(t,Vcausal,'Linewidth',2)
ylim([-1e-4 1e-4])
xlim([-.5 .5])
ylabel('{\bf Velocity [$\frac{m}{s}$]}','Interpreter','latex')
xlabel('Time [s]')
title('Causal version')

figproperties
```



Discussion

For this instrument, the response starts to fall off below about .1 Hz The phase spectra shows the response is about zero from nyquist down to .1 hz. Below .1 Hz the phase slowly increases until about pi. I interpret this to mean there can be change in the phase up to pi at low frequencies so that by .01 Hz the polarity is reversed. The time domain of the frequency response shows a near delta function, so the response should reflect the motion since the time series does not show any resonance.

Part 3: Instrument Deconvolution

Import Data

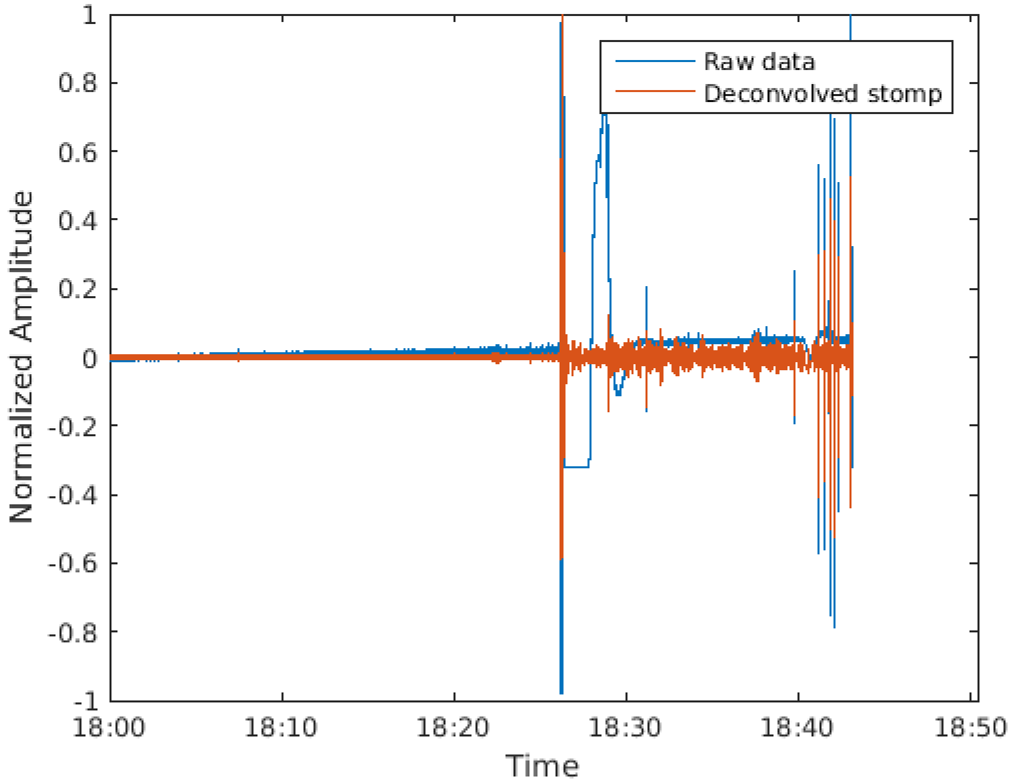
```
t = cat(1,Meta.t);
rawdata = cat(1,Meta.d);

% %plot initial data
% Xlim = [min(cat(1,Meta.t)),max(cat(1,Meta.t))];
% figure;
% plot(t,rawdata)
% set(gca,'XLim',Xlim)
% datetick('x','keeplimits')
```

Deconvolve the data

```
Sensitivity = 4.000000E+05;
% ordl Butterworth order at flo, the low cutoff (between 2 and 4)
ordl = 2;
% ordh Butterworth order at fhi, the high cutoff (between 3 and 7)
ordh = 3;
% digout: inverse gain (m/s/count)
digout = Sensitivity/A0;
% digoutf: frequency of normalization (Hz)
digoutf = 1;
% ovrsampl: over-sampling factor for digital filter accuracy (e.g., 5)
ovrsampl = 10;
% idelay: intrinsic delay in the acquisition system
idelay = 0;

flo= 10e-2;
fhi = 10e2;
badvals= -2e31;

