Package 'rwirelesscom'

October 18, 2015

Title Basic Wireless Communications Simulation

Version 1.4.3

Maintainer Alberto Gutierrez <algutier1@gmail.com>

Description A communications simulation package supporting common wireless modulation formats including BPSK, QPSK, 8-PSK, 16-PSK, 16-QAM and 64-QAM. The package includes AWGN noise generation, and raised cosine and square root raised cosine pulse shaping. It also includes convenience functions for plotting constellation diagrams, density plots, stem plots and eye diagrams. Together these functions enable the evaluation of bit error and symbol error rates, evaluation of pulse shaping and inter-symbol interference and support visualization of the respective signals and noise in a variety of formats.

Depends R (>= 3.1)

License GPL (>= 2)

LazyData true

Imports ggplot2, stats, Rcpp (>= 0.12.1)

Suggests testthat

LinkingTo Rcpp

NeedsCompilation yes

Author Alberto Gutierrez [aut, cre]

Repository CRAN

Date/Publication 2015-10-18 10:13:25

R topics documented:

`eyediagram` *Eye Diagram*

Description

Receives a vector x of real or complex points and plots the Re or Im part of x in the form of an "eyediagram." The symbol period is indicated by the input Ns (samples per symbol) and horizontal sweep Np indicates the the number of symbol periods to plot along the horizontal access. The eyediagram is useful for evaluating inter-symbol interference associated with the in-phase or quadrature part of a modulated signal.

Usage

```
eyediagram(x, Ns = 1, Np = 3, No = 1, iq = "r", pch = 19, cex = 0.1,
  ...)
```

Arguments

`x`	- vector of real or complex points
`Ns`	- number of samples per symbol period
`Np`	- number of symbol periods to plot along the horizontal axis
`No`	- offset (n points) alignment of the eyediagram along the horizontal asis
`iq`	- parameter indicates whether to plot the in-phase Re(x) (iq="r" default) or quadrature Im(x) (iq="q")
`pch`	- Graphical parameter pch (plotting character) set to 19 by default ("point")
`cex`	- Graphical parameter cex magnificaiton of plotting symbols relative to 1 default set to 0.1.
`...`	- graphical environment parameters are input to eyediagram

See Also

Other rwirelesscom functions: f16pskdemod; f16pskmod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fNo; fbpskdemod; fbpskmod; fqpskdemod; iqdensityplot; iqscatterplot; stemplot

Examples

```
# Step 1: generate random set of bits
M=4
Nsymbols=10000
Nbits=log2(M)*Nsymbols
bits <- sample(0:1,Nbits, replace=TRUE)

# Step 2: Generate a BPSK modulated signal including raised cosine
#  pulse shaping sampled at 64 samples per symbol period and roll-off
#  factor of 0.5.
Ns=64
B=0.5
hx=seq(-4*Ns,4*Ns,by=1)
h=rcosine(hx,B,Ns)
s <- fqpskmod(bits,Ns,h)

# Step 3: Plot the transmitted waveform with the eyediagram function.
#    Remove the initial 1000 and tail points for a cleaner diagram, i.e.
#    without startup and tail artifacts
Np=3
No=27

# real part (in-phase)
eyediagram(s[1000:60000],Ns,Np,No,iq="r",xlab="Ts",ylab="I")
# imaginary part (quadrature)
eyediagram(s[1000:60000],Ns,Np,No,iq="q",xlab="Ts",ylab="Q")
```

---

f16pskdemod *16-PSK Demodulator*

Description

Receives a vector of complex values, r, corresponding to an 8-PSK modulated signal transmitted through a communications channel (e.g., signal plus noise). Three bits are output for each received symbol according to the following decision rules

input	output
$-\pi/16 \geq Arg(r) < \pi/16$	0000
$\pi/16 \geq Arg(r) < 3\pi/16$	0001
$3\pi/16 \geq Arg(r) < 5\pi/16$	0011
$5\pi/16 \geq Arg(r) < 7\pi/16$	0010
$7\pi/16 \geq Arg(r) < 9\pi/16$	0110
$9\pi/16 \geq Arg(r) < 11\pi/16$	0111

