

Signal Detection in White Gaussian Noise

This example discusses the detection of a deterministic signal in complex, white, Gaussian noise. This situation is frequently encountered in radar, sonar and communication applications.

Try This Example▼

Overview

There are many different kinds of detectors available for use in different applications. A few of the most popular ones are the Bayesian detector, maximum likelihood (ML) detector and Neyman-Pearson (NP) detector. In radar and sonar applications, NP is the most popular choice since it can ensure the probability of false alarm (P_{fa}) to be at a certain level.

In this example, we limit our discussion to the scenario where the signal is deterministic and the noise is white and Gaussian distributed. Both signal and noise are complex.

The example discusses the following topics and their interrelations: coherent detection, noncoherent detection, matched filtering and receiver operating characteristic (ROC) curves.

Signal and Noise Model

The received signal is assumed to follow the model

$$x(t) = s(t) + n(t)$$

where $s(t)$ is the signal and $n(t)$ is the noise. Without losing the generality, we assume that the signal power is equal to 1 watt and the noise power is determined accordingly based on the signal to noise ratio (SNR). For example, for an SNR of 10 dB, the noise power, i.e., noise variance will be 0.1 watt.

Matched Filter

A matched filter is often used at the receiver front end to enhance the SNR. From the discrete signal point of view, matched filter coefficients are simply given by the complex conjugated reversed signal samples.

When dealing with complex signals and noises, there are two types of receivers. The first kind is a coherent receiver, which assumes that both the amplitude and

phase of the received signal are known. This results in a perfect match between the matched filter coefficients and the signal s . Therefore, the matched filter coefficients can be considered as the conjugate of s . The matched filter operation can then be modeled as

$$y = s^*x = s^*(s + n) = |s|^2 + s^*n .$$

Note that although the general output y is still a complex quantity, the signal is completely characterized by $|s|^2$, which is a real number and contained in the real part of y . Hence, the detector following the matched filter in a coherent receiver normally uses only the real part of the received signal. Such a receiver can normally provide the best performance. However, the coherent receiver is vulnerable to phase errors. In addition, a coherent receiver also requires additional hardware to perform the phase detection. For a noncoherent receiver, the received signal is modeled as a copy of the original signal with a random phase error. With a noncoherent received signal, the detection after the matched filter is normally based on the power or magnitude of the signal since you need both real and imaginary parts to completely define the signal.

Detector

The objective function of the NP decision rule can be written as

$$J = P_d + g(P_{fa} - a),$$

i.e., to maximize the probability of the detection, P_d , while limiting the probability of false alarm, P_{fa} at a specified level a . The variable g in the equation is the Lagrange multiplier. The NP detector can be formed as a likelihood ratio test (LRT) as follows:

$$\frac{p_y(y|H_1)}{p_y(y|H_0)} >_{H_1} Th .$$

In this particular NP situation, since the false alarm is caused by the noise alone, the threshold Th is determined by the noise to ensure the fixed P_{fa} . The general form of the LRT shown above is often difficult to evaluate. In real applications, we often use an easy to compute quantity from the signal, i.e., sufficient statistic, to replace the ratio of two probability density functions. For example, the sufficient statistics, z may be as simple as

$$z = |y| ,$$

then the simplified detector becomes

$$z >_{H_1} T .$$

$$<_{H_0}$$

T is the threshold to the sufficient statistic z , acting just like the threshold Th to the LRT. Therefore, the threshold is not only related to the probability distributions, but also depends on the choice of sufficient statistic.

Single Sample Detection Using Coherent Receiver

We will first explore an example of detecting a signal in noise using just one sample.

Assume the signal is a unit power sample and the SNR is 3 dB. Using a 100000-trial Monte-Carlo simulation, we generate the signal and noise as

```
% fix the random number generator
rstream = RandStream.create('mt19937ar','seed',2009);

Ntrial = 1e5;           % number of Monte-Carlo trials
snrdb = 3;              % SNR in dB
snr = db2pow(snrdb);    % SNR in linear scale
spower = 1;            % signal power is 1
npower = spower/snr;    % noise power
namp = sqrt(npower/2);  % noise amplitude in each channel
s = ones(1,Ntrial);     % signal
n = namp*(randn(rstream,1,Ntrial)+1i*randn(rstream,1,Ntrial)); % noise
```

Note that the noise is complex, white and Gaussian distributed.

