Mobile Communication

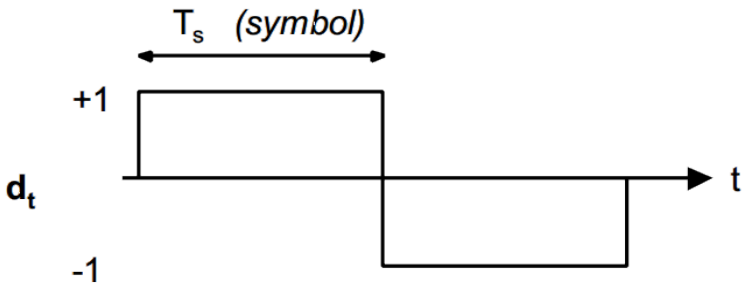
DSSS and FHSS intermediate presentation

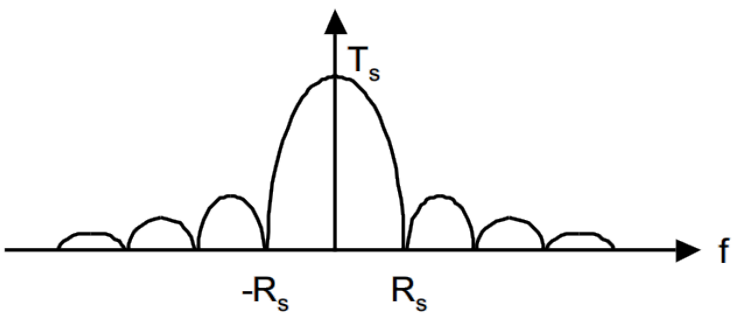
Group 6

October 31, 2014

Spread Spectrum

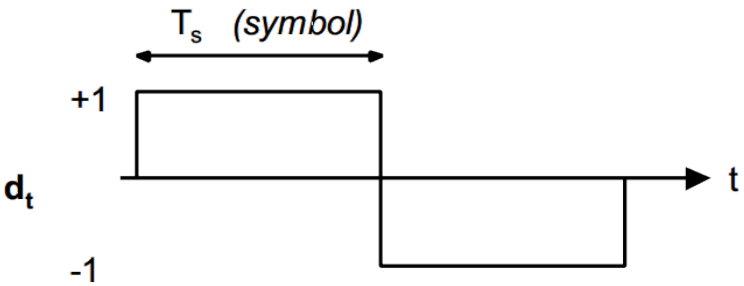
- Transmitting finite sequences requires a frequency *band*
- Spreading this band makes transmission more robust
- Use spreading schemes, that allows using the frequency band for concurrent transmission

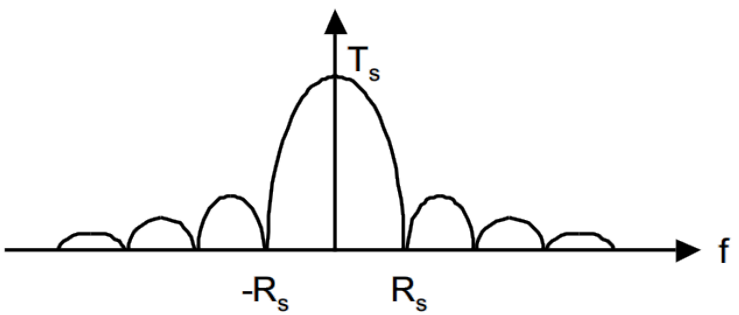




Spread Spectrum

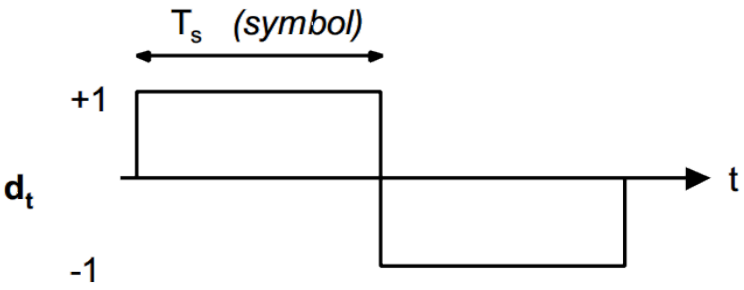
- Transmitting finite sequences requires a frequency *band*
- Spreading this band makes transmission more robust
- Use spreading schemes, that allows using the frequency band for concurrent transmission

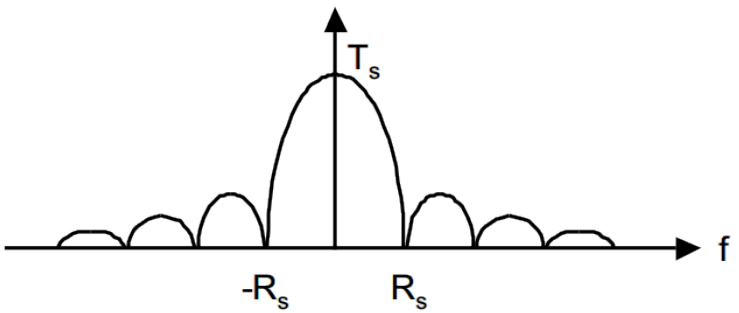




Spread Spectrum

- Transmitting finite sequences requires a frequency *band*
- Spreading this band makes transmission more robust
- Use spreading schemes, that allows using the frequency band for concurrent transmission