$$
\begin{array}{rl}
11\pi/16 \geq Arg(r) < 13\pi/16 & 0101 \\
13\pi/16 \geq Arg(r) < 15\pi/16 & 0100 \\
15\pi/16 \geq Arg(r) < 17\pi/16 & 1100 \\
-15\pi/16 \geq Arg(r) < -13\pi/16 & 1101 \\
-13\pi/16 \geq Arg(r) < -11\pi/16 & 1111 \\
-11\pi/16 \geq Arg(r) < -9\pi/16 & 1110 \\
-9\pi/16 \geq Arg(r) < -7\pi/16 & 1010 \\
-7\pi/16 \geq Arg(r) < -5\pi/16 & 1011 \\
-5\pi/16 \geq Arg(r) < -3\pi/16 & 1001 \\
-3\pi/16 \geq Arg(r) < -\pi/16 & 1000
\end{array}
$$

Usage

```
f16pskdemod(r)
```

Arguments

r	- received signal

Value

returns a vector of 1's and 0's, 3 bits per input element (i.e., 8-PSK symbol)

See Also

Other rwirelesscom functions: eyediagram; f16pskmod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fNo; fbpskdemod; fbpskmod; fqpskdemod; iqdensityplot; iqscatterplot; stemplot

Examples

```
M=16
Es=1
Eb = Es/log2(M)
Nsymbols=20
Nbits=log2(M)*Nsymbols
bits <- sample(0:1,Nbits, replace=TRUE)
s <- f16pskmod(bits)
EbNodB=7
No = Eb/(10^(EbNodB/10))
n <- fNo(Nsymbols,No,type="complex")
r <- s+n
bitsr <- f16pskdemod(r)
biterrs<-bits[bitsr!=bits]
b<-factor(bits)
Pberr=length(biterrs)/length(bits)
```

f16pskmod *16-PSK Modulator*

Description

Receives a vector of bits (1's and 0's). The received vector is mapped to in-phase (real) and quadrature (imaginary) components, according to a Binary Reflective Gray Code (BRGC, see reference). Each received pair of bits are are mapped to 16-PSK symbols, with Es (symbol energy) = 1.The bit to symbol mapping is illustrated in the following table.

input	output
0000	0
0001	$\pi/8$
0011	$\pi/4$
0010	$3\pi/8$
0110	$\pi/2$
0111	$5\pi/8$
0101	$3\pi/4$
0100	$7\pi/8$
1100	-1
1101	$-7\pi/8$
1111	$-3\pi/4$
1110	$-5\pi/8$
1010	$-\pi/2$
1011	$-3\pi/8$
1001	$-\pi/4$
1000	$-\pi/8$

Reference: E. Agrell, J Lassing, E. Strom, and T. Ottosson, Gray Coding for Multilevel Constellations In Gaussian Noise, IEEE Transactions on Communications, Vol. 53, No. 1, January 2007

Usage

```
f16pskmod(bits, Ns = 1, p = 1)
```

Arguments

bits	- vector of bits (0's and 1's).
Ns	- N samples per symbol (default, Ns = 1)
p	- a vector defining the pulse shape of the transmitted waveform (default, p = 1)

Value

Returns a complex vector of length of 8-PSK symbols. If Ns > 1 then the returned signal is shaped with pulse shape, p.

See Also

Other rwirelesscom functions: eyediagram; f16pskdemod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fNo; fbpskdemod; fbpskmod; fqpskdemod; iqdensityplot; iqscatterplot; stemplot

Examples

```
M=16
Nsymbols=20
Nbits=log2(M)*Nsymbols
bits <- sample(0:1,Nbits, replace=TRUE)
s <- f16pskmod(bits)
```

---

f16qamdemod *16 QAM Demodulator*

Description

Receives a vector of complex values, r, corresponding to a 16-QAM modulated signal transmitted through a communications channel (e.g., signal plus noise). Each received 16-QAM symbol (one symbol per r sample) is decoded into 4 bits. The 16-QAM symbol decision regions are defined with respect to the constellation generated by f16qammod(), where in-phase and quadrature constellation points take on values -3, -1, +1, +3, respectively.

