Instrument Responses

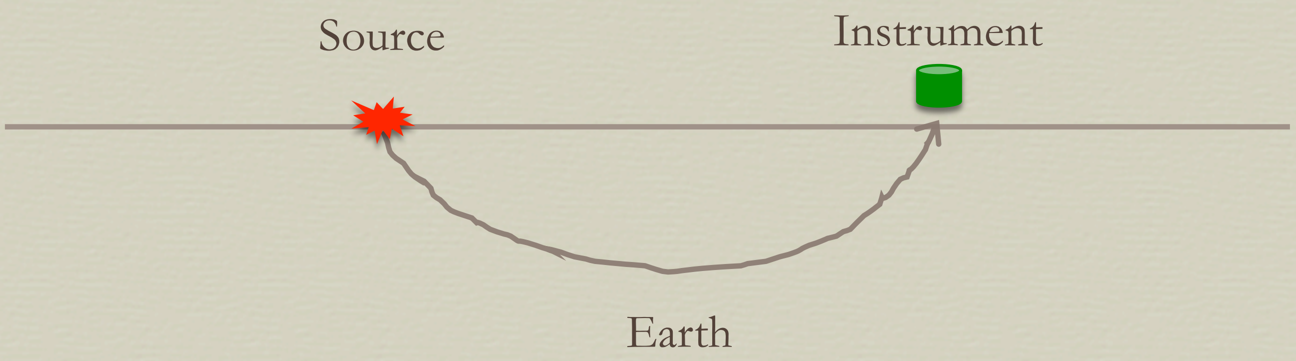
Mary Templeton
IRIS DMC
December 13, 2015

What is an Instrument Response?

A response describes how an instrument changes an input signal to produce an output signal.

When are Instrument Responses Important?

TimeSeries1(t) = Source(t) * Earth(t) * Instrument1(t)
TimeSeries2(t) = Source(t) * Earth(t) * Instrument2(t)



When are Instrument Responses Important?

TimeSeries1(t) = Source(t) * Earth(t) * Instrument1(t)
TimeSeries2(t) = Source(t) * Earth(t) * Instrument2(t)

When you're:

- Studying wave sources
- Studying earth structure or propagation effects
- Studying ground motion (size and units matter)
- Comparing or using records from diverse instrumentation
- Archiving data for others' use

When are Instrument Responses **Not** Important?

TimeSeries1(t) = Source(t) * Earth(t) * Instrument1(t)
TimeSeries2(t) = Source(t) * Earth(t) * Instrument2(t)

Sometimes when you're:

- Imaging
- Picking
- Using homogeneous instrumentation
- Unconcerned about size and units

Anatomy of an Instrument Response

- The sequence is the "response cascade"
- Each step within the cascade is a "stage"
- Cascade each stage in the order in which it was applied during recording

What We'll Do

- Sensors & Amplifiers
  - Where do amplitude and phase response (Bode) plots come from?
  - Where do poles and zeros come from?
  - How are amplitude & phase responses related to poles & zeros?
  - SEED sensor and amplifier responses
  - Other useful things to know about sensor and amplifier responses

What We'll Do

- Dataloggers and Filters
  - How dataloggers work
    - Analog to Digital Conversion
    - Oversampling, decimation and other filtering
  - Where do FIR coefficients come from?
  - How are amplitude and phase responses related to FIR coefficients?
  - SEED datalogger and filter responses
  - Other useful things to know about datalogger and filter responses

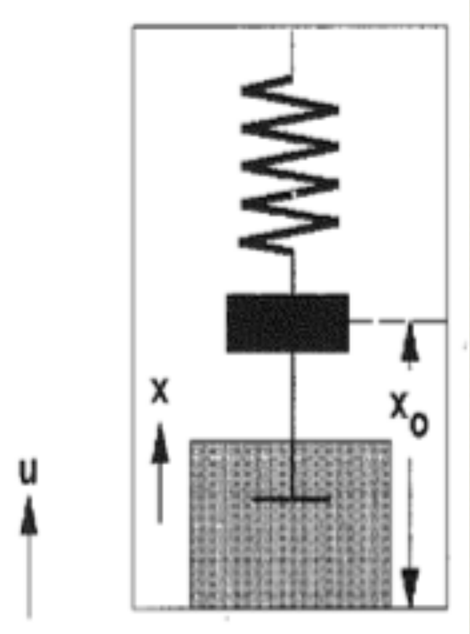
What We'll Do

- Response Tools and Notes
    - Nominal Response Library
    - Retrieving responses from the DMC
    - Writing responses (dataless SEED)
    - Removing instrument responses
    - Verifying responses

Sensors

- have continuous inputs and outputs (they're analog!)
- They usually change the units of the property being measured into Volts.
- Solving the sensor's equation for its output at all frequencies gives us its frequency response function (a polynomial) that describes the sensor's frequency-dependent amplitude and phase changes.
- The frequency response function is a special case of the more descriptive transfer function – a polynomial that can be defined by its roots (poles and zeros) if factored, or from its coefficients if expanded form.

Sensor Example: Passive Seismometer



From Scherbaum (1996)

Equation of Motion

$x'' + 2h\omega_0 x'(t) + \omega_0^2 x(t) = -u''(t)$

where

$x(t)$ = relative mass displacement
$u''(t)$ = ground acceleration (input signal)
ω_0= angular natural frequency
h = damping factor $(0 <= h <= 1)$

Sensor Example: Passive Seismometer

- From differential equations, we know to try a solution that describes harmonic oscillation where

  $x(t) = A_o e^{j\omega t}$
  $x'(t) = j\omega A_o e^{j\omega t}$
  $x''(t) = -\omega^2 A_o e^{j\omega t}$
  $u''(t) = -\omega^2 A_i e^{j\omega t}$
  ω is a constant angular frequency, for now

- and for constant ω
  - Real{x(t)} is a cosine wave with amplitude A_o
  - Imaginary{x(t)} is a sine wave with amplitude A_o

Linear Time-Invariant Systems

- But we'd like to solve for **all** frequencies. Fortunately, seismometers are linear time-invariant systems (LTI), meaning that for a function ϕ that converts input signal u(t) to output signal x(t)

$$x(t) = \phi[u(t)]$$

- superposition is valid

$$\Phi[u_1(t) + u_2(t)] = \phi[u_1(t)] + \phi[u_2(t)]$$

- and the order in which we scale doesn't matter

$$\phi[\mathbf{A}_1 u(t)] = A_1 \phi[u(t)]$$

- regardless of **when** we perform these operations

Frequency Response Function

- So we can use the Fourier Transform (the sum of solutions over all ω) to describe the behavior of a sensor over all ω. Making earlier substitutions and simplifying

$$-\omega^2 A_o + 2h\omega_0 j\omega A_o + \omega_0^2 A_o = \omega^2 A_i$$

- Solving for the ratio of output/input gives the Frequency Response Function

$$T(j\omega) = A_o / A_i$$

$$= \omega^2 / [\omega_0^2 - \omega^2 + j2h\omega_0\omega]$$

Frequency Response Function

- Where the Real part of the Frequency Response Function describes Amplitude as a function of frequency

$$|T(j\omega)| = |A_o/A_i| = |A_o|/|A_i|$$

$$= \omega^2 / \{sqrt[\omega_0{}^2 - \omega^2]^2 + 4h^2w_0{}^2w^2\}$$

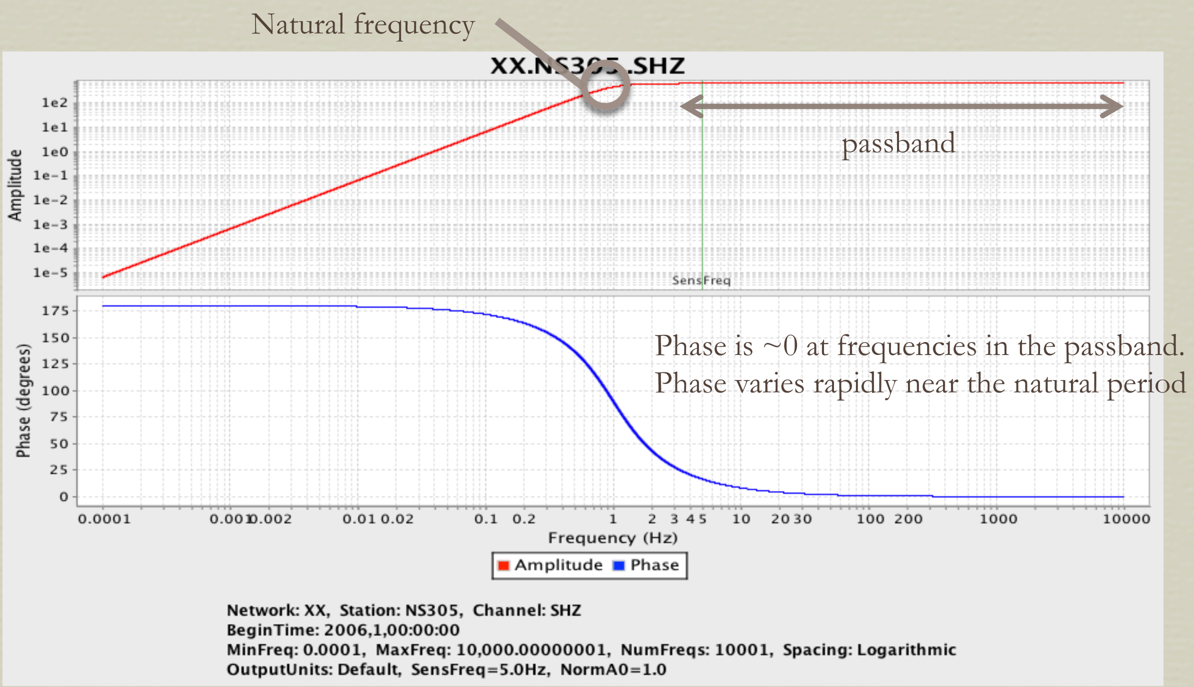
- And the phase angle is

$$\phi(\omega) = \arctan(\text{Imaginary}/\text{Real})$$

$$= \arctan(-2h\omega_0\omega / \omega_0{}^2 - \omega^2)$$

Frequency Response Function

- The plots of amplitude and phase as a function of frequency are often called Bode plots



Non-Linear Systems

- When the output of a system depends strongly on the input amplitude, superposition and scaling do not hold
- Examples of nonlinear behavior include
  - Seismometers with off-center masses
  - Analog to digital convertors with a faulty resistor
  - Others?