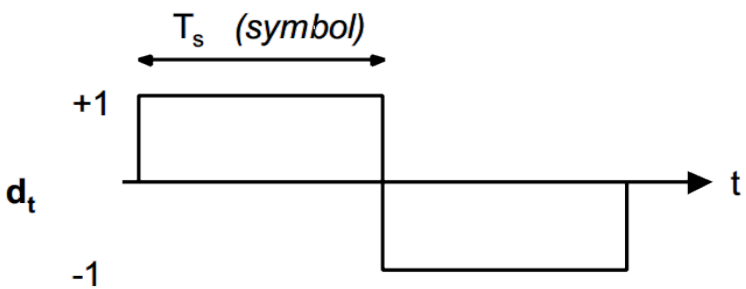


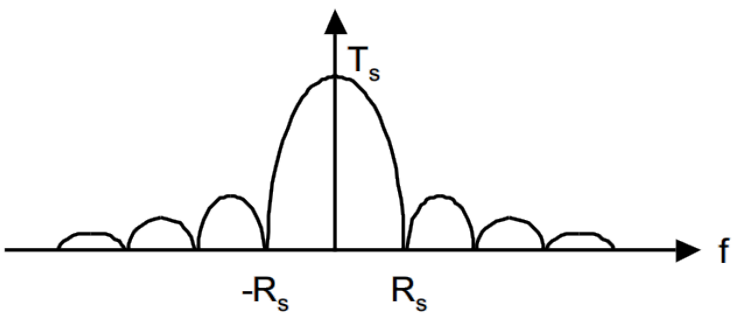


Direct Sequence Spread Spectrum

Data

- Let the data $d_t \in \{-1, 1\}^n$ be $d_t = [1, -1]$
- Signal bandwidth R_s

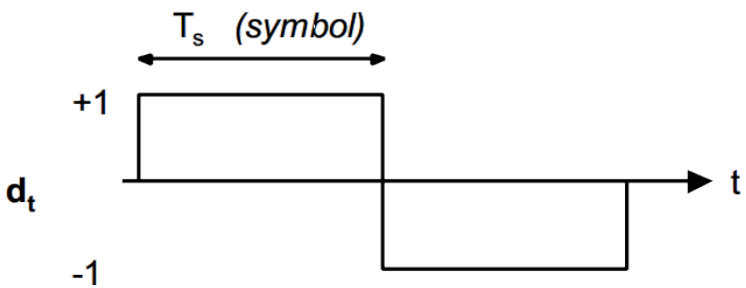


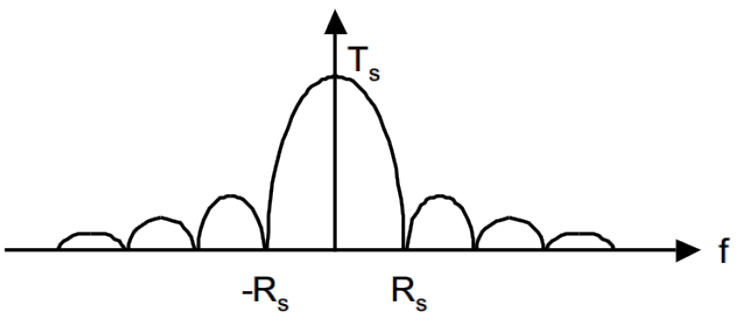


Direct Sequence Spread Spectrum

Data

- Let the data $d_t \in \{-1, 1\}^n$ be $d_t = [1, -1]$
- Signal bandwidth R_s

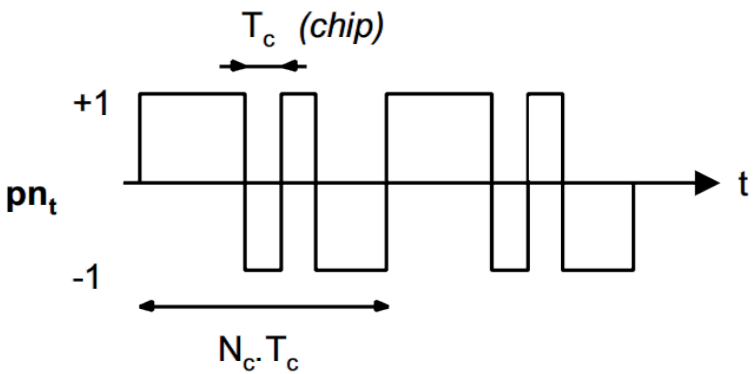


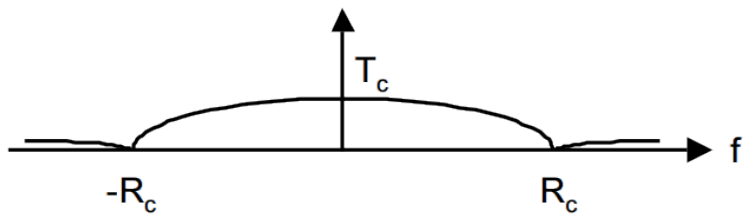


Direct Sequence Spread Spectrum

Chip Sequence

- Define chip sequence. Let the sequence $p_n \in \{-1, 1\}^n$ be $d_t = [1, 1, 1, -1, 1, -1, -1, 1, 1, 1, -1, 1, -1, -1]$
- Signal bandwidth R_c with $R_c > R_s$.
- p_n is known to sender and receiver only
- Sender and receiver are synchronized

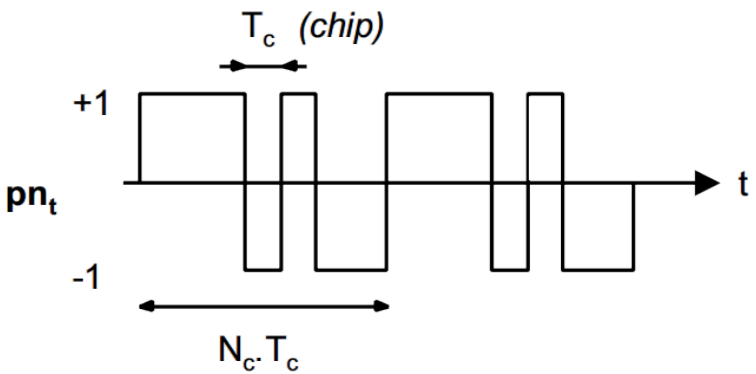


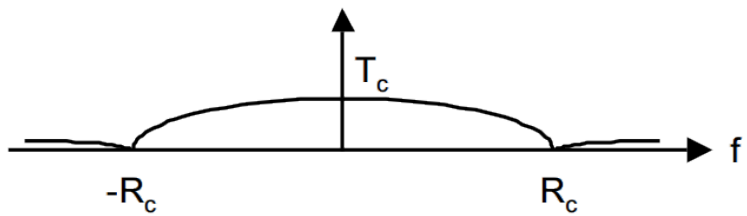


Direct Sequence Spread Spectrum

Chip Sequence

- Define chip sequence. Let the sequence $p_n \in \{-1, 1\}^n$ be $d_t = [1, 1, 1, -1, 1, -1, -1, 1, 1, 1, -1, 1, -1, -1]$
- Signal bandwidth R_c with $R_c > R_s$.
- p_n is known to sender and receiver only
- Sender and receiver are synchronized

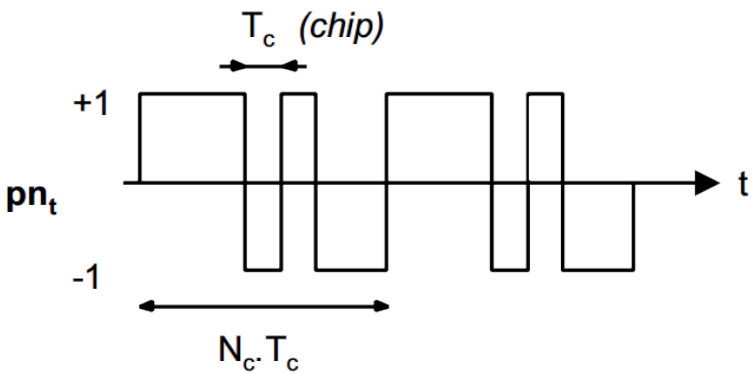


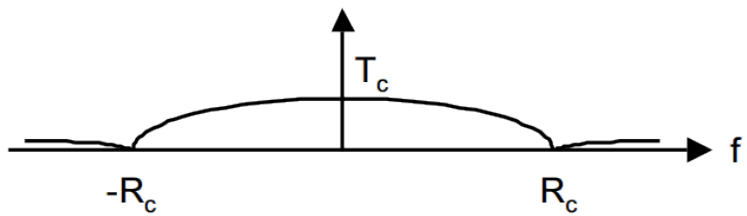


Direct Sequence Spread Spectrum

Chip Sequence

- Define chip sequence. Let the sequence $p_n \in \{-1, 1\}^n$ be $d_t = [1, 1, 1, -1, 1, -1, -1, 1, 1, 1, -1, 1, -1, -1]$
- Signal bandwidth R_c with $R_c > R_s$.
- p_n is known to sender and receiver only
- Sender and receiver are synchronized

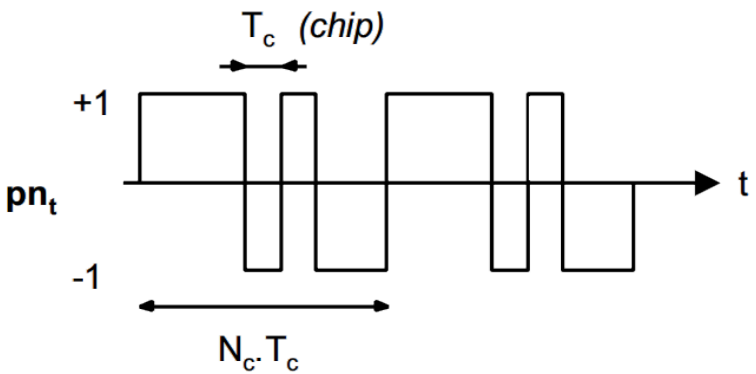


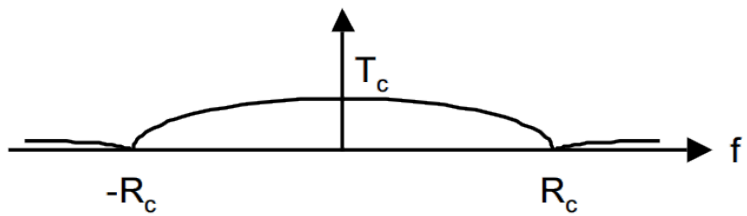


Direct Sequence Spread Spectrum

Chip Sequence

- Define chip sequence. Let the sequence $p_n \in \{-1, 1\}^n$ be $d_t = [1, 1, 1, -1, 1, -1, -1, 1, 1, 1, -1, 1, -1, -1]$
- Signal bandwidth R_c with $R_c > R_s$.
- p_n is known to sender and receiver only
- Sender and receiver are synchronized