Usage

```
f16qamdemod(r)
```

Arguments

r	- vector of complex vector

Value

a vector of 1's and 0's, 4 bits per input element (16-QAM symbol)

See Also

Other rwirelesscom functions: eyediagram; f16pskdemod; f16pskmod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fNo; fbpskdemod; fbpskmod; fqpskdemod; iqdensityplot; iqscatterplot; stemplot

Examples

```
M=16
Es=10
Eb = Es/log2(M)
Nsymbols=100
Nbits=log2(M)*Nsymbols
bits <- sample(0:1,Nbits, replace=TRUE)
s <- f16qammod(bits)
EbNodB=8
No = Eb/(10^(EbNodB/10))
n <- fNo(Nsymbols,No,type="complex")
r <- s+n
bitsr <- f16qamdemod(r)
biterrs<-bits[bitsr!=bits]
Pberr=length(biterrs)/length(bits)
```

`f16qammod` *16-QAM Modulator*

Description

Receives a vector of bits (1's and 0's). The received vector is mapped to an in-phase (real) and quadrature (imaginary) 16-QAM (4 bit) symbol according to a a Binary Reflective Gray Code (BRGC, see reference). Each symbol has an average symbol energy Es = 10, where in-phase and quadrature constellation points take on values -3, -1, +1, +3, respectively. The bit to symbol mapping is illustrated in the following constellation diagram.

-3+3i (1000)	-1+3i (1001)	+1+3i (1011)	+3+3i (1010)
-3+1i (1100)	-1+1i (1101)	+1+1i (1111)	+3+1i (1110)
-3-1i (0100)	-1-1i (0101)	+1-1i (0111)	+3-1i (0110)
-3-3i (0000)	-1-3i (0001)	+1-3i (0011)	+3-3i (0010)

Reference: E. Agrell, J Lassing, E. Strom, and T. Ottosson, Gray Coding for Multilevel Constellations In Gaussian Noise, IEEE Transactions on Communications, Vol. 53, No. 1, January 2007

Usage

```
f16qammod(bits, Ns = 1, p = 1)
```

Arguments

bits	- received vector of bits (0's and 1's).
Ns	- N samples per symbol (default, Ns = 1)
p	- a vector defining the pulse shape of the transmitted waveform (default, p = 1)

Value

Returns a complex vector of 16-QAM symbols. If Ns > 1 then the returned signal is shaped with pulse shape, p.

See Also

Other rwirelesscom functions: eyediagram; f16pskdemod; f16pskmod; f16qamdemod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fNo; fbpskdemod; fbpskmod; fqpskdemod; iqdensityplot; iqscatterplot; stemplot

Examples

```
M=16
Nsymbols=100
Nbits=log2(M)*Nsymbols
bits <- sample(0:1,Nbits, replace=TRUE)
s <- f16qammod(bits)
```

---

f64qamdemod *64-QAM Demodulator*

Description

Receives a vector of complex values, r, corresponding to a 64-QAM modulated signal transmitted through a communications channel (e.g., signal plus noise). Each received 64-QAM symbol (one symbol per r sample) is decoded into 6 bits. The 64-QAM symbol decision regions are defined with respect to the constellation generated by f64qammod(), where in-phase and quadrature constellation points take on values -7, -5, -3, -1, +1, +3, +5, +7, respectively.

