awgn

Add white Gaussian noise to signal

Syntax

```
y = awgn(x,snr)
y = awgn(x,snr,signalpower)

y = awgn(x,snr,signalpower,randobject)
y = awgn(x,snr,signalpower,seed)
y = awgn( ___ ,powertype)

[y,var] = awgn( ___ )
```

Description

`y = awgn(x,snr)` adds white Gaussian noise to the vector signal `x`. This syntax assumes that the power of `x` is 0 dBW. For more information about additive white Gaussian noise, see What is AWGN?

`y = awgn(x,snr,signalpower)` accepts an input signal power value in dBW. To measure the power of `x` before adding noise, specify `signalpower` as `'measured'`. The `'measured'` option does not generate the requested average SNR for repeated `awgn` function calls in a loop if the input signal power varies over time due to fading and the coherence time of the channel is larger than the input duration. example

`y = awgn(x,snr,signalpower,randobject)` additionally accepts a random number stream object to generate normal random noise samples. For information about producing repeatable noise samples, see Tips. example

`y = awgn(x,snr,signalpower,seed)` specifies a seed value for initializing the normal random number generator that is used to add white Gaussian noise to the input signal. example

`y = awgn( ___ ,powertype)` specifies the signal and noise power type as `'dB'` or `'linear'` in addition to the input arguments in any of the previous syntaxes. For information on the relationships between SNR and other measures of the relative power of the noise, such as E_s/N_0, and E_b/N_0, see AWGN Channel Noise Level.

`[y,var] = awgn( ___ )` also returns the total noise variance used to produce random noise samples.

Examples

collapse all

Add AWGN to Sawtooth Signal

Create a sawtooth wave.

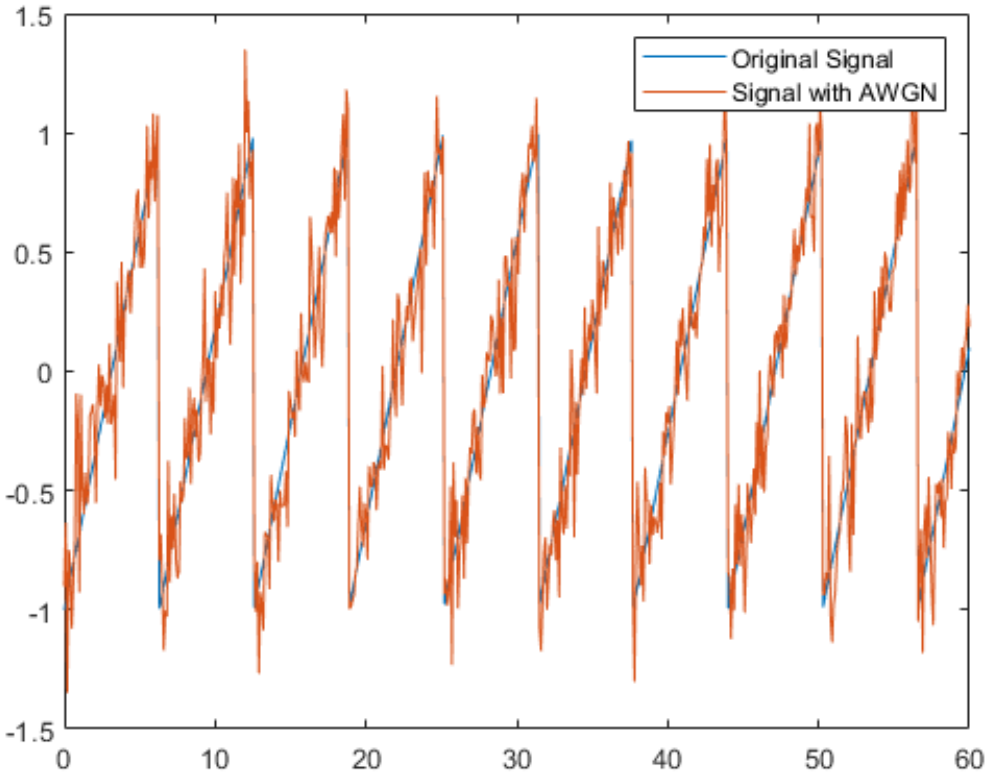
Try This Example

Copy Command

```
t = (0:0.1:60)';
x = sawtooth(t);
```

Add white Gaussian noise and plot the results.

```
y = awgn(x,10,'measured');
plot(t,[x y])
legend('Original Signal','Signal with AWGN')
```



General QAM Modulation over AWGN Channel

Transmit and receive data using a nonrectangular 16-ary constellation in the presence of Gaussian noise. Show the scatter plot of the noisy constellation and estimate the symbol error rate (SER) for two different SNRs.