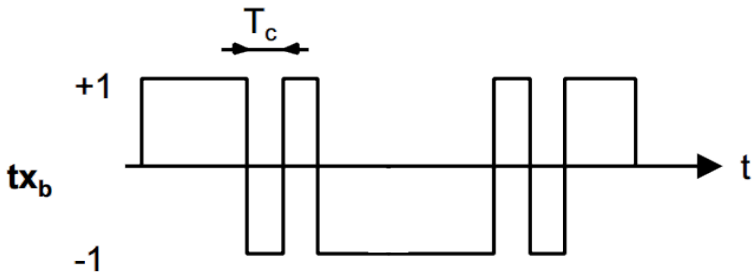


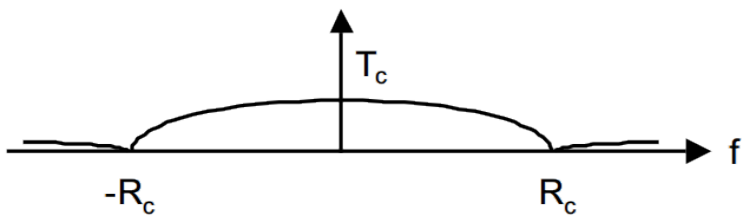


Direct Sequence Spread Spectrum

Spreading

- Point-wise multiply the data with chip sequence. $t_x = d_t p_n$
- Bandwidth of transmitted signal is R_c, the chip sequence's bandwidth.
- For transmission apply some phase modulation

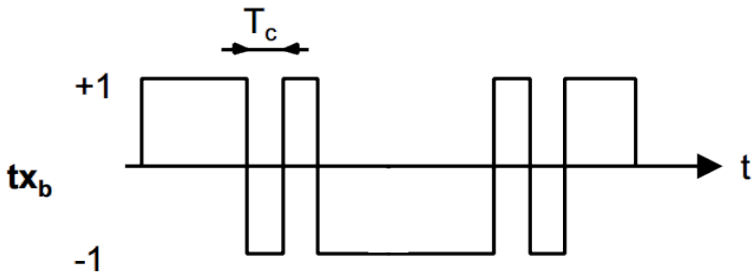


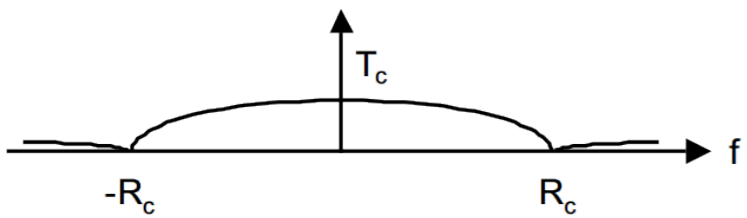


Direct Sequence Spread Spectrum

Spreading

- Point-wise multiply the data with chip sequence. $t_x = d_t p_n$
- Bandwidth of transmitted signal is R_c, the chip sequence's bandwidth.
- For transmission apply some phase modulation

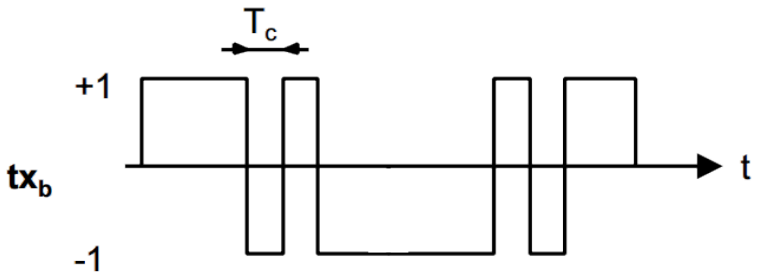


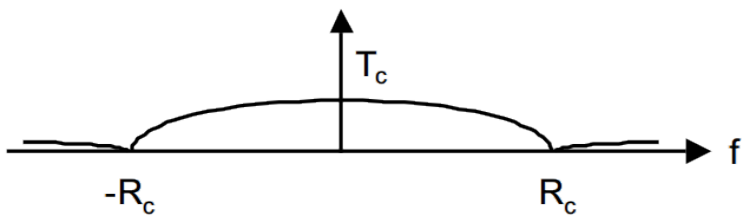


Direct Sequence Spread Spectrum

Spreading

- Point-wise multiply the data with chip sequence. $t_x = d_t p_n$
- Bandwidth of transmitted signal is R_c, the chip sequence's bandwidth.
- For transmission apply some phase modulation

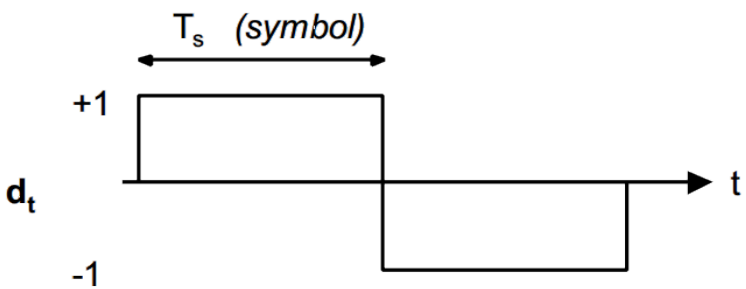


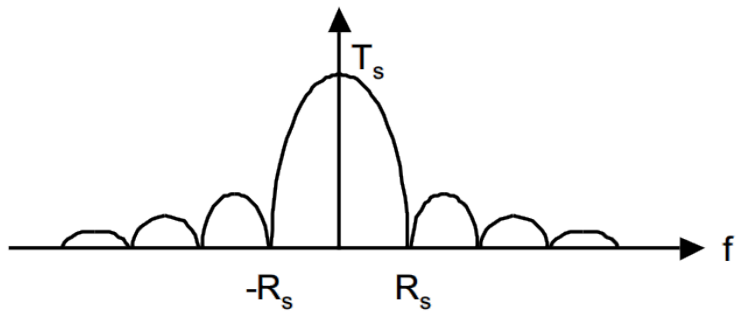


Direct Sequence Spread Spectrum

Spreading

- Point-wise multiply the signal with chip sequence.
  $d_r = t_x p_n = d_t p_n p_n = d_t$
- This returns the original data since $p_n p_n = [1, 1, \ldots]$

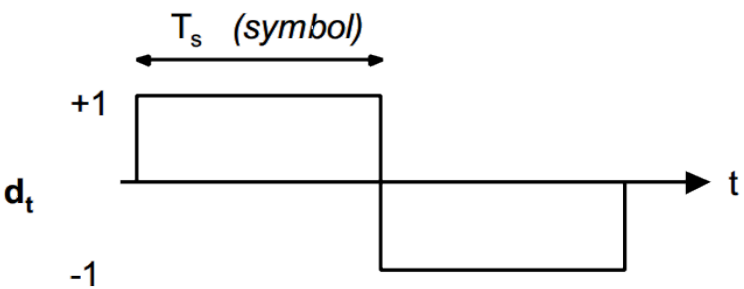


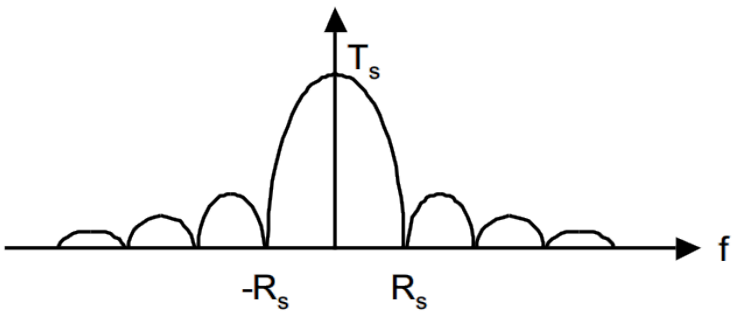


Direct Sequence Spread Spectrum

Spreading

- Point-wise multiply the signal with chip sequence.
  $d_r = t_x p_n = d_t p_n p_n = d_t$
- This returns the original data since $p_n p_n = [1, 1, \ldots]$