Usage

```
f64qamdemod(r)
```

Arguments

r	- complex valued input vector

Value

a vector of 1's and 0's, 6 bits per input element (64-QAM symbol)

See Also

Other rwirelesscom functions: eyediagram; f16pskdemod; f16pskmod; f16qamdemod; f16qammod; f64qammod; f8pskdemod; f8pskmod; fNo; fbpskdemod; fbpskmod; fqpskdemod; iqdensityplot; iqscatterplot; stemplot

Examples

```
M=64
Es=42
Eb = Es/log2(M)
Nsymbols=1000
Nbits=log2(M)*Nsymbols
bits <- sample(0:1,Nbits, replace=TRUE)
s <- f64qammod(bits)
EbNodB=12
No = Eb/(10^(EbNodB/10))
n <- fNo(Nsymbols,No,type="complex")
r <- s+n
bitsr <- f64qamdemod(r)
biterrs<-bits[bitsr!=bits]
Pberr=length(biterrs)/length(bits)
```

---

f64qammod *64-QAM Modulator*

Description

Receives a vector of bits (1's and 0's). The received vector is mapped to an in-phase (real) and quadrature (imaginary) 64-QAM (6 bit) symbol according to a Binary Reflective Gray Code (BRGC, see reference). In-phase and quadrature constellation points take on values -7, -5, -3, -1, +1, +3, +5, +7, respectively, corresponding to a symbol energy Es = 42. Mapping of bits to 64-QAM constellation points is illustrated in the following constellation.

-7+7i (100000)	-5+7i (100001)	+3+7i (100011)	-1+7i (100010)	+1+7i (100110)	+3+7i (100111)	+5+7i (100101)	+7+7i (100100)
-7+5i (101000)	-5+5i (101001)	+3+5i (101011)	-1+5i (101010)	+1+5i (101110)	+3+5i (101111)	+5+5i (101101)	+7+5i (101100)
-7+3i (111000)	-5+3i (111001)	+3+3i (111011)	-1+3i (111010)	+1+3i (111110)	+3+3i (111111)	+5+3i (111101)	+7+3i (111100)
-7+1i (110000)	-5+1i (110001)	+3+1i (110011)	-1+1i (110010)	+1+1i (110110)	+3+1i (110111)	+5+1i (110101)	+7+1i (110100)
-7-1i (010000)	-5-1i (010001)	+3-1i (010011)	-1-1i (010010)	+1-1i (010110)	+3-1i (010111)	+5-1i (010101)	+7-1i (010100)

-7-3i (011000)	-5-3i (011001)	+3-3i (011011)	-1-3i (011010)	+1-3i (011110)	+3-3i (011111)	+5-3i (011101)	+7-3i (011100)
-7-5i (001000)	-5-5i (001001)	+3-5i (001011)	-1-5i (001010)	+1-5i (001110)	+3-5i (001111)	+5-5i (001101)	+7-5i (001100)
-7-7i (000000)	-5-7i (000001)	+3-7i (000011)	-1-7i (000010)	+1-7i (000110)	+3-7i (000111)	+5-7i (000101)	+7-7i (000100)

Reference: E. Agrell, J Lassing, E. Strom, and T. Ottosson, Gray Coding for Multilevel Constellations In Gaussian Noise, IEEE Transactions on Communications, Vol. 53, No. 1, January 2007

Usage

```
f64qammod(bits, Ns = 1, p = 1)
```

Arguments

`bits`	- received vector of bits (0's and 1's).
`Ns`	- N samples per symbol (default, Ns = 1)
`p`	- a vector defining the pulse shape of the transmitted waveform (default, p = 1)

Value

Returns a complex vector of 64-QAM symbols. If Ns > 1 then the returned signal is shaped with pulse shape, p.

See Also

Other rwirelesscom functions: `eyediagram`; `f16pskdemod`; `f16pskmod`; `f16qamdemod`; `f16qammod`; `f64qamdemod`; `f8pskdemod`; `f8pskmod`; `fNo`; `fbpskdemod`; `fbpskmod`; `fqpskdemod`; `iqdensityplot`; `iqscatterplot`; `stemplot`