Transfer Function

$$x'' + 2h\omega_0 x'(t) + \omega_0^2 x(t) = -u''(t)$$

- Another way to solve the seismometer's equation of motion is to solve its Laplace transform. Recall that

$$x(t) \Leftrightarrow X(s)$$
$$x'(t) \Leftrightarrow sX(s)$$
$$x''(t) \Leftrightarrow s^2X(s)$$
$$u''(t) \Leftrightarrow s^2U(s)$$
$$s = \sigma + j\omega$$

- Substituting

$$s^2X(s) + 2h\omega_0 sX(s) + \omega_0^2 X(s) = -s^2U(s)$$

Transfer Function

$$s^2X(s) + 2h\omega_0 sX(s) + \omega_0^2X(s) = -s^2U(s)$$

- Solving for the ratio of output/input gives the Transfer Function

$$T(s) = X(s)/U(s)$$

$$= -s^2/[s^2 + 2h\omega_0 s + \omega_0^2]$$

- Values of s that make the numerator go to zero are "zeros". Where are they in this example?
- Values that make the denominator go to zero are "poles". Factoring the denominator gives the value of its two poles

Transfer Function

$$T(s) = -s^2 / [s^2 + 2h\omega_0 s + \omega_0^2]$$
$$= -s^2 / [(s - p_1)(s - p_2)]$$

- Factoring the denominator using the quadratic equation, gives two poles

$$p_1 = -[h - \text{sqrt}(h^2 - 1)]\,\omega_0$$
$$p_2 = -[h + \text{sqrt}(h^2 - 1)]\,\omega_0$$

If the sensor is underdamped ($h<1$), the term under the sqrt will be imaginary.

- You can recreate the transfer function knowing just its poles and zeros.
- You can also recreate the transfer function if you store the coefficients of the numerator (0, 0, -1) and denominator (ω_0^2, $2h\omega_0$, 1)

Relationship between the Frequency Response and Transfer Functions

$$T(s) = -s^2 / [s^2 + 2h\omega_0 s + \omega_0^2]$$
$$T(j\omega) = \omega^2 / [-\omega^2 + j2h\omega_0\omega + \omega_0^2]$$

- Notice how similar the Transfer and Frequency Response Functions are.
- Recall that complex $s = \sigma + j\omega$.
  - The Frequency Response Function is a special case of the Transfer Function where $\sigma = 0$.
  - In other words, the Frequency Response Function is the imaginary part of the Transfer Function.

Relationship between the Frequency Response and Transfer Functions

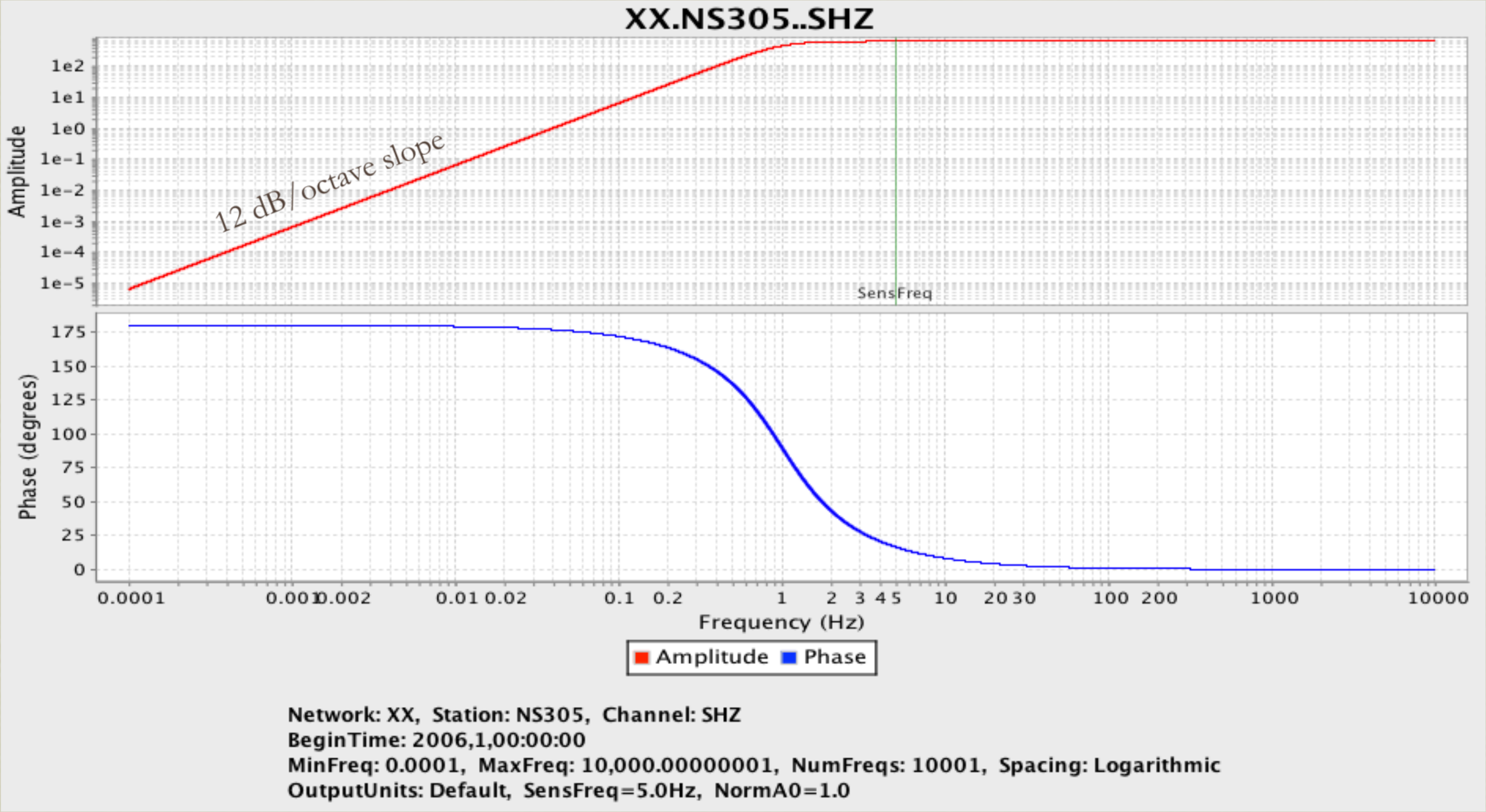
- The corner frequency of a pole or zero can be found by taking its modulus (sqrt[$Re^2 + Im^2$]). Remember that you may need to convert from radians into Hz!

- Each zero introduces a positive slope of the amplitude response on a log-log plot by 6 dB/octave (or 20 dB/decade) at frequencies higher than its corner frequency

- Each pole introduces a negative slope of the amplitude response on a log-log plot by 6 dB/octave (or 20 dB/decade) at frequencies higher than its corner frequency

- A pole and zero at the same corner frequency will cancel each other.

Relationship between the Frequency Response and Transfer Functions



A Note About the Time Domain

- Superposition and Scaling allow us to multiply the Amplitude spectra of successive LTI response stages in the frequency domain. The time-domain equivalent of this is convolution.
- There is also a time-domain representation of the response called the Impulse Response Function. It is the output signal that results from a dirac delta input signal.
- The Fourier Transform of the Impulse Response Function is the Frequency Response Function.
- The Laplace Transform of the Impulse Response Function is the Transfer Function.
- Manufacturers often “fit” poles and zeros to the Fourier Transform of the impulse response rather than deriving them.