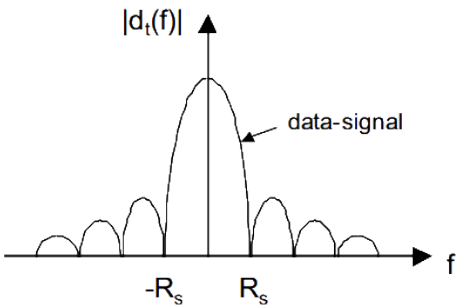


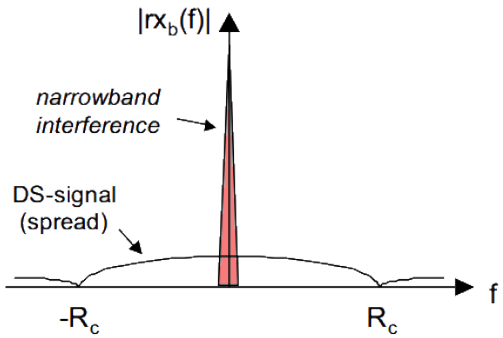


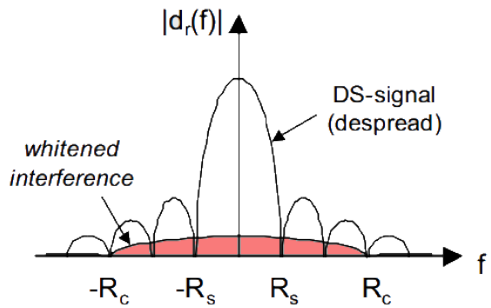
Interference

Narrow-band Interference

- Narrowband interference is spread in the despreading part
- Remember: spreading and despreading is the same operation
- Does not lower the SNR too much



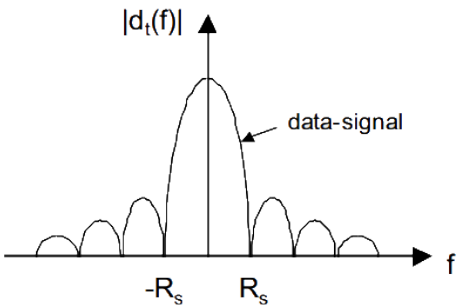


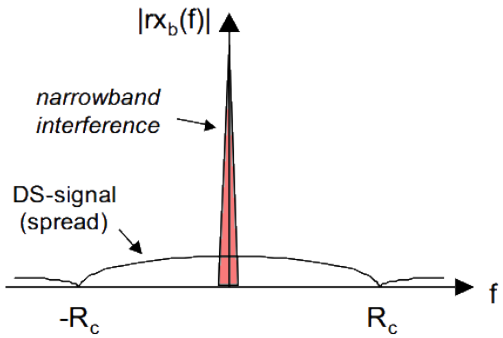


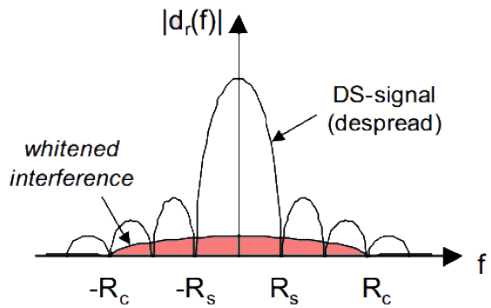
Interference

Narrow-band Interference

- Narrowband interference is spread in the despreading part
- Remember: spreading and despreading is the same operation
- Does not lower the SNR too much



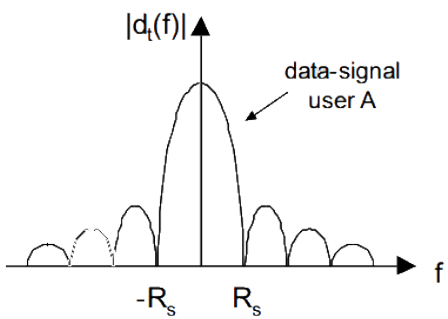


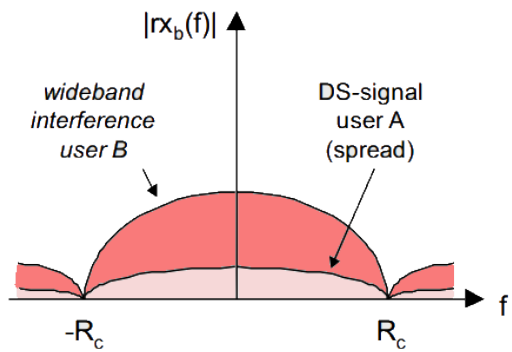


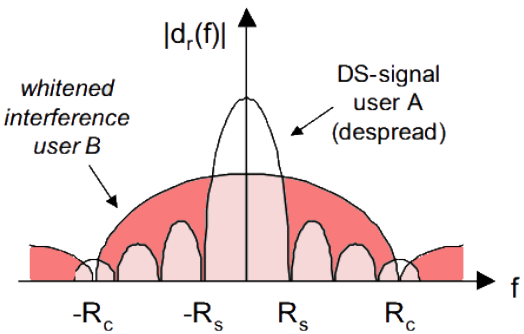
Interference

Broad-band Interference

- Despreading does not change the broad-band noise, it is uncorrelated with p_n. It's bandwidth remains the same.
- Can affect the SNR.
- Transmissions of other users are received as broadband noise



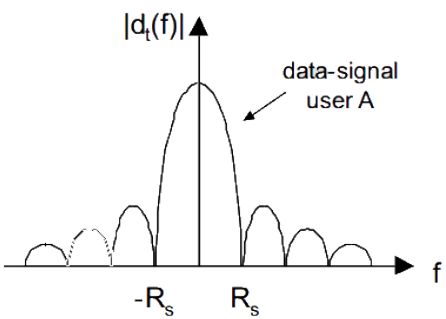


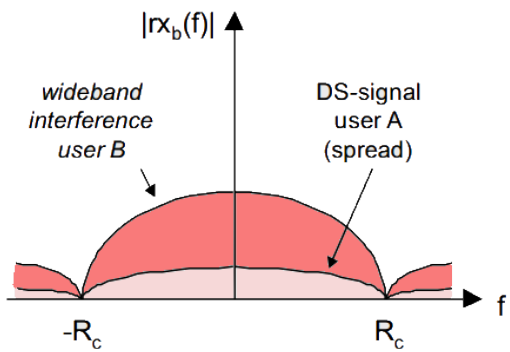


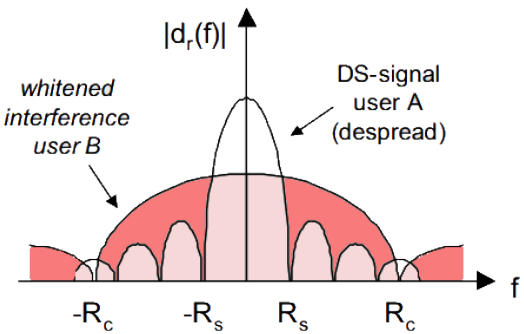
Interference

Broad-band Interference

- Despreading does not change the broad-band noise, it is uncorrelated with p_n. It's bandwidth remains the same.
- Can affect the SNR.
- Transmissions of other users are received as broadband noise