Examples

```
M=64
Es=42
Eb = Es/log2(M)
Nsymbols=1000
Nbits=log2(M)*Nsymbols
bits <- sample(0:1,Nbits, replace=TRUE)
s <- f64qammod(bits)
```

f8pskdemod *8-PSK Demodulator*

Description

Receives a vector of complex values, r, corresponding to an 8-PSK modulated signal transmitted through a communications channel (e.g., signal plus noise). Three bits are output for each received symbol according to the following decision rules

input	output
$-\pi/8 \geq Arg(r) < \pi/8$	000
$\pi/8 \geq Arg(r) < 3\pi/8$	001
$3\pi/8 \geq Arg(r) < 5\pi/8$	011
$5\pi/8 \geq Arg(r) < 7\pi/8$	010
$7\pi/8 \geq Arg(r) < 9\pi/8$	110
$-7\pi/8 \geq Arg(r) < -5\pi/8$	111
$-5\pi/8 \geq Arg(r) < -3\pi/8$	101
$-3\pi/8 \geq Arg(r) < -\pi/8$	100

Usage

```
f8pskdemod(r)
```

Arguments

r	- received signal

Value

returns a vector of 1's and 0's, 3 bits per input element (i.e., 8-PSK symbol)

See Also

Other rwirelesscom functions: eyediagram; f16pskdemod; f16pskmod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskmod; fNo; fbpskdemod; fbpskmod; fqpskdemod; iqdensityplot; iqscatterplot; stemplot

Examples

```
M=8
Es=1
Eb = Es/log2(M)
Nsymbols=10
Nbits=log2(M)*Nsymbols
bits <- sample(0:1,Nbits, replace=TRUE)
s <- f8pskmod(bits)
EbNodB=7
No = Eb/(10^(EbNodB/10))
```

```
n <- fNo(Nsymbols,No,type="complex")
r <- s+n
bitsr <- f8pskdemod(r)
biterrs<-bits[bitsr!=bits]
b<-factor(bits)
Pberr=length(biterrs)/length(bits)
```

---

f8pskmod *8-PSK Modulator*

Description

Receives a vector of bits (1's and 0's). The received vector is mapped to in-phase (real) and quadrature (imaginary) components, according to a Binary Reflective Gray Code (BRGC, see reference). Each received pair of bits are are mapped to 8-PSK symbols, with $sqrt(E_s)$ (symbol energy) = 1.The bit to symbol mapping is illustrated in the following constellation diagram.

input	output
000	0
001	$\pi/4$
011	$\pi/2$
010	$3\pi/4$
110	pi
111	$-3\pi/4$
101	$-\pi/2$
100	$-\pi/4$

Reference: E. Agrell, J Lassing, E. Strom, and T. Ottosson, Gray Coding for Multilevel Constellations In Gaussian Noise, IEEE Transactions on Communications, Vol. 53, No. 1, January 2007

Usage

```
f8pskmod(bits, Ns = 1, p = 1)
```

Arguments

bits	- vector of bits (0's and 1's).
Ns	- N samples per symbol (default, Ns = 1)
p	- a vector defining the pulse shape of the transmitted waveform (default, p = 1)

Value

Returns a complex vector of length = (length(bits) mod 3), 8-PSK symbols. If Ns > 1 then the returned signal is shaped with pulse shape, p.

See Also

Other rwirelesscom functions: eyediagram; f16pskdemod; f16pskmod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; fNo; fbpskdemod; fbpskmod; fqpskdemod; iqdensityplot; iqscatterplot; stemplot

Examples

```
M=8
Nsymbols=10
Nbits=log2(M)*Nsymbols
bits <- sample(0:1,Nbits, replace=TRUE)
s <- f8pskmod(bits)
```

---

fbpskdemod *BPSK Demodulator*

Description

Receives a vector of real values, corresponding to a BPSK modulated signal transmitted through a communications channel (e.g., signal plus noise). An input value < 1 is mapped to an output value of 0, otherwise to a value of 1.

Usage

```
fbpskdemod(r)
```

Arguments

r	- received signal vector

Value

returns a vector of 1's and 0's corresponding to BPSK demodulation of the input vector

See Also

Other rwirelesscom functions: eyediagram; f16pskdemod; f16pskmod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fNo; fbpskmod; fqpskdemod; iqdensityplot; iqscatterplot; stemplot

Examples

```
Eb=1
Nbits=10
bits <- sample(0:1,Nbits, replace=TRUE)
s <- fbpskmod(bits)
EbNodB=8
No = Eb/(10^(EbNodB/10))
```

```
n <- fNo(Nbits,No)
r <- s+n
bitsr <- fbpskdemod(r)
biterrs<-bits[bitsr!=bits]
Pberr=length(biterrs)/length(bits)
```

---

fbpskmod *BPSK Modulator*

Description

Receives a vector of bits, each with value 0 or 1, and outputs a vector with values 1 and -1, respectively.

