Interpolation & Decimation

This document is intended to be a guide for the various interpolation and decimation related functions that are available in MATLAB.

Linear and Spline Interpolation

The first function that we will look at is `interp1.m`. Specifically we look at interpolation that fits a local polynomial to the data in a small window to fill in the extra information. The synopsis of this function is given below.

```
YI = INTERP1(X,Y,XI,'method') specifies alternate methods.
    The default is linear interpolation.
    X: Input time interval
    Y: Input time series
    XI : Desired output time interval
    YI : Interpolated time series

    Available methods are:

      'nearest'  - nearest neighbor interpolation
      'linear'   - linear interpolation
      'spline'   - piecewise cubic spline interpolation (SPLINE)
      'pchip'    - piecewise cubic Hermite interpolation (PCHIP)
      'cubic'    - same as 'pchip'
      'v5cubic'  - the cubic interpolation from MATLAB 5,
                   which does no extrapolate and uses 'spline'
                   if X is not equally spaced
```

"Nearest neighbour" interpolation refers to interpolation that is based on just adjacent samples to fill in a new sample. Linear interpolation uses the interpolation filter described in the class.

Lowpass Interpolation

The second interpolation function available in MATLAB is the `interp.m` function that implements Sinc–interpolation. The synopsis of this function is given below.

```
INTERP Resample data at a higher rate using lowpass interpolation.
    Y = INTERP(X,R) resamples the sequence in vector X at R times
    the original sample rate.  The resulting resampled vector Y is
    R times longer, LENGTH(Y) = R*LENGTH(X).
```

```
    A symmetric filter, B, allows the original data to pass through
    unchanged and interpolates between so that the mean square error
    between them and their ideal values is minimized.
    Y = INTERP(X,R,L,ALPHA) allows specification of arguments
    L and ALPHA which otherwise default to 4 and .5 respectively.
    2*L is the number of original sample values used to perform the
    interpolation.  Ideally L should be less than or equal to 10.
    The length of B is 2*L*R+1. The signal is assumed to be band
    limited with cutoff frequency 0 < ALPHA <= 1.0.
    [Y,B] = INTERP(X,R,L,ALPHA) returns the coefficients of the
    interpolation filter B.
```

Decimation

Decimation or rate reduction is accomplished by using the function `decimate.m`. The synopsis of this function is given below.

```
DECIMATE Resample data at a lower rate after lowpass filtering.
    Y = DECIMATE(X,R) resamples the sequence in vector X
    at 1/R times the original sample rate.
    The resulting resampled vector Y is R times shorter,
    LENGTH(Y) = LENGTH(X)/R.
    DECIMATE filters the data with an eighth order Tchebychev
    Type I lowpass filter with cutoff
    frequency .8*(Fs/2)/R, before resampling.
    Y = DECIMATE(X,R,N) uses an N'th order Chebyshev filter.
    Y = DECIMATE(X,R,'FIR') uses the 30 point FIR filter generated by
    FIR1(30,1/R) to filter the data.
    Y = DECIMATE(X,R,N,'FIR') uses the N-point FIR filter.
```