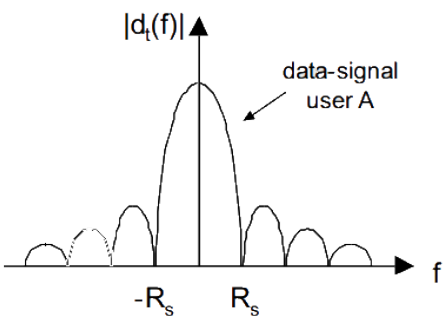


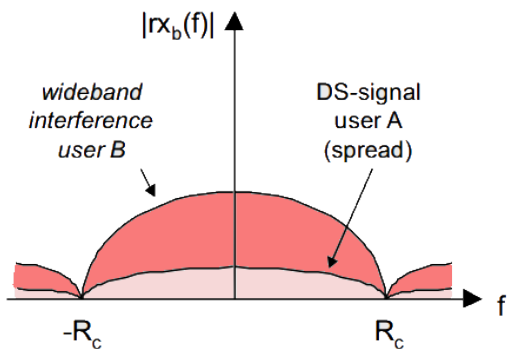


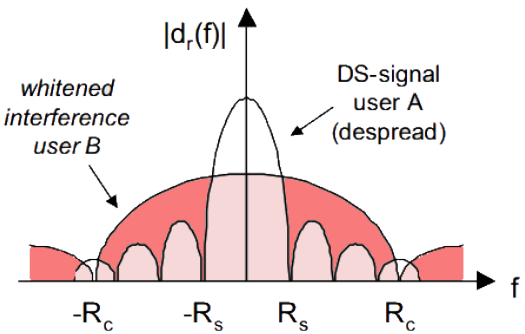
Interference

Broad-band Interference

- Despreading does not change the broad-band noise, it is uncorrelated with p_n. It's bandwidth remains the same.
- Can affect the SNR.
- Transmissions of other users are received as broadband noise

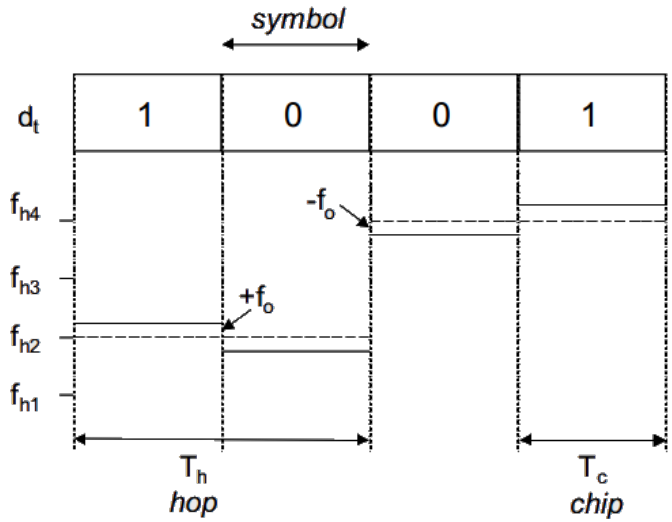


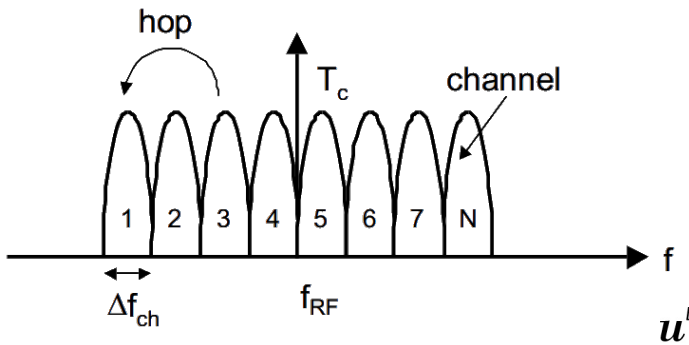




Frequency Hopping Spread Spectrum

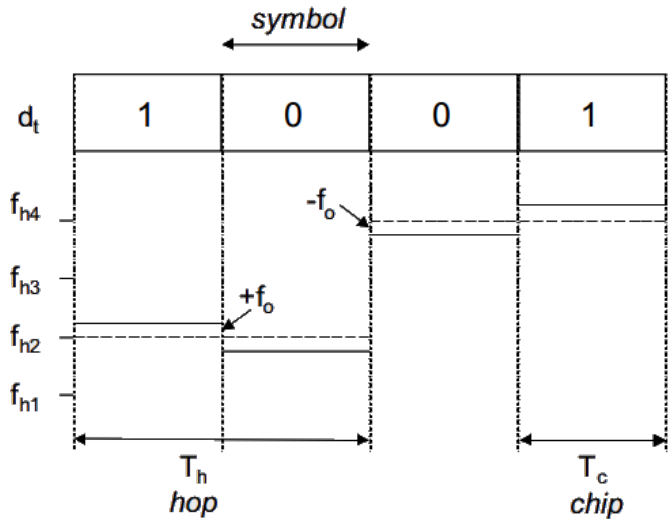
- Divide frequency band into *N* sub-bands
- Define a chip sequence $p_n \in [f_1, f_N]^n$. Let $p_n = [f2, f4, \ldots]$
- Transmit data on current frequency f_{i_j} according to chip pattern and hop to next frequency $f_{i_{j+1}}$ after some time
- Apply frequency modulation
- Sender and receiver know p_n and are synchronized

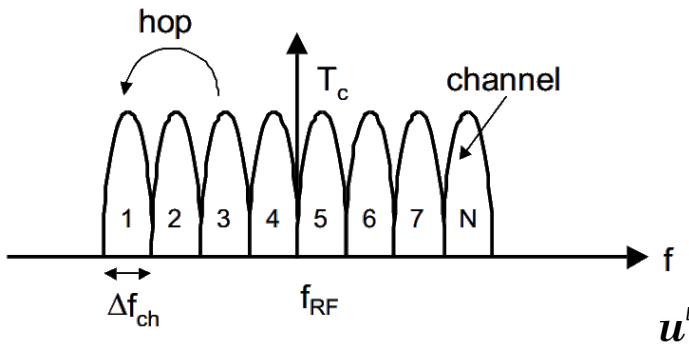




Frequency Hopping Spread Spectrum

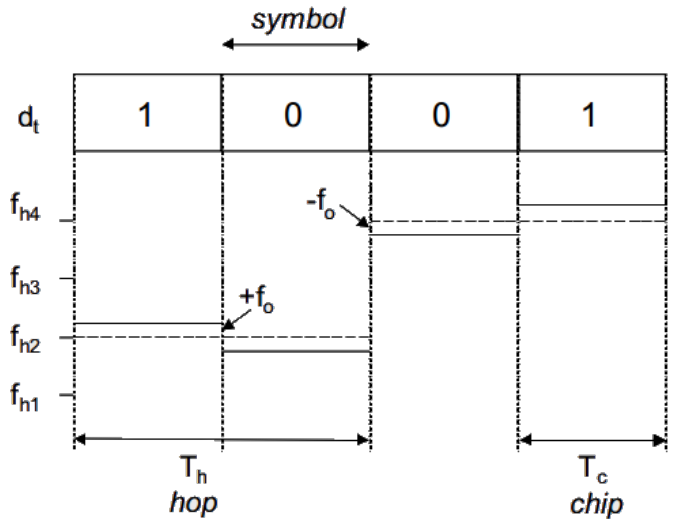
- Divide frequency band into N sub-bands
- Define a chip sequence $p_n \in [f_1, f_N]^n$. Let $p_n = [f2, f4, \ldots]$
- Transmit data on current frequency f_{i_j} according to chip pattern and hop to next frequency $f_{i_{j+1}}$ after some time
- Apply frequency modulation
- Sender and receiver know p_n and are synchronized