Usage

```
fbpskmod(bits, Ns = 1, p = 1)
```

Arguments

bits	- vector of bits (0's and 1's)
Ns	- N samples per symbol (default, Ns = 1)
p	- a vector defining the pulse shape of the transmitted waveform (default, p = 1)

Value

Returns a BPSK modulated vector, each element taking on values of 1 or -1. If Ns > 1 then the returned signal is shaped with pulse shape, p.

See Also

Other rwirelesscom functions: eyediagram; f16pskdemod; f16pskmod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fNo; fbpskdemod; fqpskdemod; iqdensityplot; iqscatterplot; stemplot

Examples

```
bits <- sample(0:1,10, replace=TRUE)
fbpskmod(bits)
```

fNo *AWGN*

Description

Generates a vector of normally distributed noise samples with mean of zero and noise spectral density (No/2), a.k.a. AWGN.

Usage

```
fNo(N, No, type = "real")
```

Arguments

N	- number of noise samples
No	- noise spectral density
type	- "real" or "complex" defaults to real

Value

returns a vector of distributed noise samples of length N, mean of zero and variance of No/2

See Also

Other rwirelesscom functions: eyediagram; f16pskdemod; f16pskmod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fbpskdemod; fbpskmod; fqpskdemod; iqdensityplot; iqscatterplot; stemplot

Examples

```
n <- fNo(N=10,No=10)
bits <- sample(0:1,10, replace=TRUE)
s=fbpskmod(bits)
r=s+n

n <- fNo(N=20,No=10,type="complex")
bits <- sample(0:1,20, replace=TRUE)
s=fqpskmod(bits)
r=s+n
```

fqpskdemod *QPSK Demodulator*

Description

Receives a vector of complex values, r, corresponding to a QPSK modulated signal transmitted through a communications channel (e.g., signal plus noise). The received signal, r, is mapped to its in-phase (real part) and quadrature parts (imaginary part) and demodulated, such that two binary bits are output for each value of r. If the in-phase part is > 0 then the corresponding output bit value is 1, otherwise 0. Similarly, if the quadrature part (imaginary) > 0 then the corresponding bit value is 1, otherwise 0.

Usage

```
fqpskdemod(r)
```

Arguments

r	- received signal plus noise.

Value

returns a vector of 1's and 0's, 2 bits per input element (i.e., QPSK symbol)

See Also

Other rwirelesscom functions: eyediagram; f16pskdemod; f16pskmod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fNo; fbpskdemod; fbpskmod; iqdensityplot; iqscatterplot; stemplot

Examples

```
M=4
Es=1
Nsymbols=10
Nbits=log2(M)*Nsymbols
Eb=Es/log2(M)
bits <- sample(0:1,Nbits, replace=TRUE)
s <- fqpskmod(bits)
EbNodB=8
No = Eb/(10^(EbNodB/10))
n <- fNo(Nsymbols,No,type="complex")
r <- s+n
bitsr <- fqpskdemod(r)
biterrs<-bits[bitsr!=bits]
Pberr=length(biterrs)/length(bits)
```

fqpskmod *QPSK Modulator*

Description

Receives a vector of bits (1's and 0's). The 1's and 0's are mapped to in-phase (real) and quadrature (imaginary) components. Correspondingly, a bit of 1 is mapped to +1/sqrt(2), otherwise to -1/sqrt(2) according to the following mapping.

input	output
00	(-1 - 1i) / sqrt(2)
01	(-1 + 1i) / sqrt(2)
10	(+1 - 1i) / sqrt(2)
11	(+1 + 1i) / sqrt(2)

Usage

```
fqpskmod(bits, Ns = 1, p = 1)
```

Arguments

bits	- received vector of bits (0's and 1's).
Ns	- N samples per symbol (default, Ns = 1)
p	- a vector defining the pulse shape of the transmitted waveform (default, p = 1)

Value

Returns a complex vector of QPSK symbols. If Ns > 1 then the returned signal is shaped with pulse shape, p.