Try This Example

Copy Command

Create a 16-QAM constellation based on the V.29 standard for telephone-line modems.

```
c = [-5 -5i 5 5i -3 -3-3i -3i 3-3i 3 3+3i 3i -3+3i -1 -1i 1 1i];
sigpower = pow2db(mean(abs(c).^2));
M = length(c);
```

Generate random symbols.

```
data = randi([0 M-1],2000,1);
```

Modulate the data by using the `genqammod` function. General QAM modulation is necessary because the custom constellation is not rectangular.

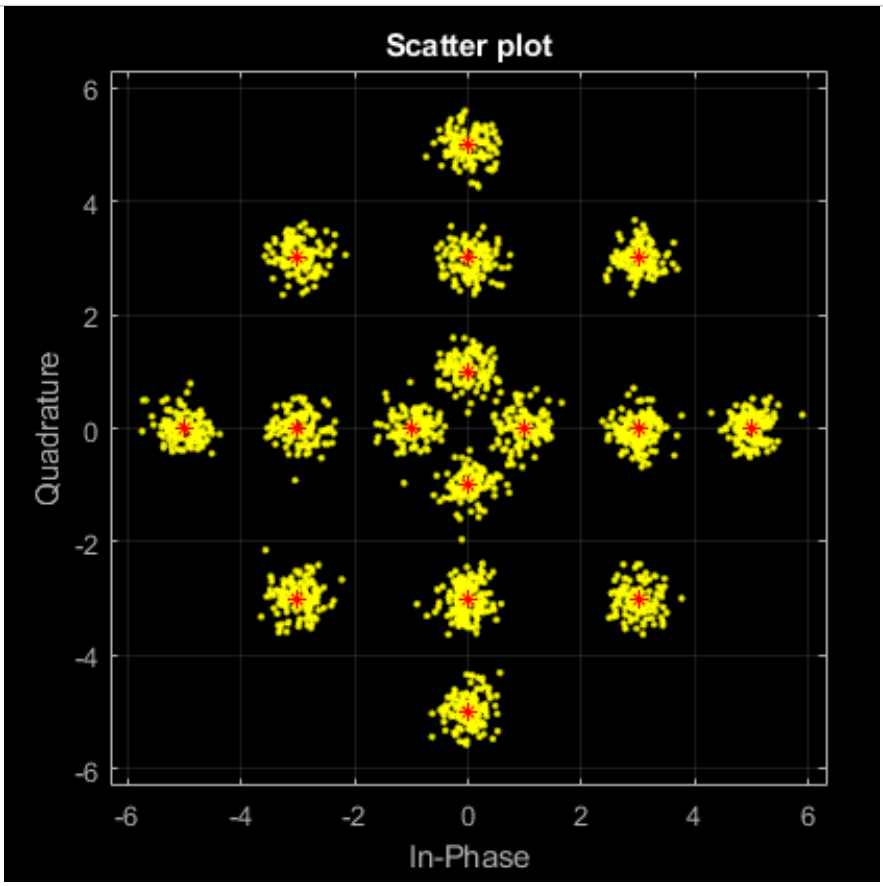
```
modData = genqammod(data,c);
```

Pass the signal through an AWGN channel with a 20 dB SNR.

```
rxSig = awgn(modData,20,sigpower);
```

Display a scatter plot of the received signal and the reference constellation `c`.

```
h = scatterplot(rxSig);
hold on
scatterplot(c,[],[],'r*',h)
grid
hold off
```



Demodulate the received signal by using the `genqamdemod` function. Determine the number of symbol errors and the SER.

```
demodData = genqamdemod(rxSig,c);
[numErrors,ser] = symerr(data,demodData)
```

```
numErrors = 1
ser = 5.0000e-04
```

Repeat the transmission and demodulation process with an AWGN channel with a 10 dB SNR. Determine the SER for the reduced SNR. As expected, the performance degrades when the SNR is decreased.

```
rxSig = awgn(modData,10,sigpower);
demodData = genqamdemod(rxSig,c);
[numErrors,ser] = symerr(data,demodData)
```

```
numErrors = 461
```

```
ser = 0.2305
```

Repeatable AWGN Simulation

Generate random data symbols and the 4-PSK modulated signal.

Try This Example

Copy Command

```
M = 4;
k = log2(M);
snr = 3;
data = randi([0 M-1],2000,1);
x = pskmod(data,M);
```

Set the random number generator seed.

```
seed = 12345;
```

Generate repeatable random noise using the `rng` function before calling the `awgn` function.

```
rng(seed);
y = awgn(x,snr);
```

Compute the bit errors.

```
dataHat = pskdemod(y,M);
numErr1 = biterr(data,dataHat,k)
```

```
numErr1 = 321
```

Reset the random number generator seed.

```
rng(seed);
```

Demodulate the PSK signal and compute the bit errors.

```
y = awgn(x,snr);
dataHat = pskdemod(y,M);
numErr2 = biterr(data,dataHat,k)
```

```
numErr2 = 321
```

Compare `numErr1` to `numErr2`. The errors are equal even after you reset the random number generator seed.

```
isequal(numErr1, numErr2)
```

```
ans = logical
   1
```

Repeatable AWGN with RandStream

Generate white Gaussian noise addition results using a `RandStream` object and the `reset` object function.