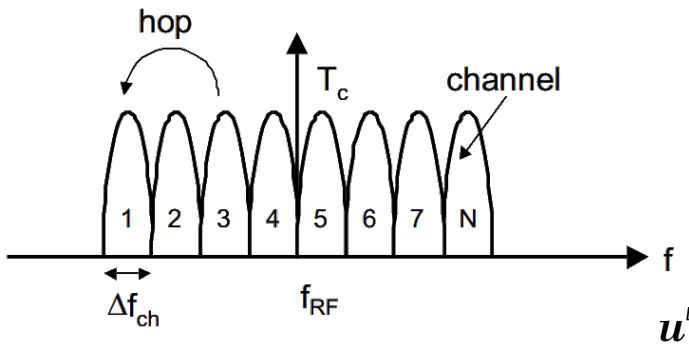




Frequency Hopping Spread Spectrum

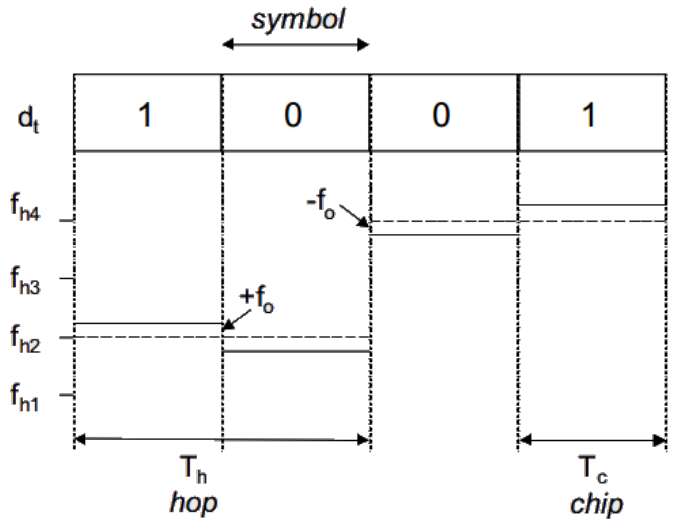
- Divide frequency band into N sub-bands
- Define a chip sequence $p_n \in [f_1, f_N]^n$. Let $p_n = [f2, f4, \ldots]$
- Transmit data on current frequency f_{i_j} according to chip pattern and hop to next frequency $f_{i_{j+1}}$ after some time
- Apply frequency modulation
- Sender and receiver know p_n and are synchronized

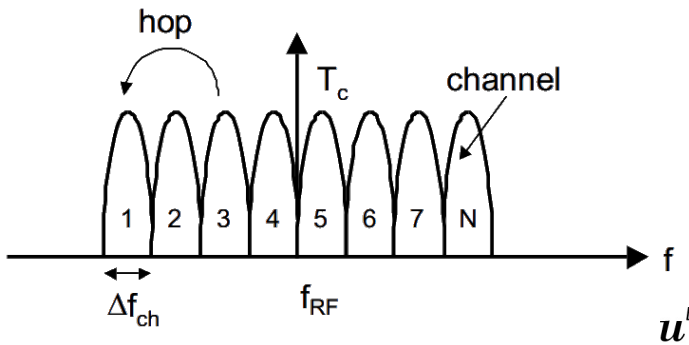




Frequency Hopping Spread Spectrum

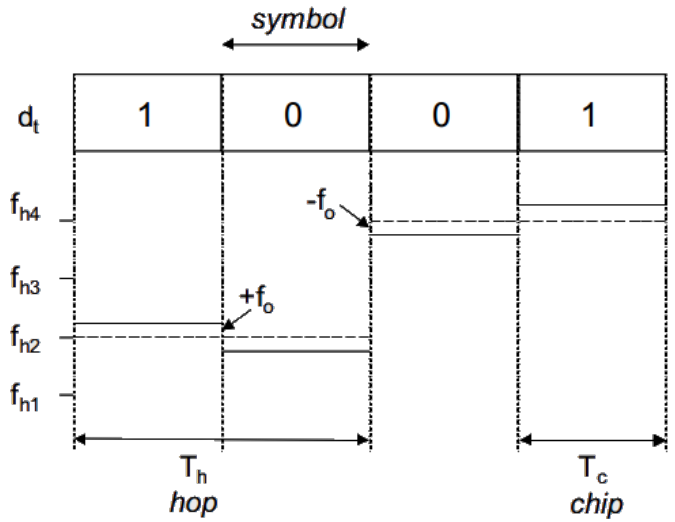
- Divide frequency band into N sub-bands
- Define a chip sequence $p_n \in [f_1, f_N]^n$. Let $p_n = [f2, f4, \ldots]$
- Transmit data on current frequency f_{i_j} according to chip pattern and hop to next frequency $f_{i_{j+1}}$ after some time
- Apply frequency modulation
- Sender and receiver know p_n and are synchronized

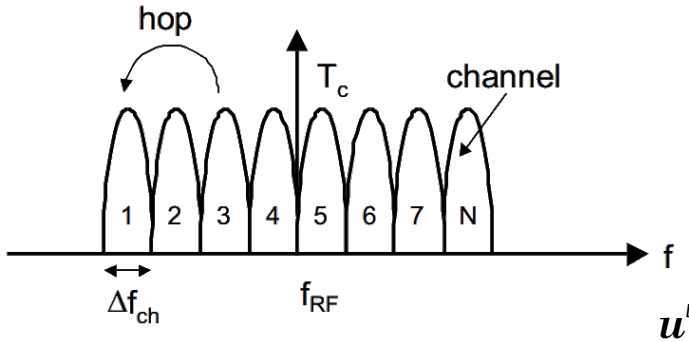




Frequency Hopping Spread Spectrum

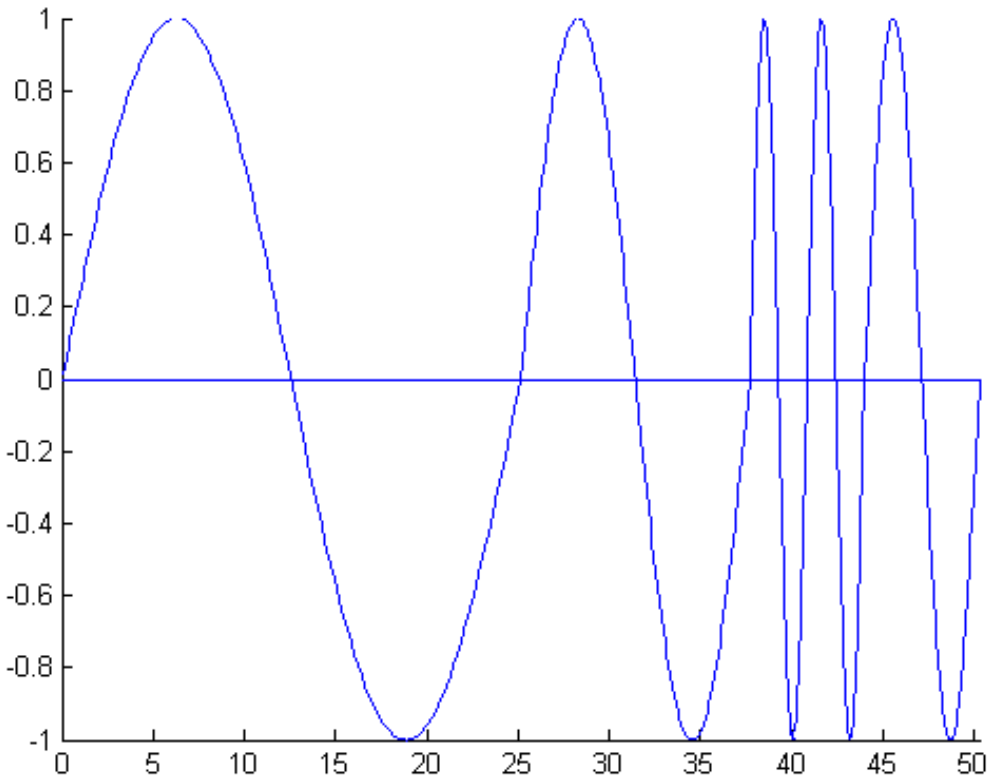
- Divide frequency band into N sub-bands
- Define a chip sequence $p_n \in [f_1, f_N]^n$. Let $p_n = [f2, f4, \ldots]$
- Transmit data on current frequency f_{i_j} according to chip pattern and hop to next frequency $f_{i_{j+1}}$ after some time
- Apply frequency modulation
- Sender and receiver know p_n and are synchronized