decondata =
 rm_instrum_resp(rawdata,badvals,sf,poles,zeros,flo,fhi,ordl,ordh,digout,digoutf,o
```

plot all data

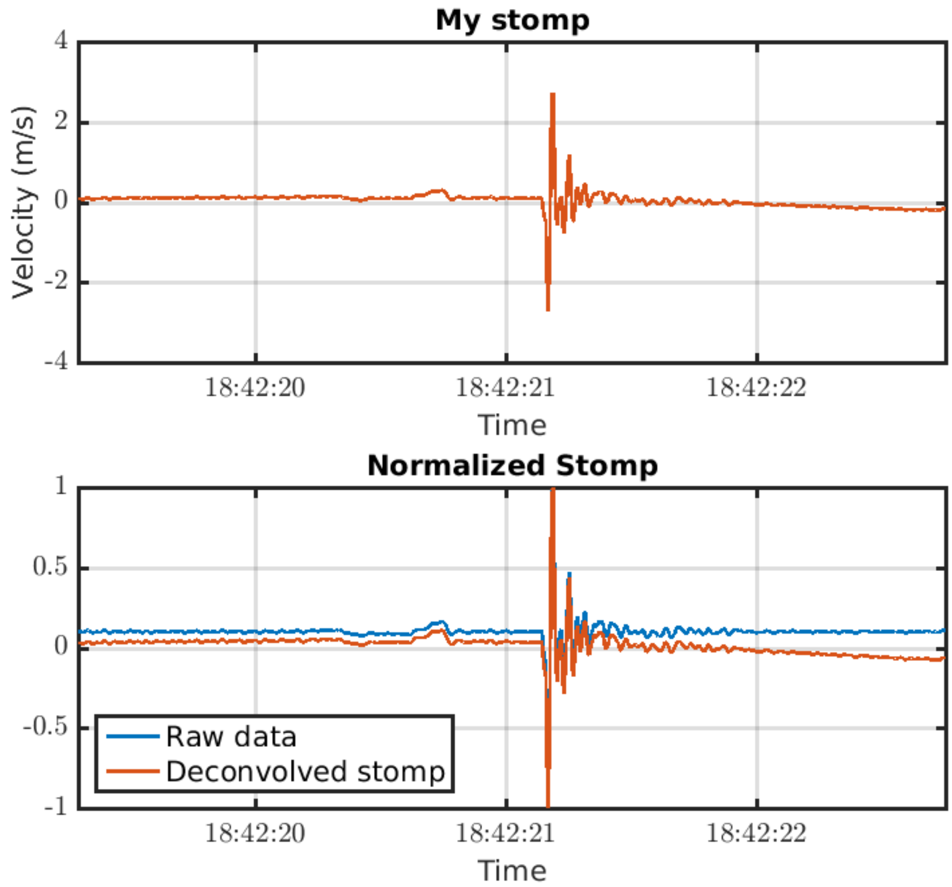
```
normraw = rawdata./max(abs(rawdata));
normdecon = decondata./max(abs(decondata));
figure;
plot(t,normraw)
hold on
plot(t,normdecon)
ylabel('Normalized Amplitude')
xlabel('Time')
datetick('x','keeplimits')
legend('Raw data','Deconvolved stomp')
```



Plot my stomp

```
mylim = [7.3670777939 7.3670777943]*1e5; % I zoomed into mystop and
 got xlim to get these numbers

% cut the data to my stomp

%find index numbers
[~,idxstart] = min(abs(t-mylim(1)));
[~,idxstop] = min(abs(t-mylim(2)));

% cut data and time vectors
mystompraw = rawdata(idxstart:idxstop);
mystompraw = mystompraw./max(abs(mystompraw));

mystompdecon = decondata(idxstart:idxstop);
mytim = t(idxstart:idxstop);
% plot my stomp-----------------------
figure;
subplot(2,1,1)
plot(mytim,mystompdecon);hold on;
plot(mytim,mystompdecon)
xlim(mylim)
datetick('x','keeplimits')
title('My stomp')
```

```
xlabel('Time')
ylabel('Velocity (m/s)')

figproperties;
grid on
%--------------------------------
mystompdecon = mystompdecon./(max(abs(mystompdecon)));




subplot(2,1,2)
plot(mytim,mystompraw)
hold on
plot(mytim,mystompdecon);
xlim(mylim)
datetick('x','keeplimits')
legend('Raw data','Deconvolved stomp','Location','SouthWest')
hold off
title('Normalized Stomp')
xlabel('Time')

figproperties;
grid on

set(gcf,'Units','Inches')
set(gcf,'Position',[1 1.5 8.7 7.3])
```



Discussion

Compared to the deconvolved data, the raw data shows the same shape as the deconvolved data. The main difference is that there seems to be a D.C. offset that is nearly corrected to zero for the deconvolved data. Where there is no event, however, the deconvolved data fluctuates around zero.

Published with MATLAB® R2015b