API Collection

Finite impulse response filters

Perform finite impulse response filtering with decimation and antialiasing on vectors of real or complex values.

Topics

FIR Filter Creation

`vDSP_wiener`

Solves a system of linear equations for a single-precision symmetric Toeplitz coefficient matrix.

`vDSP_wienerD`

Solves a system of linear equations for a double-precision symmetric Toeplitz coefficient matrix.

Real Vectors

📄 Resampling a signal with decimation

Reduce the sample rate of a signal by specifying a decimation factor and applying a custom antialiasing filter.

```
static func downsample<T, U>(U, decimationFactor: Int, filter: T) -> [
Double]
```

Returns the downsampled double-precision vector.

```
static func downsample<T, U>(U, decimationFactor: Int, filter: T) -> [
Float]
```

Returns the downsampled single-precision vector.

```
static func downsample<T, U, V>(U, decimationFactor: Int, filter: T,
result: inout V)
```

Calculates the downsampled double-precision vector.

```
static func downsample<T, U, V>(U, decimationFactor: Int, filter: T,
result: inout V)
```

Calculates the downsampled single-precision vector.

`vDSP_desamp`

Performs single-precision FIR filtering with decimation and antialiasing.

`vDSP_desampD`

Performs double-precision FIR filtering with decimation and antialiasing.

Complex Vectors

`vDSP_zrdesamp`

Performs complex-real single-precision FIR filtering with decimation and antialiasing.

`vDSP_zrdesampD`

Performs complex-real double-precision FIR filtering with decimation and antialiasing.

See Also

Vector filtering

≔ Biquadratic IIR filters

Apply biquadratic filters to single-channel and multichannel data.

≔ Single-channel biquadratic filters

Filter a single-channel signal with a cascade of biquadratic sections.

≔ Multichannel biquadratic filters

Filter a multichannel signal with a cascade of biquadratic sections.

≔ Recursive filters

Perform two-pole two-zero recursive filtering on a vector.