If the received signal contains the target, it is given by

```
x = s + n;
```

The matched filter in this case is trivial, since the signal itself is a unit sample.

```
mf = 1;
```

In this case, the matched filter gain is 1, therefore, there is no SNR gain.

Now we do the detection and examine the performance of the detector. For a coherent receiver, the received signal after the matched filter is given by

```
y = mf'*x; % apply the matched filter
```

The sufficient statistic, i.e., the value used to compare to the detection threshold, for a coherent detector is the real part of the received signal after the matched filter, i.e.,

```
z = real(y);
```

Let's assume that we want to fix the P_{fa} to $1e-3$. Given the sufficient statistic, z , the decision rule becomes

$$\begin{array}{c} H_1 \\ z > T \\ H_0 \\ z < T \end{array}$$

where the threshold T is related to P_{fa} as

$$P_{fa} = \frac{1}{2} \left[1 - \operatorname{erf} \left(\frac{T}{\sqrt{NM}} \right) \right]$$

In the equation, N is the signal power and M is the matched filter gain. Note that T is the threshold of the signal after the matched filter and NM represents the noise power after the matched filter, so $\frac{T}{\sqrt{NM}}$ can be considered as the ratio between the signal and noise magnitude, i.e., it is related to the signal to noise ratio, SNR.

Since SNR is normally referred to as the ratio between the signal and noise power, considering the units of each quantity in this expression, we can see that

$$\frac{T}{\sqrt{NM}} = \sqrt{\text{SNR}}.$$

Since N and M are fixed once the noise and signal waveform are chosen, there is a correspondence between T and SNR. Given T is a threshold of the signal, SNR can be considered as a threshold of the signal to noise ratio. Therefore, the threshold equation can then be rewritten in the form of

$$P_{fa} = \frac{1}{2} [1 - \operatorname{erf}(\sqrt{\text{SNR}})].$$

The required SNR threshold given a complex, white Gaussian noise for the NP detector can be calculated using the `npwgnthresh` function as follows:

```
Pfa = 1e-3;
snrthreshold = db2pow(npwgntthresh(Pfa, 1, 'coherent'));
```

Note that this threshold, although also in the form of an SNR value, is different to the SNR of the received signal. The threshold SNR is a calculated value based on the desired detection performance, in this case the Pfa ; while the received signal SNR is the physical characteristic of the signal determined by the propagation environment, the waveform, the transmit power, etc.

The true threshold T can then be derived from this SNR threshold as

$$T = \sqrt{NM} \cdot \sqrt{\text{SNR}}.$$

```
mfgain = mf'*mf;
% To match the equation in the text above
% npower - N
% mfgain - M
% snrthreshold - SNR
threshold = sqrt(npower*mfgain*snrthreshold);
```

The detection is performed by comparing the signal to the threshold. Since the original signal, s , is presented in the received signal, a successful detection occurs when the received signal passes the threshold, i.e. $z > T$. The capability of the detector to detect a target is often measured by the Pd . In a Monte-Carlo simulation, Pd can be calculated as the ratio between the number of times the signal passes the threshold and the number of total trials.

```
Pd = sum(z>threshold)/Ntrial
```

```
Pd = 0.1390
```

On the other hand, a false alarm occurs when the detection shows that there is a target but there actually isn't one, i.e., the received signal passes the threshold when there is only noise present. The error probability of the detector to detect a target when there isn't one is given by Pfa .