SEED Sensor Stage

```
B050F03     Station:       NS305
B050F16     Network:       XX
B052F03     Location:
B052F04     Channel:       SHZ
B052F22     Start date:    2006,001,00:00:00
B052F23     End date:      2599,365,23:59:59
#
#                  +-----------------------------------+
#                  |     Response (Poles and Zeros)     |
#                  |        XX  NS305  SHZ             |
#                  |     01/01/2006 to 12/31/2599       |
#                  +-----------------------------------+
#
B053F03     Transfer function type:             A
B053F04     Stage sequence number:              1
B053F05     Response in units lookup:           M/S - Velocity in Meters/Second
B053F06     Response out units lookup:          V - Volts
B053F07     AO normalization factor:            +1.00000E00
B053F08     Normalization frequency:            +5.00000E00
B053F09     Number of zeroes:                   2
B053F14     Number of poles:                    2
#              Complex zeroes:
#                i  real          imag          real_error    imag_error
B053F10-13       0  +0.00000E00  +0.00000E00  +0.00000E00  +0.00000E00
B053F10-13       1  +0.00000E00  +0.00000E00  +0.00000E00  +0.00000E00
#              Complex poles:
#                i  real          imag          real_error    imag_error
B053F15-18       0  -4.44300E00  +4.44300E00  +0.00000E00  +0.00000E00
B053F15-18       1  -4.44300E00  -4.44300E00  +0.00000E00  +0.00000E00
##
#                  +-----------------------------------+
#                  |     Channel Sensitivity/Gain       |
#                  |        XX  NS305  SHZ             |
#                  |     01/01/2006 to 12/31/2599       |
#                  +-----------------------------------+
#
B058F03     Stage sequence number:              1
B058F04     Sensitivity:                        +6.29000E02
B058F05     Frequency of sensitivity:           +5.00000E00
B058F06     Number of calibrations:             0
```

Poles and zeros can be listed in units of Radians (A – most common) or Hz (B). (1 radian = 2π Hz)

Input units reflect what sensor measures (SI units)

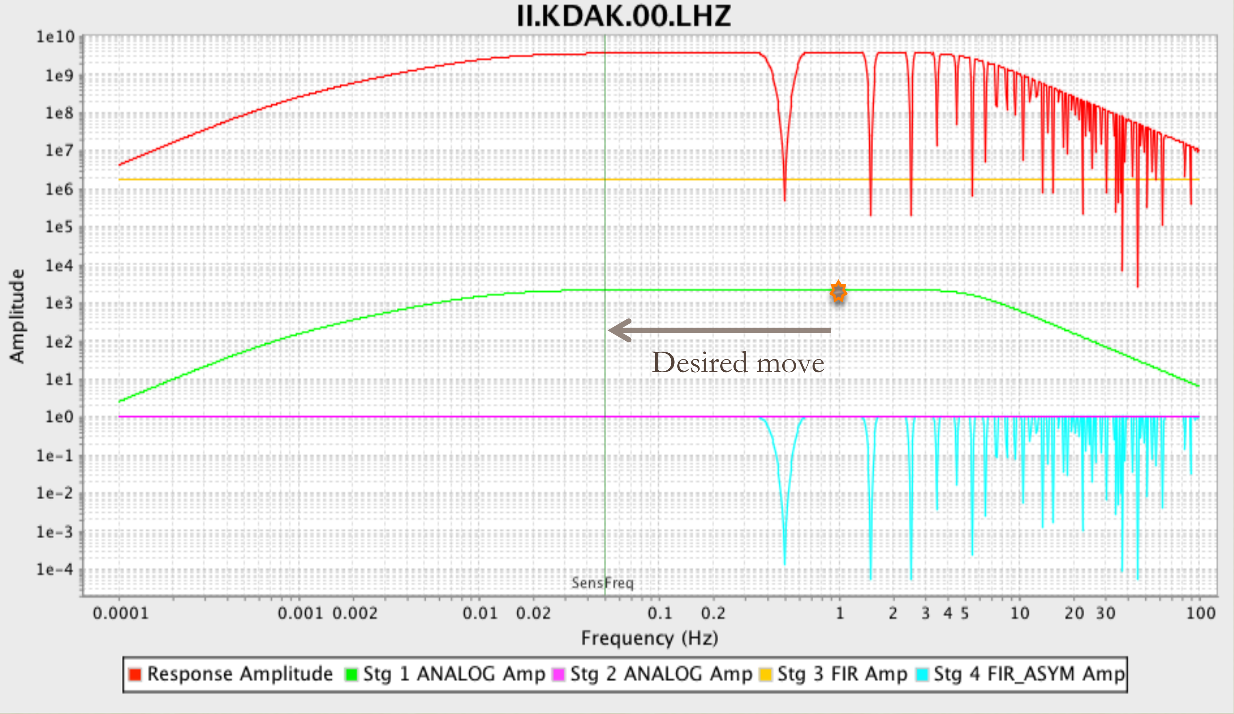
Pole-zero curve is **normalized** (=1 in passband). A_0 must have same units as poles and zeros

The stage 1 Gain blockette lists the sensor sensitivity.

Normalization

- You normalize the pole/zero curve so that you can multiply by the sensor gain and the resulting curve will equal the sensor gain in the passband.
- A_0 is the factor you multiply the pole/zero curve by at the normalization frequency to get a value of 1.
- If your sample rate is low enough that sensor normalization frequency is no longer in the passband, you may need to normalize at a lower frequency.
- If the passband is not exactly flat and you need to move your normalization frequency, you may need to specify a sensor gain that differs a little from that reported by the manufacturer.

Normalization



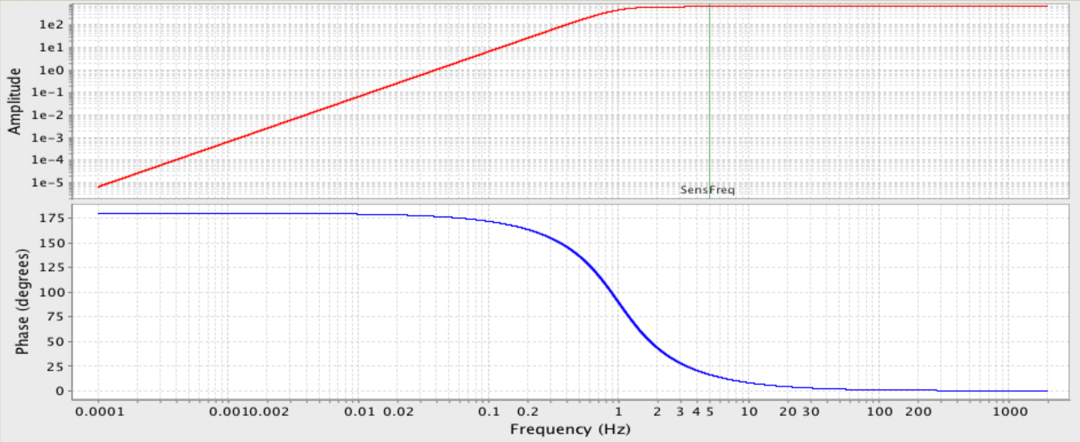
Suppose the sensor gain is known at 1 Hz, but your 1 sps LHZ channel has no amplitude there?

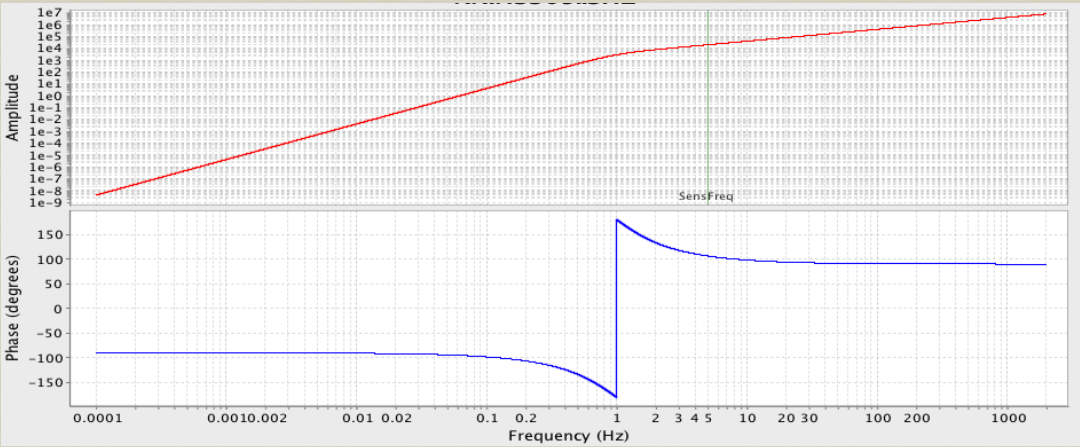
1. Find a lower frequency in the passband.
2. Find the sensor gain value at that frequency (plot only the sensor stage).
3. Find and enter A0 for the new frequency.
4. Change the sensor gain to the value in step 2.

Displacement, Velocity and Acceleration

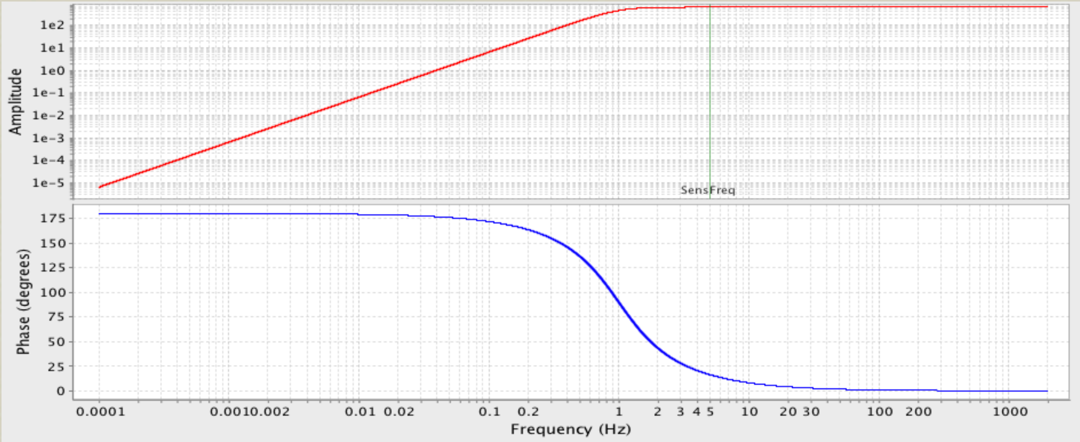
- For SEED, it's preferred that the sensor's response have a passband that is flat to the property being measured. A velocity transducer should have a "velocity response" – its passband is flat to velocity with input units of Meters/second.
- It's also possible to create an "acceleration response" for a velocity transducer. Since $T'(s) = sT(s)$, taking the derivative of a velocity response adds a zero at 0.
- Creating a "displacement response" from a velocity response is equivalent to removing a zero since integrating $T(s)$ is equivalent to dividing by s.

