r\_est\_from\_clip\_simplified Estimate sound source location from microphone array data

This is the core function of Muse. It takes the voltage signals from the microphone array, and outputs an estimated sound source location.

[r\_est,rsrp\_max,rsrp\_grid,a,vel,N\_filt,V\_filt,V,rsrp\_per\_pair\_grid]= r\_est\_from\_clip\_simplified(v,fs,f\_lo,f\_hi,Temp,x\_grid,y\_grid,in\_cage,R, verbosity)

Inputs:

v: An N x K array of microphone signals, N the number of time points, K the number of microphones.

fs: The sampling frequency of the audio data, in Hz.

f\_lo: The lower bound of the frequency band used for analysis, in Hz. Frequency compenents outside this band are zeroed after the data is intially FFT'ed.

f\_hi: The upper bound of the frequency band used for analysis, in Hz.

Temp: The ambient temperature at which data was taken, in degrees celsius.

x\_grid: The reduced steered response power (RSRP) is calculated at every point on a grid. This gives the x-coordinate of each point on the grid. It is a 2D array.

y\_grid: A 2D array of the same shape as x\_grid, giving the y-coordinate of each point at which RSRP is to be calculated. The points indicated by x\_grid and y\_grid are assumed to be in a Cartesian coordinate system.

in\_cage: A logical array of the same size as x\_grid, indicating which grid points are in the interior of the cage. In theory, the RSRP would only be calculated at these points. In reality, this argument is not used.

R: The microphone positions in 3D space, a 3 x K array, in meters, in a standard right-handed Cartesian coordinate system. Positive z coordinates are assumed to be above the plane of the x\_grid, y\_grid points.

verbosity: An integer indicating how much information about intermediate computations should be output to the console and/or figures. A value of 0 indicates no output, higher values indicate more output.

Outputs:

r\_est: A 2 x 1 array giving the estimated position of the sound source.

rsrp\_max: The value of the RSRP at r\_est. This will have units of arbs^2, if v is in arbs.

rsrp\_grid: A 2D array of the same shape as x\_grid, giving the RSRP at every point in the grid. rsrp\_max gives the largest value in rsrp\_grid, and r\_est the (x,y) point at which this value occurs. This will have units of arbs^2, if v is in arbs.

a: A 1 x K array of values that estimates the gain of each microphone, in same units as v.

vel: The speed of sound in air used in the calculation, as computed from Temp. In m/s.

N\_filt: The number of frequency values in the passband of the band-pass filter, given f\_lo, f\_hi, and (implicit) spacing between frequency samples used.

V\_filt: The FFT of the values in v, after band-pass filtering, in the same units as v.

V: The FFT of the values in v, \*before\* band-pass filtering, in the same units as v.

rsrp\_per\_pair\_grid: A 3D array, Nx x Ny x Npairs, where Nx x Ny is the size of x\_grid, and Npairs is the number of unordered pairs of microphones, not including self-pairs (K\*(K-1)/2). This contains the RSRP calculated for each microphone pair. Summing rsrp\_per\_pair\_grid across pages yields rsrp\_grid. This will have units of arbs^2, if v is in arbs. The function mixing\_matrix\_from\_n\_mics(K) can be used to determine which pair corresponds to which microphones.

# rsrp\_grid\_from\_clip\_and\_xy\_grids

[rsrp\_grid,a,vel,N\_filt,V\_filt,V,rsrp\_per\_pair\_grid]= rsrp\_grid\_from\_clip\_and\_xy\_grids(v, fs, f\_lo, f\_hi, Temp, x\_grid,y\_grid, R, verbosity);

# argmax\_grid

[r\_argmax,objective\_max]= argmax\_grid(x\_grid,y\_grid,objective,gate\_grid)

Returns the point on the grid where the objective function is maximized.

x\_grid: m x n, x values at each point on the grid

y\_grid: m x n, y values at each point on the grid

objective: m x n, objective function values at each point on grid (in this case objective = rsrp\_grid).

gate\_grid: m x n, a boolean array that is true for "admissible" grid points. If absent or empty, all grid points are considered admissible

r\_argmax: 2x1, containing the x and y where the objective function is largest, considering only the admissible points.

objective\_max: scalar, value of objective at r\_argmax.

# xcorr\_raw\_from\_dfted\_clip

Calculates the unnormalized cross-correlation between all mic pairs

[xcorr\_raw,tau\_line]=xcorr\_raw\_from\_dfted\_clip(V,dt,M,verbosity);

Where V is the fft of the original recorded signal of all the mics (each columns of V is related to a mic). xcorr\_raw is the unnormalized cross-correlation. (not sure what tau\_line represents at this point).

# rsrp\_from\_xcorr\_raw\_and\_delta\_tau

Calculates sum of the upper cross terms in the formula for steered response power (SRP) for a set of time shifts in dt\_arr.

[rsrp,rsrp\_per\_pair]=rsrp\_from\_xcorr\_raw\_and\_delta\_tau(xcorr\_raw,tau\_line,tau\_diff);

Where xcorr\_raw is N x n\_pairs, and contains the raw cross-correlation between a pair of signals. Note that this is unnormalized.

tau\_line is N x 1, and gives the time lag (in s) for each row of xcorr\_raw.

tau\_diff is n\_pairs x n\_r, and gives the predicted relative time lag for each pair of signals, in s. n\_r is an arbitrary number. These lags are used to interpolate into tau\_line to get values of xcorr\_raw at a particular set of lags.

RSRP, on return, is 1 x n\_r, and gives the sum of the upper cross terms in the steered response power. The SRP is a sum across all pairs of signals, including self-pairs. The return value of this function excludes the self pairs, and only includes one of (i,j) and (j,i) (hence the "upper"). We're using the definition of the SRP given in:

Zhang C, Florencio S, Ba D, Zhang Z (2007) Maximum likelihood sounds source localization and beamforming for directional microphone arrays in distributed meetings. If a particular position leads to delayed signals x(n,k), where n indexes time points and k indexes signals, the SRP is:

SRP =

So here we're computing

SRP =

we do this fast by interpolating into the pre-computed values in xcorr\_raw.

# xcorr\_raw\_from\_dfted\_clip

Calculates the cross correlation between each pair of mics, such that

Xcorr\_raw\_this= V(:,i\_mike).\*conj(V(:,j\_mike));

Where V is the fft of the original recorded signal of all the mics (each columns of V is related to a mic).

# pad\_at\_high\_freqs

Fills the original complex signal with zeros for very high frequencies, producing an output with N-padded elements. It increases resolution in time domain.

Xcorr\_raw\_this\_padded=pad\_at\_high\_freqs(Xcorr\_raw\_this,N\_line);