FHSS

Example



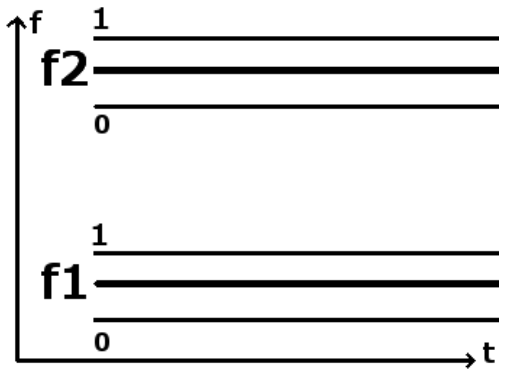
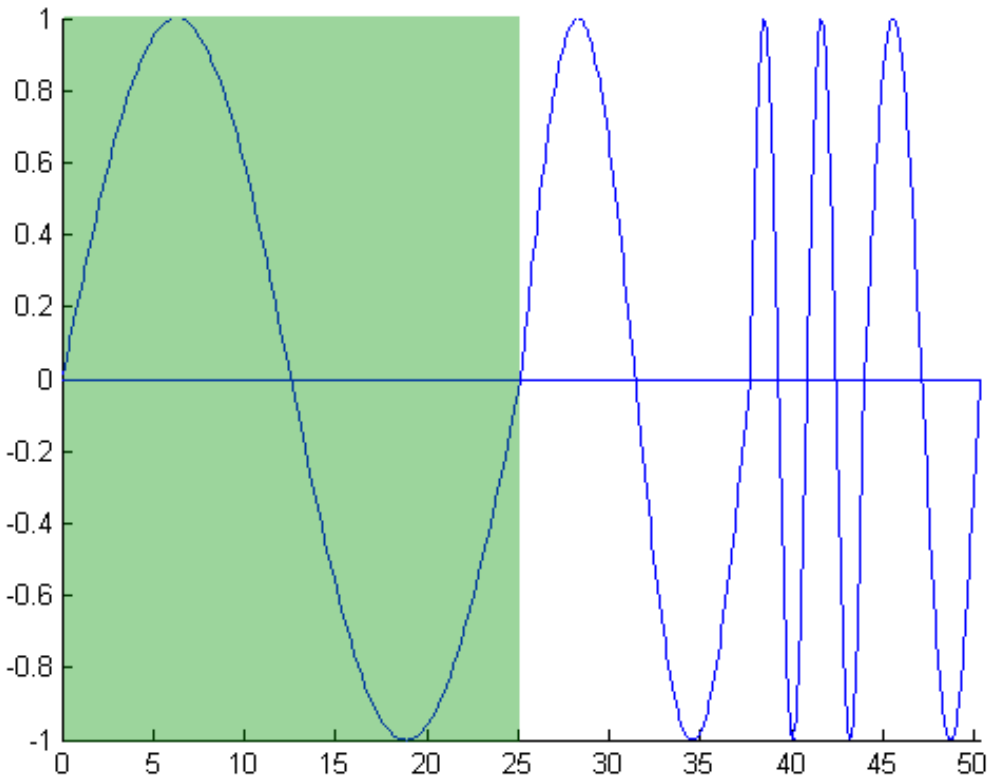


Figure: Frequency = , value =

FHSS

Example



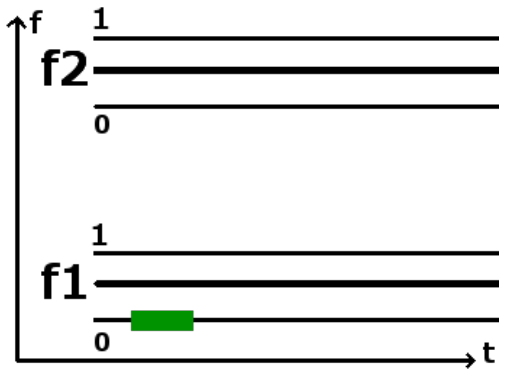
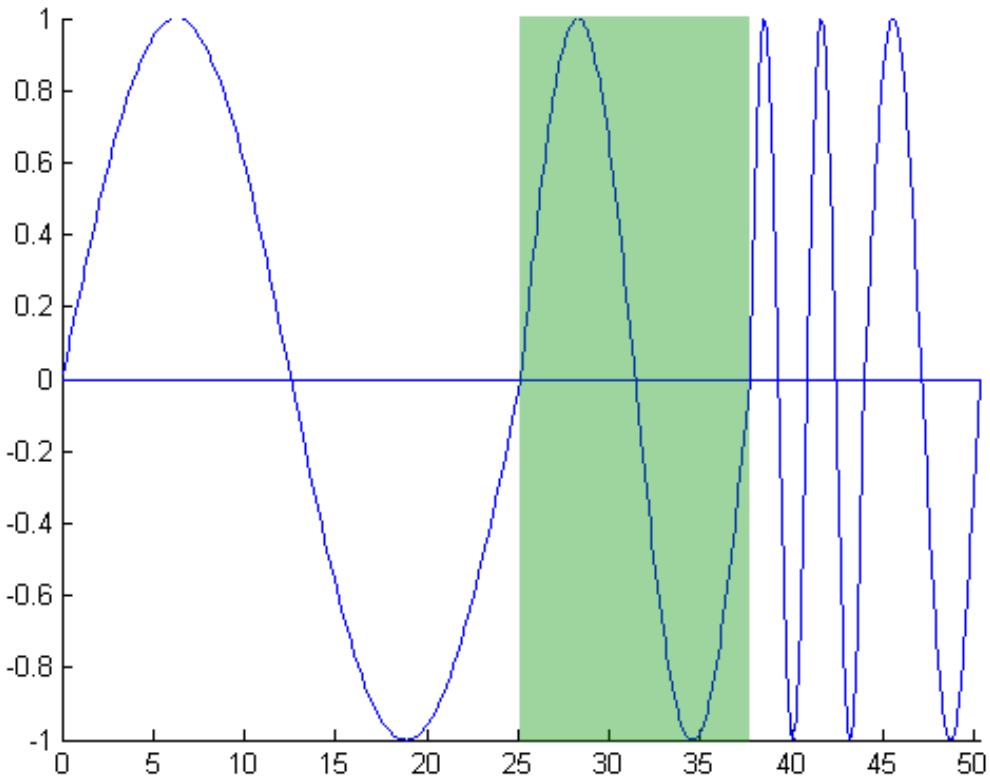


Figure: Frequency = f1, value = 0

FHSS

Example



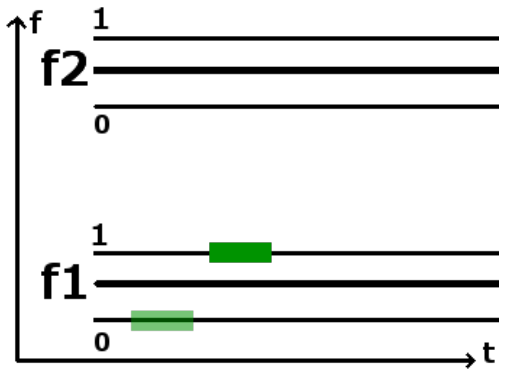
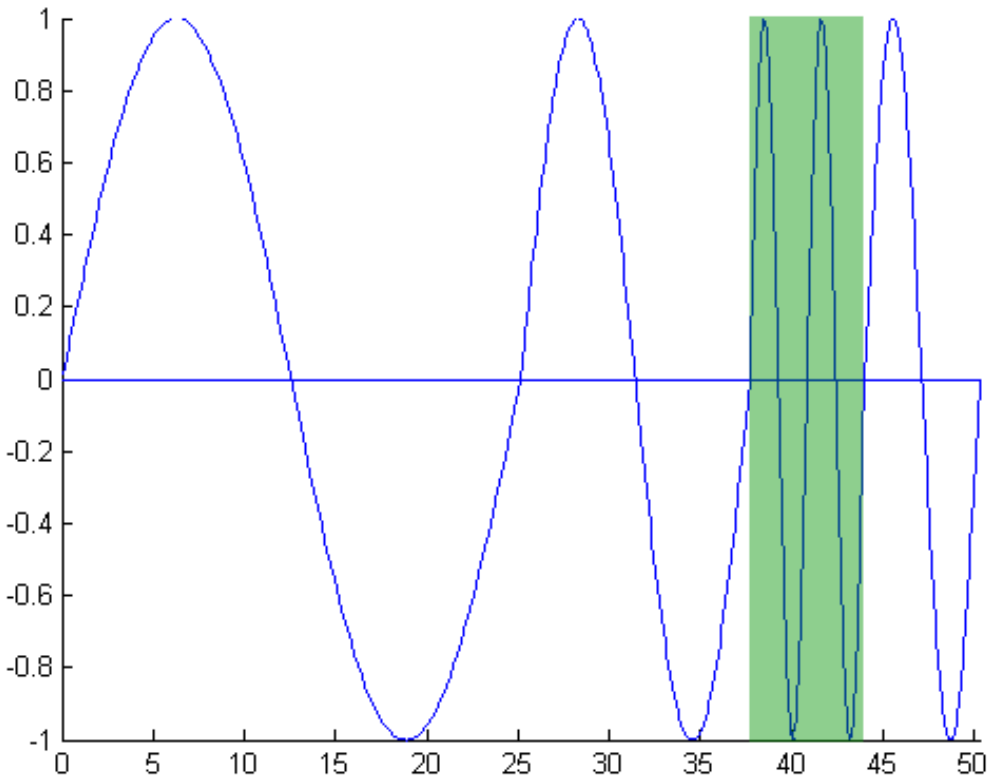