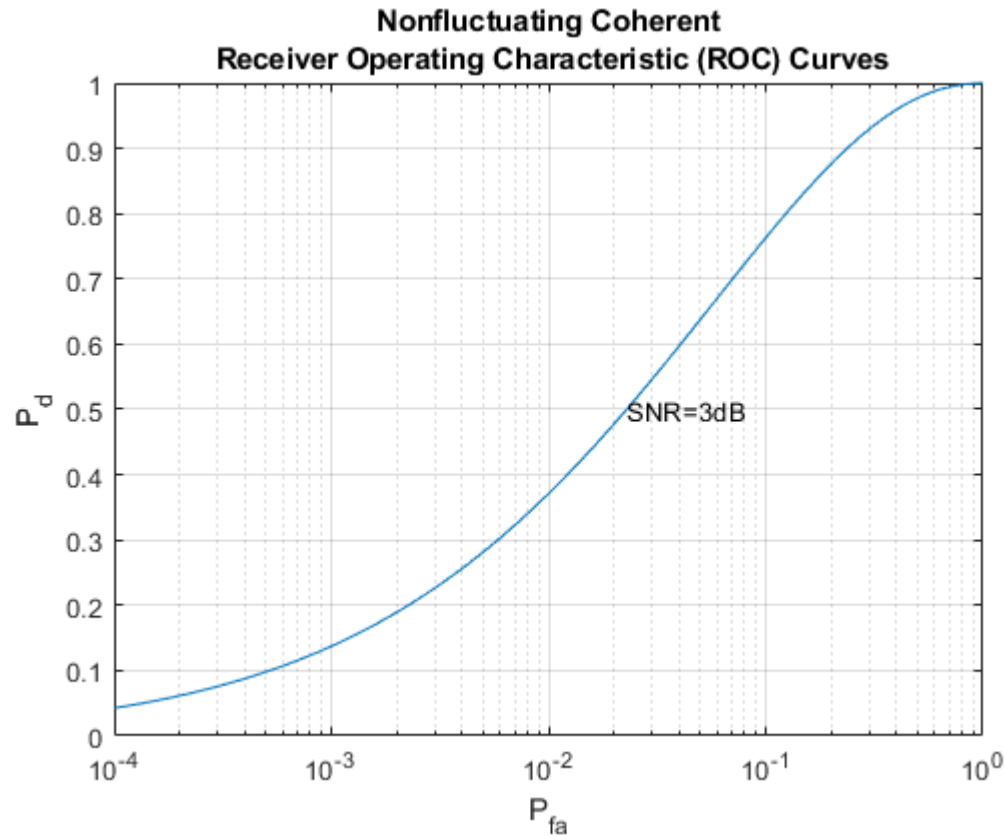
```
x = n;
y = mf'*x;
z = real(y);
Pfa = sum(z>threshold)/Ntrial
```

```
Pfa = 9.0000e-04
```

which meets our requirement.

To see the relation among SNR, P_d and P_{fa} in a graph, we can plot the theoretical ROC curve using the `rocsnr` function for a SNR value of 3 dB as

```
rocsnr(snrdb, 'SignalType', 'NonfluctuatingCoherent', 'MinPfa', 1e-4);
```



It can be seen from the figure that the measured $P_d=0.1390$ and $P_{fa}=0.0009$ obtained above for the SNR value of 3 dB match a theoretical point on the ROC curve.

Single Sample Detection Using Noncoherent Receiver

A noncoherent receiver does not know the phase of the received signal, therefore, for target present case, the signal x contains a phase term and is defined as

```
% simulate the signal
x = s.*exp(1i*2*pi*rand(rstream,1,Ntrial)) + n;
y = mf'*x;
```

When the noncoherent receiver is used, the quantity used to compare with the threshold is the power (or magnitude) of the received signal after the matched filter. In this simulation, we choose the magnitude as the sufficient statistic.

```
z = abs(y);
```

Given our choice of the sufficient statistic z , the threshold is related to P_{fa} by the equation

$$P_{fa} = \exp\left(-\frac{T^2}{NM}\right) = \exp(-\text{SNR}) .$$

The signal to noise ratio threshold SNR for an NP detector can be calculated using `npwgnthresh` as follows:

```
snrthreshold = db2pow(npwgnthresh(Pfa, 1, 'noncoherent'));
```

The threshold, T , is derived from SNR as before

```
mfgain = mf'*mf;
threshold = sqrt(npower*mfgain*snrthreshold);
```

Again, P_d can then be obtained using

```
Pd = sum(z>threshold)/Ntrial
```

```
Pd = 0.0583
```

Note that this resulting P_d is inferior to the performance we get from a coherent receiver.