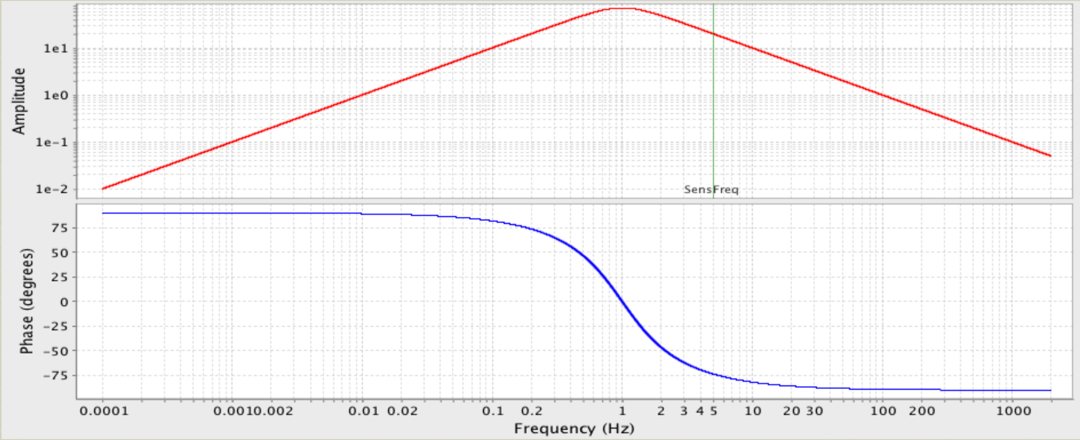
How Do These Differ?





How Do These Differ?





Amplifiers

- Many dataloggers have analog preamplifiers that boost signal prior to digitization. Some stations use separate amplifiers.
- Amplifiers change only the amplitude of the signal independently (we assume) of frequency.
- In SEED, it is recommended that the amplifier have its own stage and include only a Gain description.

```
#               +                  +-----------------------------------------+
#               +                  |        Channel Gain, NQ008 ch LHZ       |
#               +                  +-----------------------------------------+
#
B058F03     Stage sequence number:                  2
B058F04     Gain:                                   3.000000E+01
B058F05     Frequency of gain:                      5.000000E-02 HZ
B058F06     Number of calibrations:                 0
```

More about Sensors

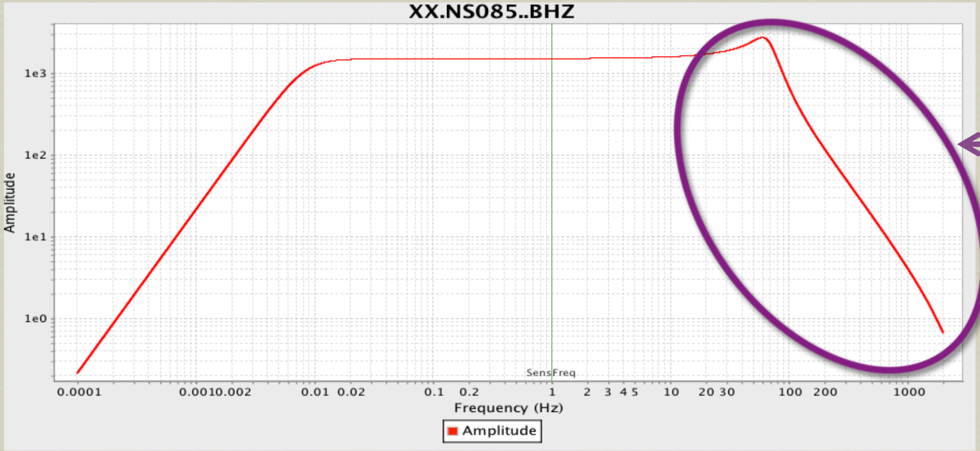
Passive velocity seismometers

- have a simple mass-spring-damping system that requires no electricity for operation.
- have 2 zeros at 0 and 2 poles at the natural period related to the mass-spring system.
- sensitivity, poles, zeros and damping depend on their resistors, mass, period and mechanical damping as described here: http://ds.iris.edu/NRL/sensors/sercel/passive_responses.html
- If the impedance contrast between sensor and amplifier is less than 2 orders of magnitude, the amplifier will change the sensor damping and, therefore, its poles and zeros.

More about Sensors

Active velocity seismometers

- use feedback electronics to modify the natural period of the mass-spring system and to control the damping, therefore they require electricity for operation
- Have 2 zeros at 0, 2 poles at the natural frequency, plus additional poles and/or zeros at higher frequencies that describe the feedback electronics



Feedback electronics

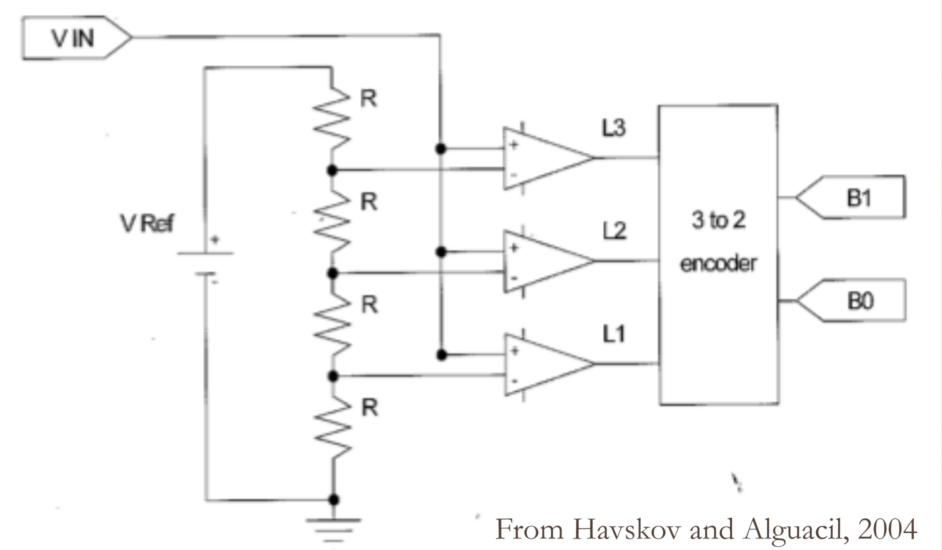
Dataloggers

- may include an analog preamplifier that changes the gain of the signal
- sample the input voltage, changing its gain and units and creating an initial sample rate
- decimate the sampled voltage using digital Finite Impulse Response (FIR) filters, which changes its sample rate and occasionally changes its gain.
- may include additional filters such as
  - an analog anti-alias filter,
  - Infinite Impulse Response (IIR) filters

Dataloggers

Analog to Digital Conversion

- A simple analog to digital converter (ADC) samples by comparing an input voltage at regular time intervals to reference voltages to determine its size



From Havskov and Alguacil, 2004

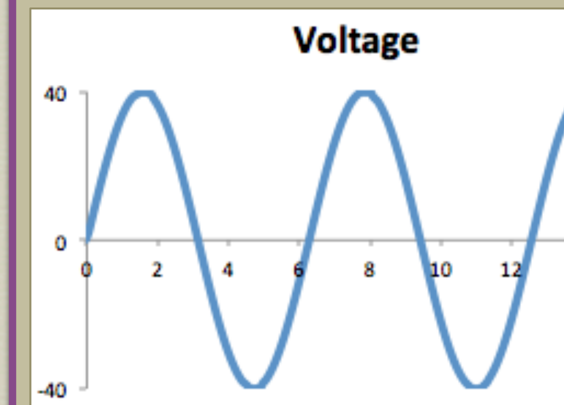
- The states for comparators L1, L2 and L3 are initially (0,0,0).
- Each comparator whose voltage is exceeded by Vin gets set to 1.
- A voltage with comparator states (1,1,0) has 2 counts.

Dataloggers

Analog to Digital Conversion

- The input sample rate is determined by the ADC
- The ADC scale factor in Counts/Volt depends on the ADC size (the number of comparisons it can make = the number of counts it can recognize) and the the voltage range allowed. So a true 24-bit ADC sampling a voltage range of 40 Vpp has scale factor

ADC scale factor = 2^{24} Counts / 40 Volts
= 4.194×10^{5} Counts/Volt
= 1 / Least Significant Bit (LSB)



Voltage Range Equivalents
- 40 Volts peak-to-peak (Vpp)
- 20 Volts peak (Vp)
- +/-20 Volts Full Scale Voltage

SEED ADC Stage

Analog to Digital Conversion

```
#
#             +                  +---------------------------------------------+
#             +                  |   Response (Coefficients), NR201 ch LHZ    |
#             +                  +---------------------------------------------+
#
B054F03     Transfer function type:                  D
B054F04     Stage sequence number:                   3
B054F05     Response in units lookup:                V - Volts
B054F06     Response out units lookup:               COUNTS - Digital Counts
B054F07     Number of numerators:                    1
B054F10     Number of denominators:                  0
#               Numerator coefficients:
#                 i, coefficient,   error
B054F08-09     0  1.000000e+00  0.000000E+00
#
#             +                         +-------------------------------+
#             +                         |   Decimation, NR201 ch LHZ    |
#             +                         +-------------------------------+
#
B057F03     Stage sequence number:                   3
B057F04     Input sample rate:                       1.024000e+05
B057F05     Decimation factor:                       1
B057F06     Decimation offset:                       0
B057F07     Estimated delay (seconds):               0.000000E+00
B057F08     Correction applied (seconds):            0.000000E+00
#
#             +                     +---------------------------------------+
#             +                     |     Channel Gain, NR201 ch LHZ        |
#             +                     +---------------------------------------+
#
B058F03     Stage sequence number:                   3
B058F04     Gain:                                    3.669720e+05
B058F05     Frequency of gain:                       5.000000e-02 HZ
B058F06     Number of calibrations:                  0
```