Examples

```
M=4
Nsymbols=10
Nbits=log2(M)*Nsymbols
bits <- sample(0:1,Nbits, replace=TRUE)
s <- fqpskmod(bits)
```

iqdensityplot *IQ Density Plot*

Description

A convenience function to plot a density function of a vector containing the in-phase and quadrature signal (plus noise).

Usage

```
iqdensityplot(r, iq = "r")
```

Arguments

r	- complex or real valued vector
iq	- if iq = "r" (default) then plot density of Re(r) else if iq = "q" then plot density of Im(r)

See Also

Other rwirelesscom functions: eyediagram; f16pskdemod; f16pskmod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fNo; fbpskdemod; fbpskmod; fqpskdemod; iqscatterplot; stemplot

Examples

```
M=4
Es=1
Eb = Es/log2(M)
Nsymbols=1000
Nbits=log2(M)*Nsymbols
bits <- sample(0:1,Nbits, replace=TRUE)
s <- fqpskmod(bits)
EbNodB=4
No = Eb/(10^(EbNodB/10))
n <- fNo(Nsymbols,No,type="complex")
r <- s+n
```

---

iqscatterplot *IQ Scatter Plot*

Description

A convenience function that plots a scatter diagram of Im(r) vs. Re(r). The function is useful for visualizing constellations such as M-PSK or M-QAM.

Usage

```
iqscatterplot(r)
```

Arguments

r	- complex or real valued vector

See Also

Other rwirelesscom functions: eyediagram; f16pskdemod; f16pskmod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fNo; fbpskdemod; fbpskmod; fqpskdemod; iqdensityplot; stemplot

Examples

```
M=8
Es=1
Eb = Es/log2(M)
Nsymbols=10000
Nbits=log2(M)*Nsymbols
bits <- sample(0:1,Nbits, replace=TRUE)
s <- f8pskmod(bits)
EbNodB=7
No = Eb/(10^(EbNodB/10))
n <- fNo(Nsymbols,No,type="complex")
r <- s+n
iqscatterplot(r)
```

---

rcosine *Raised Cosine*

Description

rcosine(x,B,Ns) = [sin(pi*x[n]/Ns)/(pi*x[n]/Ns)]*cos(pi*B*x[n]/Ns) /(1-(2*B*x[n]/Ns)^2)

Reference: G. Proakis, Digital Communications, 3rd ed., New York: McGraw-Hill, 1995.

Usage

```
rcosine(x,B,Ns)
```

Arguments

x	- input vector
B	- roll-off factor
Ns	- Ns samples per symbol

Value

Response (Double, Numeric vector) of the rcosine function applied to the input vector x with roll-off factor B and Ns samples per symbol.

See Also

Other rwirelesscom functions: f16pskdemod; f16pskmod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fbpskdemod; fqpskdemod; fqpskmod; rcosine; sqrtrcosine; iqdensityplot; iqscatterplot; stemplot; eyediagram; sinc

Examples

```
B=0
Ns=8
hx=seq(-3*Ns,3*Ns,by=1)
h=rcosine(hx,B,Ns)
plot(hx/Ns,h, ylim=c(-0.2,1), xlim=c(-3,3), pch=19, cex=0.1, ylab="h", xlab="T", type="l")
grid( col = "grey50", lty = "dotted")
```

`rwirelesscom` *R Wireless Communications Package*

Description

A communications simulation package supporting common modulations formats including BPSK, QPSK, 8-PSK, 16-PSK, 16-QAM and 64-QAM. The package includes AWGN noise generation, and raised cosine and square root raised cosine pulse shaping. It also includes functions for plotting constellation diagrams, density plots, stem plots and eye diagrams. The rwirelesscom package includes the followng functions:

- fNo(),
- fbpskmod(), fbpskdemod(),
- f8pskmod(), f8pskdemod(),
- f16pskmod(), f16pskdemod(),
- f16qammod(), f16qamdemod(),
- f64qammod(), f64qamdemod,
- rcosine(), sqrtrcosine(), sinc(),
- iqscatterplot(), iqdensityplot(),
- stemplot(), eyediagram()

Together these functions enable the evaluation of bit error and symbol error rates, evalutation of pulse shaping and inter-symbol interferance and support visualization of the respective signals and noise.

