Working with simulation filters

Simulation filters transform records of an instrument, preferably a broadband instrument, into a simulated record of another instrument. Before generating such simulation filters it should be cleared whether the transfer function of the recording instrument allows the simulation of the desired one. The recording system must provide enough signal energy in all frequency ranges of the simulated instrument. E.g. from records of STS-2 seismometers simulations of short- and longperiod instruments like WWSSN-SP and SRO-LP may be computed, but inversely, e.g. from WWSSN-SP to STS-1 is definitely not possible. In principle, the construction of simulation filters is rather simple if we use the poles-and-zeros definition of filters. These numbers can be directly written into an SH-FFT-filter file (see filter formats). A simulation filter is created by taking the transfer function of the recording instrument (e.g. an STS-2), inverting it (take reciprocal value of the normalization, exchange poles and zeros), multiply it with the transfer function of the simulated instrument (e.g. WWSSN-SP) and shorten all possible values of the rational expression. The resulting filter (example: STS2-to-WWSSN-SP) may be used for application in SHM. Please make sure that all elements (normalization constants, poles-and-zeros definition of the filter) refer to the same recording input (usually either velocity or displacement). In practice, this is quite often a reason for confusion. For a quick view of the amplitude response of your FFT filters use graphic display of FFT filters. If you want to create recursive simulation filters, use the utility programs of SH/SHM (in *$SH_UTIL*, programs *butrec*, *simrec*) for some standard cases or follow instructions for computing recursive filters given e.g. in ???. A tool for creating a set of FFT filters for your recording equipment is provided in *$SH_UTIL*.

When using SHM, the filter files need to match a naming convention so that the files can be found on request. The filter files must be located either in *$PWD* or in *$SH_FILTER*. Looking for the filter files SHM assumes file names of a defined format:

FFT-files: *<rec-instr>_S+<sim-filter>.FLF*

<rec-instr>
: Descriptor of the recording instrument. By default this is <station>:<channel>:<comp>

<station>
: Station name (examples: BFO, GEC2)

<channel>
: 2-character SEED channel code (examples: BH, HH)

<comp>
: Component (examples: Z, N)

<sim-filter>
: Descriptor of simulated instrument (examples: WOODAND, G_WWSSN_SP)

recursive files: *<smprate>HZ_<rec-instr>_S+<sim-filter>.FLR*

<smprate>
: Integer sample rate in Hz (examples: 20, 40)

<rec-instr>
: See above

<sim-filter>
: See above

All filenames must have uppercase letter only! Hyphens in filter filenames are not supported by SHM. As an example, a legal filename for an FFT simulation filter for `WWSSN-LP` recorded at station `RAR-BH-Z` would be *RAR:BH:Z_S+WWSSN_LP.FLF*. Since it is rather inconvenient to have separate filter files for each stream, it is possible to combine several streams to a single class name within a lookup table. Class names found in the lookup table replace the individual stream names within the filenames. SHM searches two lookup tables:

1. an internal table (implemented in the source code) and an
2. external table in the file *$SH_INPUTS/filter_lookup.txt*.

The internal table maps all GRF-BH-streams (e.g. `GRA1-BH-Z`, `GRB1-BH-N`) to the name `GRF`, all GRSN-BH-streams (e.g. `BFO-BH-Z`, `CLZ-BH-E`) to the name `GRSN` and all GERESS-HH-streams (e.g. `GEA0-HH-Z`) to the name `GERESS`. Due to this lookup table the simulation filter for `WWSSN-LP` on stream `BFO-BH-Z` is *GRSN_S+WWSSN_LP.FLF* instead of *BFO:BH:Z_S+WWSSN_LP.FLF*. In the external lookup table arbitrary mappings of additional stations can be added. For example if you have two stations `XYZ` and `XXX`, both with an instrument `CMG40T` which has the same transfer function for all three components then you should add the lines

```
       XYZ-BH-Z CMG40T
       XYZ-BH-N CMG40T
       XYZ-BH-E CMG40T
       XXX-BH-Z CMG40T
       XXX-BH-N CMG40T
       XXX-BH-E CMG40T
```

to the external table. Then the appropriate simulation filter for `WWSSN-LP` would be *CMG40T_S+WWSSN_LP.FLF*.

The general systax of a line in filter_lookup.txt is since version SHM-2.2f:

```
       <chan> <filname> [[<stime> <etime>] <flags>]
```

where <chan> is the channel string (like XYZ-BH-Z in example above), <filname> is the name given to the filter (like `CMG40T` in the example above), <stime> and <etime> specifies the valid time window (start and end time in SH time format) of the information given in this line. Three dots (...) as time value indicate an empty time, which means no time boundary. If the equipment at a station site changes at some time a new entry for this station has to be added in filter_lookup.txt. <flags> may contain the following strings:

-FORCETAB-
    indicates that the simulation filters of this station are given as tabulated (FAP) filter files (.FLT-files). A utility for creating these files is *$SH_UTIL/make_sim4fap.csh*.

-FORCEFFT-
    indicates that the simulation filters of this station are given as FFT filter files (.FLF-files) regardless of what default filter type is set.

-ALERT-
    marks output traces of this filter entry in red. This could be used to remind the user that this entry is possibly incorrect if you are testing filters or if you have incomplete information on your recording system.

Example entries of a filter_lookup.txt file are given below:

```
       XYZ-BH-Z CMG40T ... 23-Jul-2002
       XYZ-BH-Z STS2 23-Jul-2002 ...
       OLD-SH-Z OLDINSTR ... ... -FORCETAB-ALERT-
```

back to documentation index