Digital stage

Units change

One coefficient (unity)

Input sample rate

This is not a FIR stage, so FIR delays are zero

ADC scale factor & normalization frequency

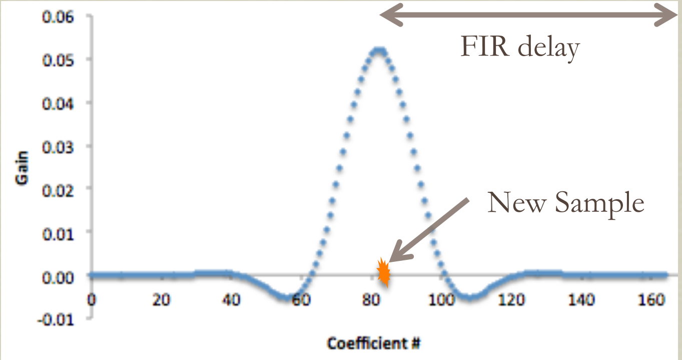
Dataloggers

FIR Filtering - Oversampling and Decimation

- Older dataloggers relied on an analog anti-alias low-pass filter to prevent aliasing during sampling.
- Modern dataloggers oversample and decimate data using digital Finite Impulse Response (FIR) filters. FIR filtering extends the passband up to 70-90% of the Nyquist frequency.
- Oversampling and FIR Decimation also mitigates quantization noise.

Dataloggers

FIR Filters



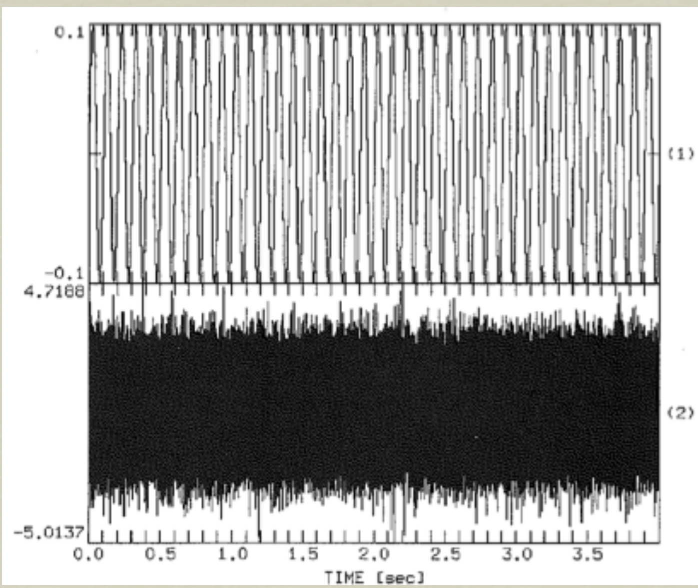
- are digital filters typically represented in the time domain using coefficients.
- are weighted averages – they decimate by averaging the amplitudes of surrounding input samples to obtain output samples (stable).
- must average future samples, so there is a delay caused by waiting for these future samples to arrive. Dataloggers correct time tags for this delay.
- must be normalized (the coefficients must sum to 1) or else they will change the gain of each sample.

Dataloggers

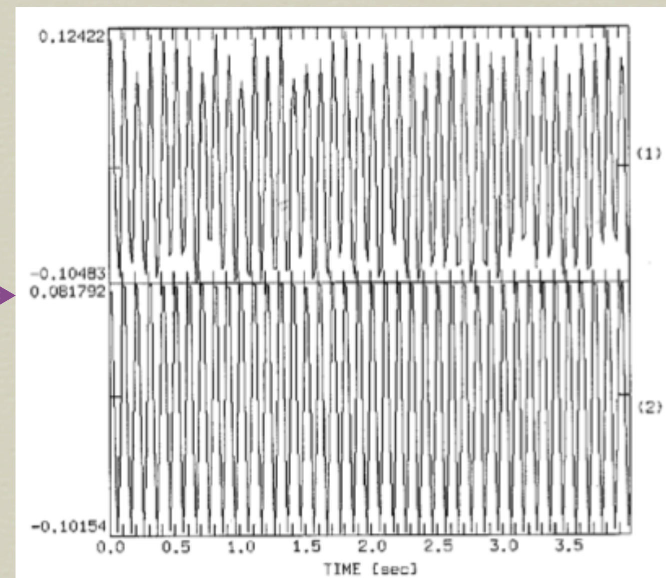
Because FIR Filters average amplitudes over neighboring samples, they mitigate quantization error.

Input Signal

Gaussian Noise + Sine Wave



Output Signal



Simple sampling

Oversampling & Decimation

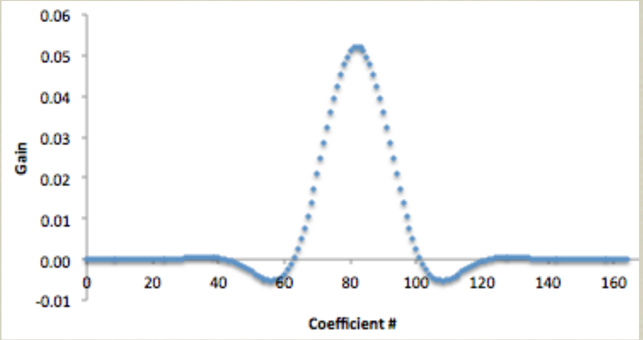
From Scherbaum (1996)

Dataloggers

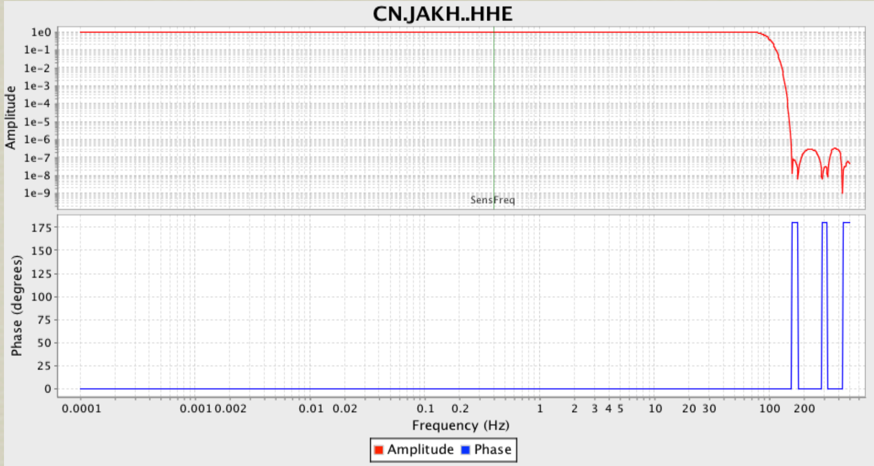
FIR filters are

- zero phase (they don't alter phase),
- low-pass filters with
- unity gain (they don't alter amplitude).

Their decimation factor reflects how frequently they are applied to the input time series.



FT
<==>



SEED FIR Stages

```
#
#               +-----------------------------------+
#               |      Response (Coefficients)      |
#               |        CN  JAKH         HHE       |
#               |     02/16/2010 to 12/31/2599      |
#               +-----------------------------------+
#
B054F03     Transfer function type:                  D
B054F04     Stage sequence number:                   3
B054F05     Response in units lookup:                COUNTS - digital counts
B054F06     Response out units lookup:               COUNTS - digital counts
B054F07     Number of numerators:                    165
B054F10     Number of denominators:                  0
#              Numerator coefficients:
#                 i  coefficient   error
B054F08-09        0  -4.04791E-10  +0.00000E+00
B054F08-09        1  -1.39029E-10  +0.00000E+00
B054F08-09        2  +6.72800E-10  +0.00000E+00
B054F08-09        3  +2.75797E-09  +0.00000E+00
B054F08-09        4  +7.54651E-09  +0.00000E+00
```

Digital stage, units of Counts

Normalized coefficients

```
B054F08-09      162  +6.72800E-10  +0.00000E+00
B054F08-09      163  -1.39029E-10  +0.00000E+00
B054F08-09      164  -4.04791E-10  +0.00000E+00
#
#               +-----------------------------------+
#               |            Decimation             |
#               |        CN  JAKH         HHE       |
#               |     02/16/2010 to 12/31/2599      |
#               +-----------------------------------+
#
B057F03     Stage sequence number:                    3
B057F04     Input sample rate (HZ):                   3.0000E+04
B057F05     Decimation factor:                        00015
B057F06     Decimation offset:                        00000
B057F07     Estimated delay (seconds):               +2.7333E-03
B057F08     Correction applied (seconds):            +2.7333E-03
#
#               +-----------------------------------+
#               |     Channel Sensitivity/Gain      |
#               |        CN  JAKH         HHE       |
#               |     02/16/2010 to 12/31/2599      |
#               +-----------------------------------+
#
B058F03     Stage sequence number:                   3
B058F04     Sensitivity:                             +1.00000E+00
B058F05     Frequency of sensitivity:                +4.00000E-01
B058F06     Number of calibrations:                  0
```

Input sample rate & decimation factor

FIR delay (positive)
= (#coeffs - 1) / (2 * Input sample rate)
for symmetric (acausal or zero-phase) filters