`sinc` *sinc*

Description

sinc(x) = 1 at x = 0, sinc(x) = sin(x)/x for x != 0

Usage

```
sinc(x)
```

Arguments

x	- input vector

Value

Response (Double, Numeric vector) of the sinc function applied to the input vector x.

See Also

Other rwirelesscom functions: f16pskdemod; f16pskmod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fbpskdemod; fqpskdemod; fqpskmod; rcosine; sqrtrcosine; iqdensityplot; iqscatterplot; stemplot; eyediagram; sinc

Examples

```
x <- seq(-6*pi, 6*pi, by = pi/10)
p <- sinc(x)
plot(x/pi,p,ylim=c(-0.3,1.1), pch=19, cex=0.25, ylab="p", xlab="x*pi", type="l")
grid( col = "grey50", lty = "dotted")
```

---

sqrtrcosine *Square Root Raised Cosine*

Description

sqrtrcosine(x,B,Ns) = [sin((1-B)*pi*x[n]/Ns)/(pi*x[n]/Ns)+(4*B*x[n]/Ns)*cos((1+B)*pi*x[n]/Ns)]/((pi*x[n]/Ns)(1-(4*B*x[n]/Ns)^2))

Reference:

1. S. Daumont, R. Basel, Y. Louet, "Root-Raised Cosine filter influences on PAPR distribution of single carrier signals", ISCCSP 2008, Malta, 12-14 March 2008.

2. 3GPP TS 25.104 V6.8.0 (2004-12)

Usage

```
sqrtrcosine(x,B,Ns)
```

Arguments

x	- input vector
B	- roll-off factor
Ns	- Ns samples per symbol

Value

Response (Double, Numeric vector) of the suare root raised rcosine function applied to the input vector x with roll-off factor B and Ns samples per symbol.

See Also

Other rwirelesscom functions: f16pskdemod; f16pskmod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fbpskdemod; fqpskdemod; fqpskmod; rcosine; sqrtrcosine; iqdensityplot; iqscatterplot; stemplot; eyediagram; sinc

Examples

```
Ns=16
B=0.5
hx=seq(-5*Ns,5*Ns,by=1)
h1=sqrtrcosine(hx,B,Ns=Ns)
plot(hx/Ns,h1, ylim=c(-0.2,1.1), xlim=c(-3,3), pch=19, cex=0.1, ylab="h", xlab="T", type="l")
grid( col = "grey50", lty = "dotted")
```

`stemplot` *Stem Plot*

Description

Receives a vector of x and y values and plots a stemplot (line and "points").

Usage

```
stemplot(x, y, pch = 16, linecol = 1, linew = 1, ...)
```

Arguments

x	- vector of x axis points
y	- vector of y axis points
pch	- plot character default = 19
linecol	- default line color = 1 (black)
linew	- default line width = 1
...	- graphical environment parameters are input to stemplot

Details

#' Reference: M, Pastell, http://www.r-bloggers.com/matlab-style-stem-plot-with-r

See Also

Other rwirelesscom functions: eyediagram; f16pskdemod; f16pskmod; f16qamdemod; f16qammod; f64qamdemod; f64qammod; f8pskdemod; f8pskmod; fNo; fbpskdemod; fbpskmod; fqpskdemod; iqdensityplot; iqscatterplot

Examples

```
x <- seq(-3*pi, 3*pi, by = 0.2)
y <- sinc(x)
stemplot(x/pi,y,ylim=c(-0.3,1.1), pch=19, cex=0.3, ylab="y", xlab="x/pi")
```

Index