Matt Pistacchio

661114637

**Automatic Reception of Acoustic Sounds Function Specification**

**Software Design Document**

**Brief:**

ARAS implementation currently incorporates two primary scripts: Training.py and Testing.py. Training.py uses Function #1 (see **Table 1** for all function ID references) to simulate a specified number of signals, and uses Function #3 to store its output adjacent to a specified class ID. Testing.py calls Functions #1&2 and identifies the class ID which demonstrates the lowest output of Function #4.

**Table 1: System Function Specifications**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function #** | **Function Name** | **Input(s)** | **Output(s)** | **Description** |
| 1 | sim\_data | sample frequency (fs), number of samples (Np), signal frequency vector (f), signal amplitude (A), number of signals (Ns), signal to noise ratio (snr) | Amplitude time series vector (S) | Derives sample period and signal duration from fs and Np, and uses those constants to compute empty time (t) and signal (S) vectors. In two separate subsequent loops, the S vector is first cleared, and in the second, a nested loop is used to fill each index of S with a sinusoid with amplitude index of vector A, frequency index of vector f, and time index of vector t for each signal in the range specified by Ns. The snr variable defines the amount of Gaussian noise to add to the sinusoids, such that . |
| 2 | PowerSpectrum | signal vector (S) | power spectrum vector (P) | Computes P as a power vs. frequency vector of a waveform using a Fast Fourier Transform (FFT) upon the contents of S. |
| 3 | FeatureExtract | signal vector (S) | feature struct (Feat) | Computes the following features (parameters of Feat) after calling the power spectrum function:   * the fundamental frequency (ff) * the ratio of the second harmonic and the fundamental powers (ratio21) * the ratio of the third harmonic and the second powers (ratio32) * the ratio of the total power in the first half of the P spectrum and the second (halfnhalf) * the center of gravity computation of each P index * the bandwidth computation of each P index |
| 4 | Distance | two Feat structs of two different classes of signals (F1,F2) | RMSE\* distance vector between the two classes’ features (distance) | Computes RMSE calculation between each parameter of the Feat structs. |
| 5 | ClassMean | vector of all Feat structs which are generated by Training.py (Feats) | mean of all the different classes’ Feat parameters (delimited and exported to separate text file) | Searches for all features in each class ID, finds them all, and outputs the mean of each class. |
| 6 | ClassStDev | vector of all Feat structs which are generated by Training.py (Feats) | standard deviation of all the different classes’ Feat parameters (delimited and exported to separate text file) | Searches for all features in each class ID, finds them all, and outputs the standard deviation of each class. |

\*Whether or not to implement a weighted RMSE computation is currently under review.

**Assorted Formulae**

Let the following vector be an amplitude time series of a signal,

Let be the Discrete Fourier Transform of .

Let be the raw power spectrum,

To create a background estimate for waveform normalization, the following steps are executed,

1. Define the convolution kernel,

Nominally, and .

Let the vector be the smooth power spectrum.

Then, let be the normalized power spectrum, where

To detect all the tones within the normalized power spectrum, find all the local maxima greater than the threshold , where, nominally, .

Let the vector, where

if and and ;

is the number of values in that satisfy the above criteria.

Also store the vector , equal to the indices of satisfying the local max criteria.

Let the following be features of the power spectrum:

1. The fundamental frequency, or where is the bandwidth of each power spectral bin. , where is the digital sample rate of the original signal, .
2. The ratio of the second harmonic and the fundamental powers, or
3. The ratio of the third harmonic and the second powers, or
4. The ratio of the total power in the first half of the power spectrum and the second, or .
5. The center of gravity of the raw power spectrum, or .
6. The effective bandwidth of the raw power spectrum, or

The feature extraction routine reduces the dimensionality of the original signal representation. The classification process uses a dictionary to pair class IDs with each feature set. Each feature in the dictionary can be represented by the following notation:

, where , , and .

Furthermore,

represents the number of features, e.g. 6;

the number of classes;

the number of sample feature sets per class.

Using these parameters in the dictionary , classification is performed by minimizing the weighted mean squared error (MSE) defined below:

where

is the feature set from a signal with an unknown class ID, and

is the mean of feature for class ,

and

is the standard deviation of feature for class .

The classification decision is the class for which is a minimum and less than or equal to a threshold . The null set is selected (i.e., an unknown decision) if the minimum is greater than .

Regarding the SNR of the tones created by sim\_data, the variable snr can be defined as

where

and is the scale factor to be used to apply noise to the resultant sinusoids. Since the snr variable is used as an input to the overall function, the scaling factor can simply be found through

and is then applied to the resultant sinusoid in the form of