Unity gain at the normalization frequency

Dataloggers

Analog Anti-Alias Filters

- Some dataloggers have an analog anti-alias filter between the preamp and the ADC. It is described using poles an zeros. The following example is from the Nanometrics Taurus.

```
#
#              +                  +--------------------------------------------+
#              +                  |   Response (Poles & Zeros),  NN101 ch BHZ   |
#              +                  +--------------------------------------------+
#
B053F03     Transfer function type:                A [Laplace Transform (Rad/sec)]
B053F04     Stage sequence number:                 2
B053F05     Response in units lookup:              V - Volts
B053F06     Response out units lookup:             V - Volts
B053F07     A0 normalization factor:               1.036270E+04
B053F08     Normalization frequency:               1.000000E+00
B053F09     Number of zeroes:                      0
B053F14     Number of poles:                       1
#               Complex zeroes:
#                 i  real           imag           real_error     imag_error
#               Complex poles:
#                 i  real           imag           real_error     imag_error
B053F15-18    0 -1.036270e+04  0.000000e+00  0.000000E+00  0.000000E+00
#
#              +                     +---------------------------------------+
#              +                     |        Channel Gain,  NN101 ch BHZ     |
#              +                     +---------------------------------------+
#
B058F03     Stage sequence number:                 2
B058F04     Gain:                                  3.999988e-01
B058F05     Frequency of gain:                     1.000000e+00 HZ
B058F06     Number of calibrations:                0
```

Input and Output units are Volts

Gain need not be unity

Dataloggers

Infinite Impulse Response (IIR) filters

- Some dataloggers have an optional Infinite Impulse Response (IIR) filter available.
- IIR filters are computationally fast compared to FIR filters – they depend on fewer samples
- A value calculated by an IIR filter includes previous output samples to which the IIR filter has already been applied one or more times. Because of this, they can be notoriously unstable.
- IIR filters are not linear phase – they alter the phase of the input signal

Dataloggers

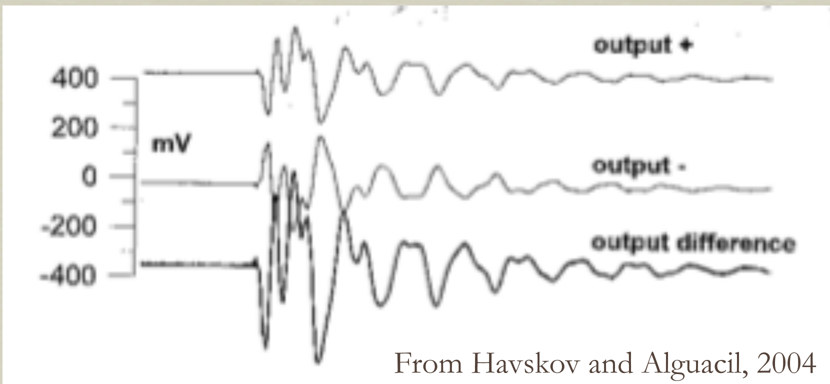
Infinite Impulse Response (IIR) filters

- IIR filters are great for real-time phase picking – they introduce little delay and can produce minimum-phase arrivals for easier picking.
- Data filtered by IIR filters is appropriate for in-house analysis, but should not be archived as the main data stream.
- In SEED, IIR filters should be represented as a digital pole-zero response stage because this introduces less round off error than a coefficient representation.

Single vs. Differential Input & Output

- Sensors may be made with
  - one signal output wire plus ground (single-ended) or
  - two signal output wires plus ground (double-ended).
- Double-ended output is called “Differential output” because the signal on the second output is inverted so that the two signals can be differenced at the datalogger. This cancels noise induced in the cable leading from sensor to datalogger.

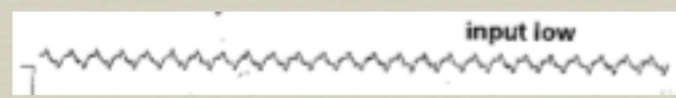
Differential Output



From Havskov and Alguacil, 2004

- The "output +" is the original sensor signal.
- "output –" is the inverted signal from the second sensor output.
- Trace 3 is the difference of the two output traces

If the noise (right) were to be induced in the sensor cable, it should be similar on both output wires. Taking the difference of the output traces subtracts out the noise, but adds the signal.



Single vs. Differential Input & Output

- Dataloggers may be made with either single-ended or differential input.

- Sensors with differential output may specify their sensitivities either in the form of “2 * 750 V/m/s” or “1500 V/m/s differential; 750 V/m/s single-ended”.

Sensor

Datalogger

Output → Input ↓	Differential	Single ended
Differential	+ → + – → – ground → ground gain = 1.0	+ → + – → ground ground → ground gain = 1.0
Single ended	+ → + – → NC ground → ground gain = 0.5	+ → + ground → ground gain = 1.0

From Havskov and Alguacil, 2004

- Connecting a differential output sensor to a single-ended input datalogger decreases the amplitude by a half.

Nominal Response Library (NRL)

What is the NRL?

- Library of manufacturers' recommended nominal instrument responses
  - SEED RESP files
  - Help matching an instrument's configuration with the correct response
  - Notes describing instrument and response differences

Nominal Response Library (NRL)

How is the NRL constructed?

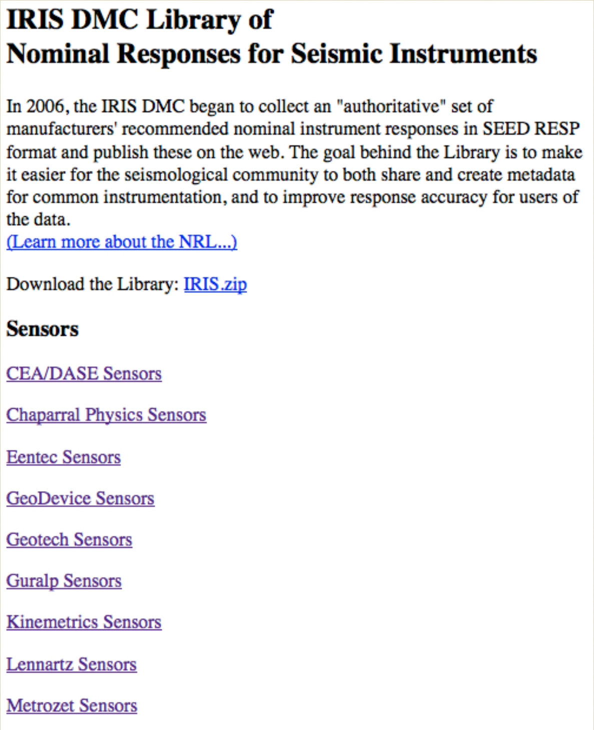
- Response information retrieved from manufacturer
- Instruction file links instrument configuration with pole/zero or FIR coefficient files
- Generate RESP files from instruction file
- Accuracy checking

When Do I Need a Custom Response?

- Update your Nominal Response if:
  - you have calibration info
  - your accelerometer full scale voltage and/or clip level differs
  - you have a passive sensor and
    - your resistors differ
    - you need to take sensor-amplifier impedance into account
  - You’ve set a software gain on your datalogger

A Few Tools for Retrieving Response Information

- Nominal Response Library
  - http://ds.iris.edu/NRL/
  - Manufacturers' recommended responses
  - RESP format (http://ds.iris.edu/ds/nodes/dmc/data/formats/resp/)



IRIS DMC Library of
Nominal Responses for Seismic Instruments

In 2006, the IRIS DMC began to collect an "authoritative" set of manufacturers' recommended nominal instrument responses in SEED RESP format and publish these on the web. The goal behind the Library is to make it easier for the seismological community to both share and create metadata for common instrumentation, and to improve response accuracy for users of the data.
(Learn more about the NRL...)

Download the Library: IRIS.zip

Sensors

CEA/DASE Sensors

Chaparral Physics Sensors

Eentec Sensors

GeoDevice Sensors

Geotech Sensors

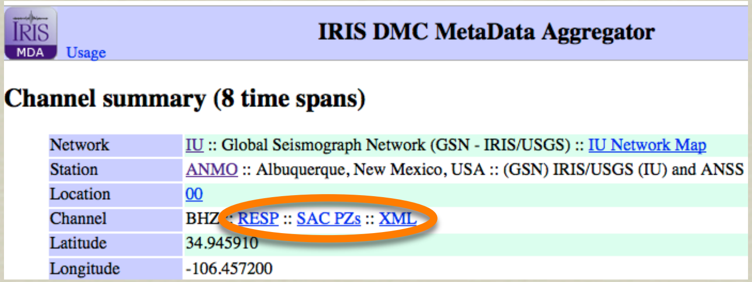
Guralp Sensors

Kinemetrics Sensors

Lennartz Sensors

Metrozet Sensors

A Few Tools for Retrieving Response Information

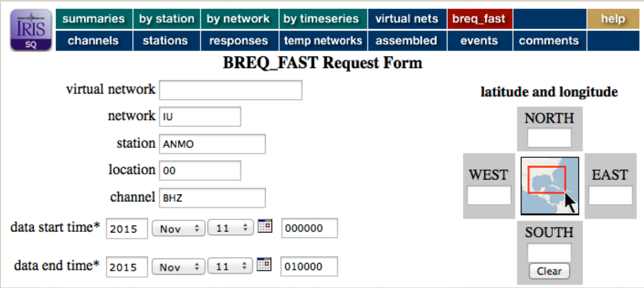


- Metadata Aggregator
  - http://ds.iris.edu/mda/
  - Response information for data archived at IRIS
  - Formats
    - RESP (http://ds.iris.edu/ds/nodes/dmc/data/formats/resp/)
    - SAC PoleZero
      - Displacement response in nm
      - Poles and zeros in radians
      - CONSTANT = total sensitivity * A_0
    - FDSN StationXML (http://www.fdsn.org/xml/station/)

