

Remove

The algorithm is based upon the McClellan-Parks-Rabiner Fortran program as found in the IEEE and Elliot references cited below.

If the filter converged, Remez sets V_Flag to 0. Otherwise it sets it to the number of iterations before it failed.

Example

This example specifies a length 41 lowpass filter with passband 0 to $0.14 \times fs$ and stopband $0.18 \times fs$ to $0.5 \times fs$. The passband weight is equal to the stopband weight.

```
Make/O/N=41 coefs = NaN
Make/O/N=(41*16) fr, wt, grid
grid = 0.5*p/numpts(grid)           // Frequencies where fr and wt define desire response
wt = 1
fr[0,0.14*numpts(fr)] = 1         // Low pass from 0 to 0.14 x fs

// Remove transition frequencies
DeletePoints 0.14*numpts(fr), 0.04*numpts(fr), fr, wt, grid

// Compute filter coefs
Remez fr, wt, grid, coefs

// Analyze the filter's frequency response
FFT/OUT=3/PAD={256}/DEST=coefs_FFT coefs

// Display filter response for 1Hz sample rate
Display coefs_FFT
```

References

J. H. McClellan, T.W. Parks, and L. R. Rabiner, *A computer program for designing optimum FIR linear phase digital filters*. IEEE Transactions on Audio and Electroacoustics, AU-21, 506-526 (December 1973).

L. R. Rabiner, J. H. McClellan, and T.W. Parks, *FIR digital filter design techniques using weighted Chebyshev approximation*, Proc. IEEE 63, 595-610 (April 1975)

Elliot, Douglas F., contributing editor, *Handbook of Digital Signal Processing Engineering Applications*, Academic Press, San Diego, CA, 1987.

IEEE Digital Signal Processing Committee, Editor, *Programs for Digital Signal Processing*, IEEE Press, New York, 1979.

See Also

[FMaxFlat](#), [FilterFIR](#)

Remove

Remove

When interpreting a command, Igor treats the Remove operation as [RemoveFromGraph](#), [RemoveFromTable](#), or [See Also](#), depending on the target window. This does not work when executing a user-defined function. Therefore, we recommend that you use [RemoveFromGraph](#), [RemoveFromTable](#), or [RemoveLayoutObjects](#) rather than Remove.

RemoveByKey

RemoveByKey(keyStr, kwListStr [, keySepStr [, listSepStr [, matchCase]]])

The RemoveByKey function returns *kwListStr* after removing the keyword-value pair specified by *keyStr*. *kwListStr* should contain keyword-value pairs such as "KEY=value1,KEY2=value2" or "Key:value1;KEY2:value2", depending on the values for *keySepStr* and *listSepStr*.

Use RemoveByKey to remove information from a string containing a "key1:value1;key2:value2;" or "key1=value1,key2=value2," style list such as those returned by functions like [AxisInfo](#) or [TraceInfo](#).

If *keyStr* is not found then *kwListStr* is returned unchanged.

keySepStr, *listSepStr*, and *matchCase* are optional; their defaults are ":";, ";", and 0 respectively.

Details

kwListStr is searched for an instance of the key string bound by *listSepStr* on the left and a *keySepStr* on the right. The key, the *keySepStr*, and the text up to and including the next *listSepStr* (if any) are removed from the returned string.

If the resulting string contains only *listSepStr* characters, then an empty string ("") is returned.