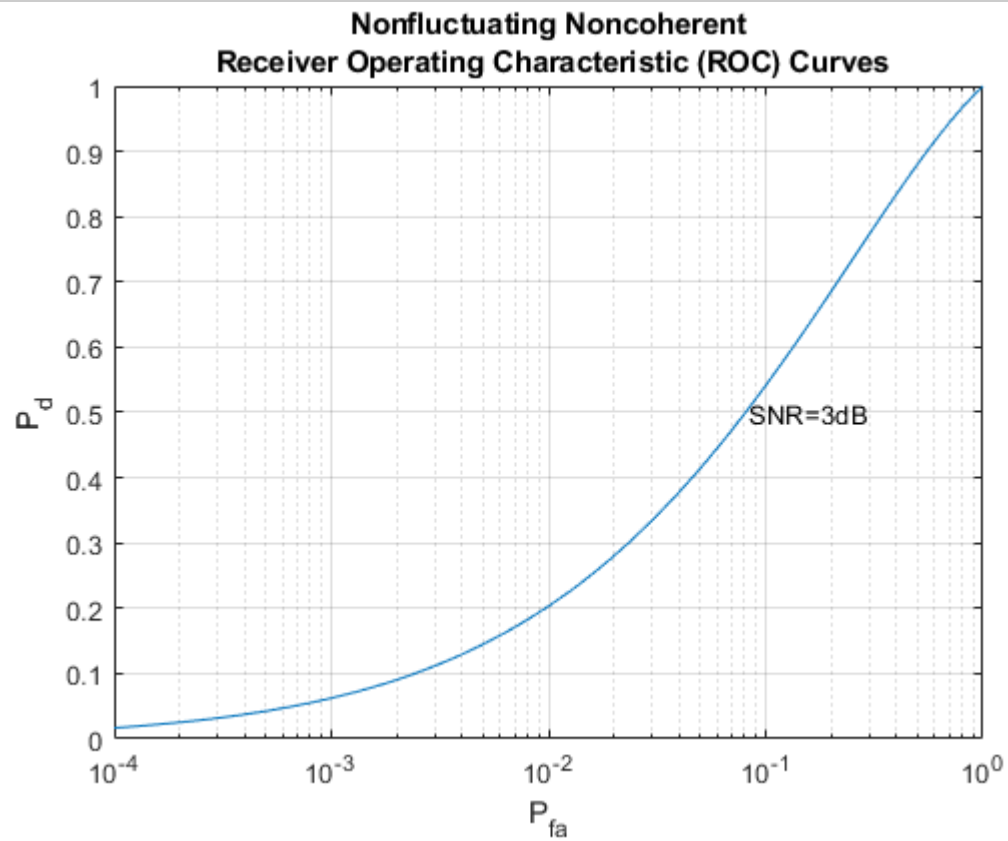
For the target absent case, the received signal contains only noise. We can calculate the P_{fa} using Monte-Carlo simulation as

```
x = n;  
y = mf'*x;  
z = abs(y);  
Pfa = sum(z>threshold)/Ntrial
```

Pfa = 9.5000e-04

The ROC curve for a noncoherent receiver is plotted as

```
rocsnr(snrdb, 'SignalType', 'NonfluctuatingNoncoherent', 'MinPfa', 1e-4);
```



We can see that the performance of the noncoherent receiver detector is inferior to that of the coherent receiver.

Summary

This example shows how to simulate and perform different detection techniques using MATLAB®. The example illustrates the relationship among several frequently encountered variables in signal detection, namely, probability of detection (P_d), probability of false alarm (P_{fa}) and signal to noise ratio (SNR). In particular, the example calculates the performance of the detector using Monte-Carlo simulations and verifies the results of the metrics with the receiver operating characteristic (ROC) curves.

There are two SNR values we encounter in detecting a signal. The first one is the SNR of a single data sample. This is the SNR value appeared in a ROC curve plot. A point on ROC gives the required single sample SNR necessary to achieve the corresponding P_d and P_{fa} . However, it is NOT the SNR threshold used for detection. Using the Neyman-Pearson decision rule, the SNR threshold, the second SNR value we see in the detection, is determined by the noise distribution and the desired P_{fa} level. Therefore, such an SNR threshold indeed corresponds to the P_{fa} axis in a ROC curve. If we fix the SNR of a single sample, as depicted in the above ROC curve plots, each point on the curve will correspond to a P_{fa} value, which in turn translates to an SNR threshold value. Using this particular SNR threshold to perform the detection will then result in the corresponding P_d .

Note that an SNR threshold may not be the threshold used directly in the actual detector. The actual detector normally uses an easy to compute sufficient statistic quantity to perform the detection. Thus, the true threshold has to be derived from the aforementioned SNR threshold accordingly so that it is consistent with the choice of sufficient statistics.

This example performs the detection using only one received signal sample. Hence, the resulting P_d is fairly low and there is no processing gain achieved by the matched filter. To improve P_d and to take advantage of the processing gain of the matched filter, we can use multiple samples, or even multiple pulses, of the received signal. For more information about how to detect a signal using multiple samples or pulses, please refer to the example [Signal Detection Using Multiple Samples](#).