Specify the power of X as `0` dBW, add noise to produce an SNR of `10` dB, and use a local random stream.

Try This Example

Copy Command

```
S = RandStream('mt19937ar','Seed',5489);
sigin = sqrt(2)*sin(0:pi/8:6*pi);
sigout1 = awgn(sigin,10,0,S);
```

Add white Gaussian noise to `sigin`. Use `isequal` to compare `sigout1` to `sigout2`. The outputs are not equal when you do not reset the random stream.

```
sigout2 = awgn(sigin,10,0,S);
isequal(sigout1,sigout2)
```

```
ans = logical
   0
```

Reset the random stream object, returning the object to its state prior to adding AWGN to `sigout1`. Add AWGN to `sigin` and compare `sigout1` to `sigout3`. The outputs are equal when you reset the random stream.

```
reset(S);
sigout3 = awgn(sigin,10,0,S);
isequal(sigout1,sigout3)
```

```
ans = logical
   1
```

Input Arguments

collapse all

x — Input signal

scalar | vector | array

Input signal, specified as a scalar, vector, or array. The power of the input signal is assumed to be 0 dBW.

Data Types: `double`
Complex Number Support: Yes

snr — Signal-to-noise ratio

scalar

Signal-to-noise ratio in dB, specified as a scalar. The function applies the same `snr` value to each channel. The columns of the input signal represent the different channels of a multichannel signal.

Data Types: `double`

`signalpower` — Signal power

scalar | `'measured'`

Signal power, specified as a scalar or `'measured'`.

- Scalar — The value is used as the signal level of `in` to determine the appropriate noise level based on the value of `snr`.
- `'measured'` — The signal level of `in` is computed to determine the appropriate noise level based on the value of `snr`.

If the input signal is a multichannel signal, the function calculates the `signalpower` value across all channels as a single value. It then uses the value to calculate the noise level for all the channels.

Data Types: `double`

`randobject` — Random number stream object

`RandStream` object

Random number stream object, specified as a `RandStream` object. The state of the random stream object determines the sequence of numbers produced by the `randn` function. Configure the random stream object using the `reset (RandStream)` function and its properties.

For information about producing repeatable noise samples, see Tips.

`seed` — Random number generator seed

scalar

Random number generator seed value, specified as a scalar.

Data Types: `double`

`powertype` — Signal power unit

`'dB'` (default) | `'linear'`

Signal power unit, specified as `'dB'` or `'linear'`.

- When `powertype` is `'dB'`, `snr` is measured in dB and `signalpower` is measured in dBW.
- When `powertype` is `'linear'`, the `snr` is measured as a ratio and `signalpower` is measured in watts assuming a reference load of 1 ohms.

To set the `powertype` argument, you must also set `snr` and `signalpower`.

Output Arguments

collapse all

y — Output signal

scalar | vector | array

Output signal, returned as a scalar, vector, or array. The returned output signal is the input signal with added white Gaussian noise.

var — Noise variance

scalar

Total noise variance, returned as a positive scalar. The function uses the noise variance to generate random noise samples.

More About

collapse all

What is AWGN?

Additive white Gaussian noise (AWGN) is a simple noise model that represents electron motion in the RF front end of a receiver. As the name implies, the noise gets added to the signal. The noise is called "white" because it is spectrally flat across the entire sampling bandwidth. Analogously, the color white contains equal spectral power levels at all frequencies of the visible light spectrum. The noise is Gaussian because its amplitude can be modeled with a normal probability distribution.

The AWGN channel is often used to model a satellite communications channel, since that channel typically does not suffer from common terrestrial impairments like fading, multipath, and interference. An AGWN channel serves as a good starting point for the analysis of terrestrial wireless links because it establishes a best-case bound on the bit error rate performance of a terrestrial link.

Tips

- For information on the relationships between SNR and other measures of the relative power of the noise, such as E_s/N_0, and E_b/N_0, see AWGN Channel Noise Level.
- To generate repeatable white Gaussian noise samples, do one of the following:
  - Use `rng(seed)` before calling the `awgn` function to generate repeatable random noise.
  - Provide a static `seed` value as an input to `awgn`.
  - Use the `reset (RandStream)` function on the `randobject` before passing it as an input to `awgn`.
  - Provide `randobject` in a known state as an input to `awgn`. For more information, see `RandStream`.

Extended Capabilities

C/C++ Code Generation

Generate C and C++ code using MATLAB® Coder™.

Version History

Introduced before R2006a

See Also

Functions

`convertSNR` | `wgn` | `randn` | `bsc` | `RandStream`

Objects

`comm.AWGNChannel`

Topics

AWGN Channel Noise Level