A Few Tools for Retrieving Response Information

- IRIS Web Services
  - http://service.iris.edu/
  - Response information for data archived at IRIS
  - Formats
    - station service (text & FDSN stationXML)
    - resp service (RESP)
    - sacpz service (SAC pole zero format)

A Few Tools for Retrieving Response Information



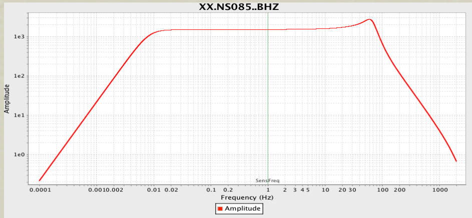
- breq_fast
  - http://ds.iris.edu/SeismiQuery/breq_fast.phtml
  - Response information for data archived at IRIS
  - Formats
    - RESP
    - Dataless SEED (http://www.fdsn.org/seed_manual/SEEDManual_V2.4.pdf)
    - Full SEED (http://www.fdsn.org/seed_manual/SEEDManual_V2.4.pdf)

A Few Tools for Writing SEED Metadata

- Antelope
  - http://www.brtt.com/software.html
  - Native response format: CSS
    (see Antelope man page for "response")
- Portable Data Collection Center (PDCC)
  - http://ds.iris.edu/ds/nodes/dmc/software/downloads/
  - Native response format: RESP from the NRL
- Station Information System (SIS)
  - USGS regional network partners
  - Native response format: RESP from the NRL, stationXML

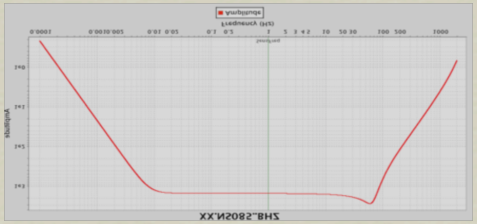
Response Correction

- An instrument response can be removed from data by
  - Deconvolution in the time domain
  - Division of amplitude spectra in the frequency domain



Data spectrum

·



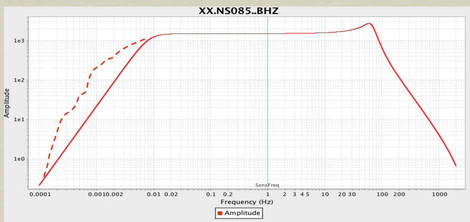
1/Amplitude response

=

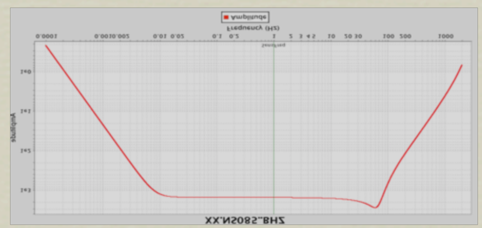
ideally…

Response Correction

- But suppose your data has extra noise at long period



·



=

Data spectrum

1/Amplitude response

in reality…

- Limiting the frequency band with a bandpass filter can help

- Spectral prewhitening can sometimes help by evening out the spectrum

A Few Tools for Response Correcting Data

- IRIS timeseries web service
  http://service.iris.edu/irisws/timeseries/1/
- SAC
  - Software request
    http://ds.iris.edu/ds/nodes/dmc/forms/sac/
  - Examples
    http://www.eas.slu.edu/eqc/eqc_cps/TUTORIAL/RESPONSE/index.html
    http://geophysics.eas.gatech.edu/people/jwalter/sacresponse.html
- Matlab Example
  http://www.mathworks.com/matlabcentral/fileexchange/48966-rawseismicinstrumentcorrection/content/RawSeismicInstrumentCorrection.m

Tools for Verifying Responses

- evalresp (http://ds.iris.edu/ds/nodes/dmc/software/downloads/)
  - Command line C program
  - Reads SEED RESP files
  - Sanity checking for basic sensitivity
  - Summarizes output sample rate & units
  - Creates ASCII files containing amplitude and phase spectra.

Verifying Responses with evalresp

- To verify responses in a new dataless SEED file
  - Create RESP files using the rdseed program (http://ds.iris.edu/ds/nodes/dmc/software/downloads/)
  - Run evalresp on each RESP file, directing the output to a file
  - For that output file, egrep -i "(FAIL|ERROR)" output_file

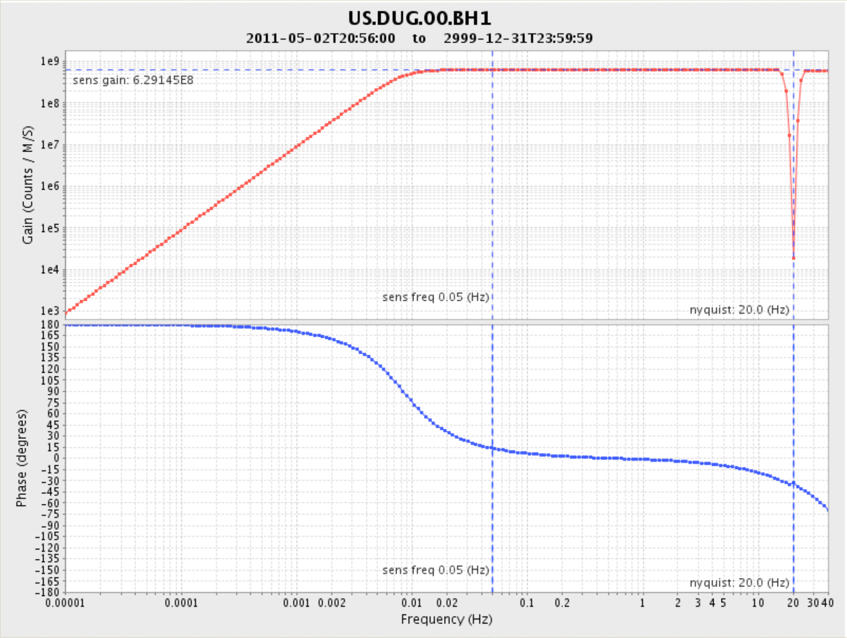
Tools for Verifying Responses

- Also, verify the response curve graphically
  - JPlotResp (http://ds.iris.edu/ds/nodes/dmc/software/downloads/)
    - Reads RESP files
    - Runs evalresp
    - Bode plots (stages plotted as composite or separately)
    - Mouse-over discovery of curve values
  - Metadata Aggregator
    - Bode plots

Verifying Responses

- Do the high- and low-frequency corners look correct?
- Does this look like a velocity response?
- Is the normalization frequency within the passband?
- Is the plotted Nyquist frequency consistent with sample rates in the dataless and miniSEED?

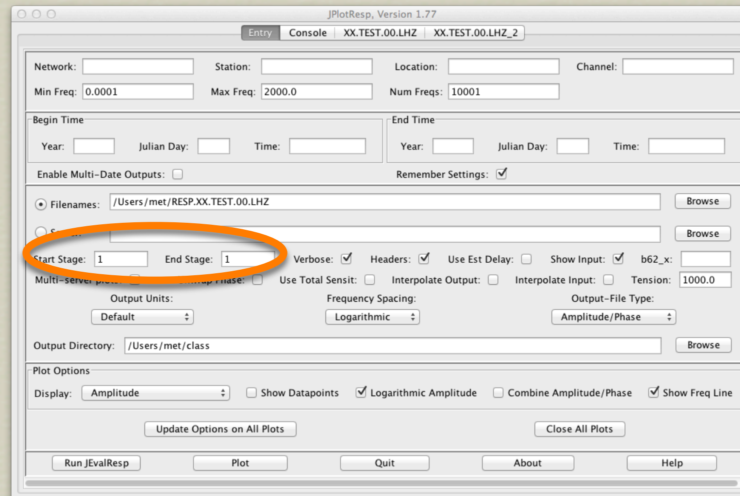
Metadata Aggregator

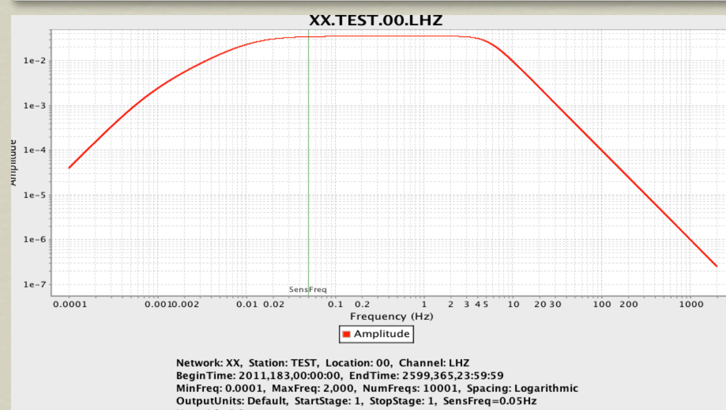


STS-2 Sensor

Finding A_0 with JPlotResp

- Create a copy of your RESP file and set A_0 and the sensor sensitivity to 1.
- Use JPlotResp to plot just stage 1 of your edited RESP.
- Use "mouseover" to find the amplitude of your pole-zero curve at your normalization frequency (SensFreq). A_0 is the inverse of this.
- Restore the sensor sensitivity in your RESP and include your new A_0.
- Replot the sensor stage to make sure the amplitude is now the sensor sensitivity.



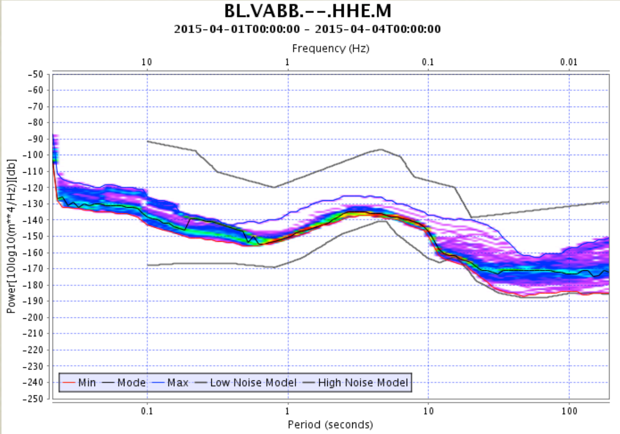


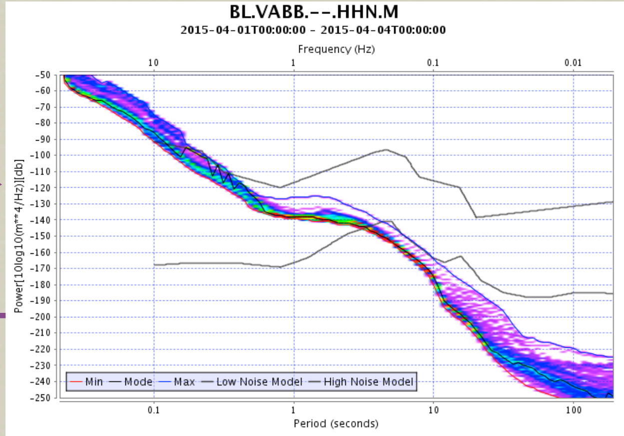
Tools for Verifying Responses

- MUSTANG data quality metrics
  - http://service.iris.edu/mustang/
  - The following metrics operate on response-corrected data. Unexpected results may indicate incorrect response information
    - noise-psd
    - noise-pdf
    - noise-mode-timeseries
    - measurements
      - dead_channel_exp
      - pct_below_nlnm
      - pct_above_nhnm
      - transfer_function

Pole Typo

Pct Below Nlnm Metric			
value	target	start	end
0.34	BL.VABB..HHE.M	4/1/15 0:00	4/1/15 23:59
0.60	BL.VABB..HHE.M	4/2/15 0:00	4/3/15 0:00
0.76	BL.VABB..HHE.M	4/3/15 0:00	4/4/15 0:00
45.28	BL.VABB..HHN.M	4/3/15 0:00	4/4/15 0:00
44.50	BL.VABB..HHN.M	4/2/15 0:00	4/3/15 0:00
43.96	BL.VABB..HHN.M	4/1/15 0:00	4/1/15 23:59
43.40	BL.VABB..HHZ.M	4/2/15 0:00	4/3/15 0:00
43.64	BL.VABB..HHZ.M	4/3/15 0:00	4/4/15 0:00
42.83	BL.VABB..HHZ.M	4/1/15 0:00	4/1/15 23:59





HHE poles:

```
Complex poles:
i  real          imag
0  -3.70200E-02  +3.70200E-02
1  -3.70200E-02  -3.70200E-02
2  -2.22110E+02  +2.22110E+02
3  -2.22110E+02  -2.22110E+02
```

HHN poles:

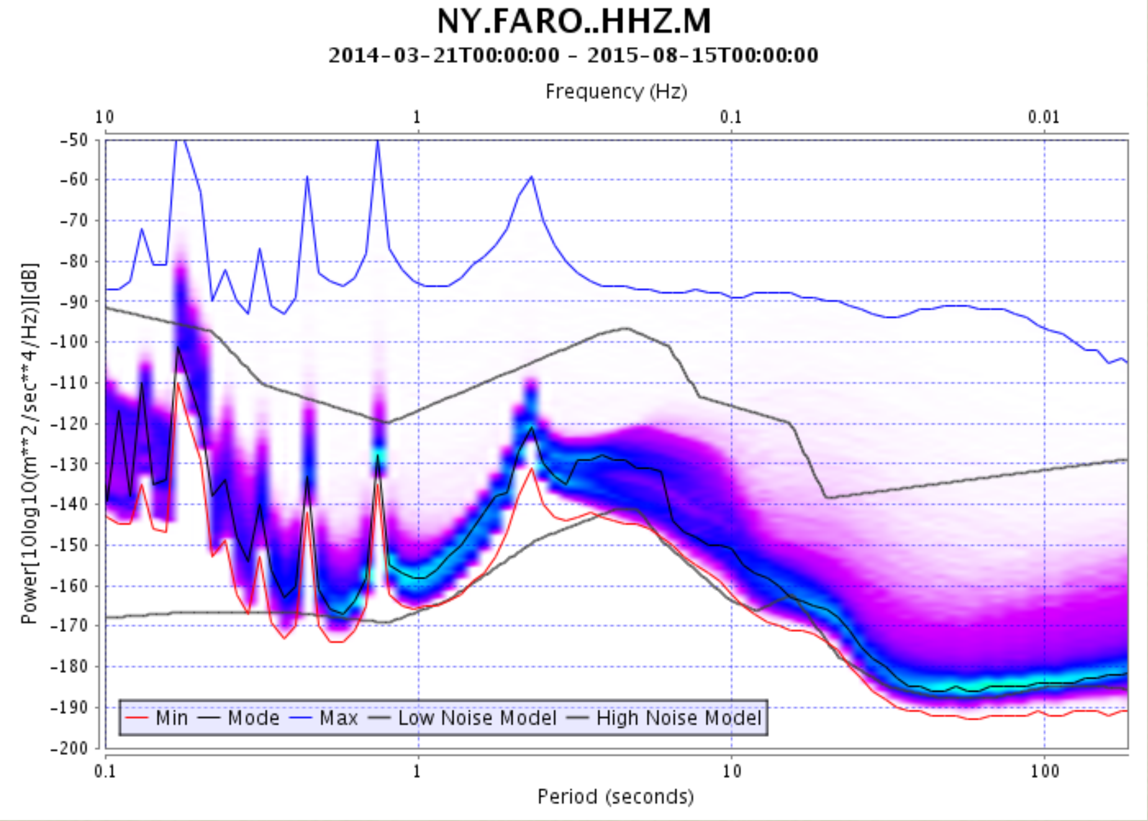
```
Complex poles:
i  real          imag
0  -3.70200E-02  +3.70200E-02
1  -3.70200E-02  -3.70200E-02
2  -2.22110E-02  +2.22110E-02
3  -2.22110E-02  -2.22110E-02
```

Sign error

Incorrect FIR Cascade

The data sample rate was 100 sps, but the FIR cascade was for a 1 sps stream.
FIR responses have lobes at f>Nyquist. Since there no energy in 1 Hz data at those frequencies, you don't see the lobes when you instrument correct…

…unless you remove this response from higher sample rate data that does have energy there!

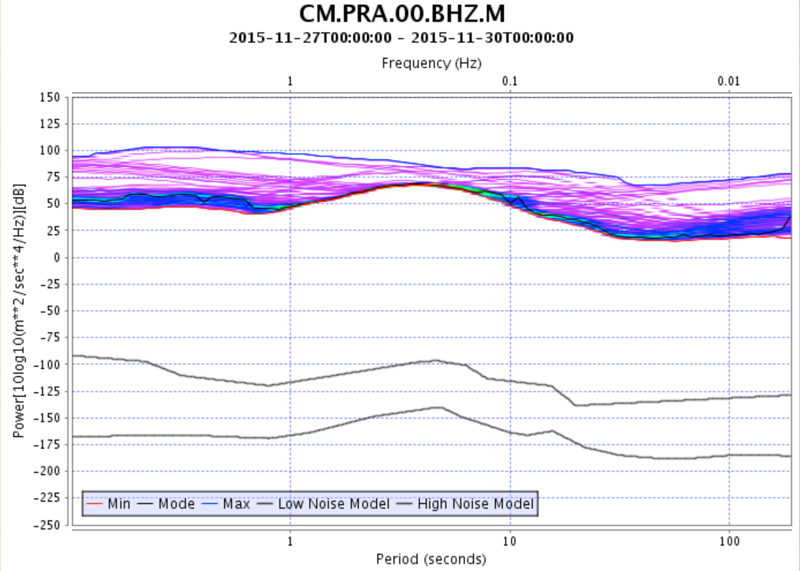


Incorrect Sensor Response

```
"Pct Above Nhnm Metric"
"value","target","start","end","lddate"
"100.000","CM.PRA.00.BHZ.M","2015/11/27
"100.000","CM.PRA.00.BHZ.M","2015/11/28
"100.000","CM.PRA.00.BHZ.M","2015/11/29
"100.000","CM.URI.00.BHZ.M","2015/11/27
"100.000","CM.URI.00.BHZ.M","2015/11/28
"100.000","CM.URI.00.BHZ.M","2015/11/29
```

This MUSTANG query retrieved values of pct_above_nhnm measurements having 20% or more energy above the New High Noise Model for the CM network.

The sensor response archived was a placeholder until the needed instrument can be added to the NRL.



References

- Havskov, J. and Alguacil, G., 2004, *Instrumentation in Earthquake Seismology*: Modern Approaches in Geophysics v. 22, Springer, 358 p.

- Sherbaum, F., 1996, *Of Poles and Zeros*: Modern Approaches in Geophysics v. 15, Kluwer Academic Publishers, 256 p.

Contacting Me

met@iris.washington.edu