Figure: Frequency = f1, value = 1

FHSS

Example



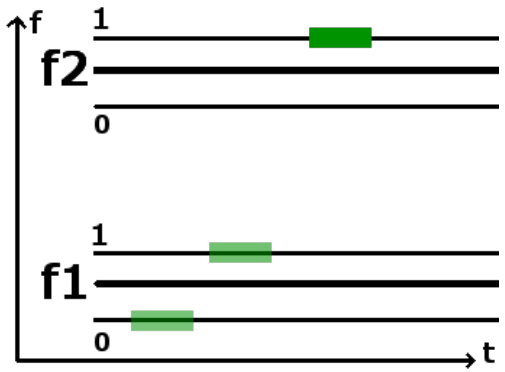
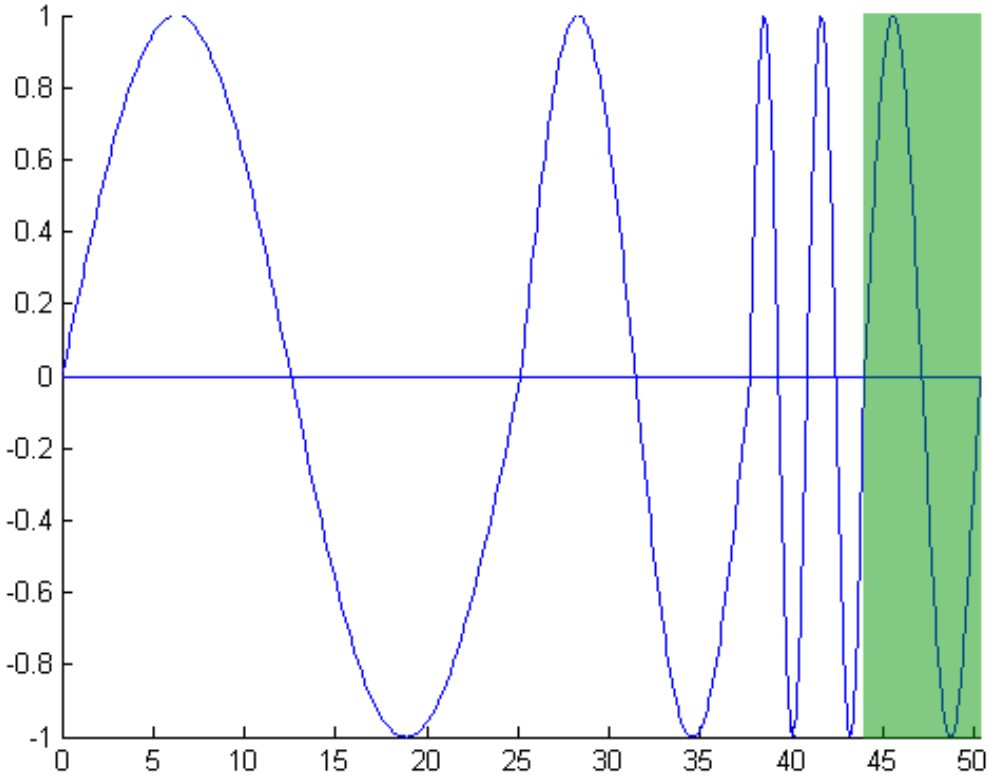


Figure: Frequency = f2, value = 1

FHSS

Example



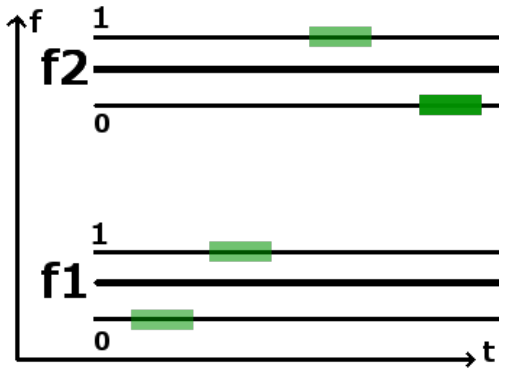


Figure: Frequency = f2, value = 0

Interference

- Robust with *narrow-band* interference, since transmission remains only for a few symbols on a frequency
- Problems with *broad-band* interference remain
- Other users will be perceived as narrow-band interference

Interference

- Robust with *narrow-band* interference, since transmission remains only for a few symbols on a frequency
- Problems with *broad-band* interference remain
- Other users will be perceived as narrow-band interference

Interference

- Robust with *narrow-band* interference, since transmission remains only for a few symbols on a frequency
- Problems with *broad-band* interference remain
- Other users will be perceived as narrow-band interference

Architecture

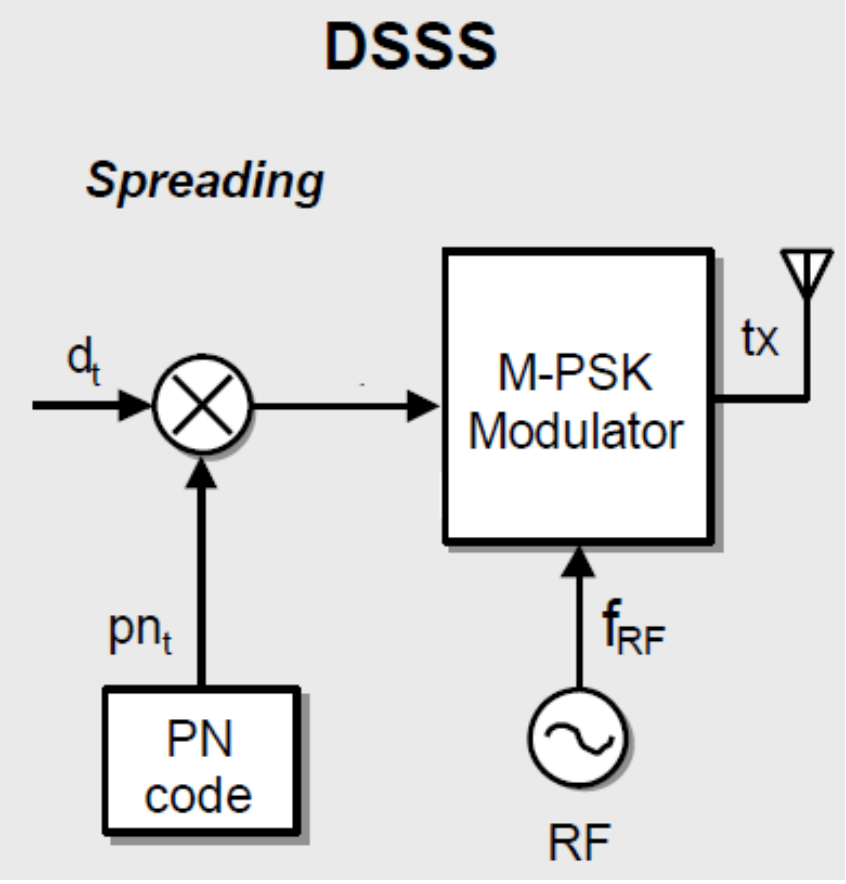
- Object-oriented Matlab
- Modulation using Communications System Toolbox

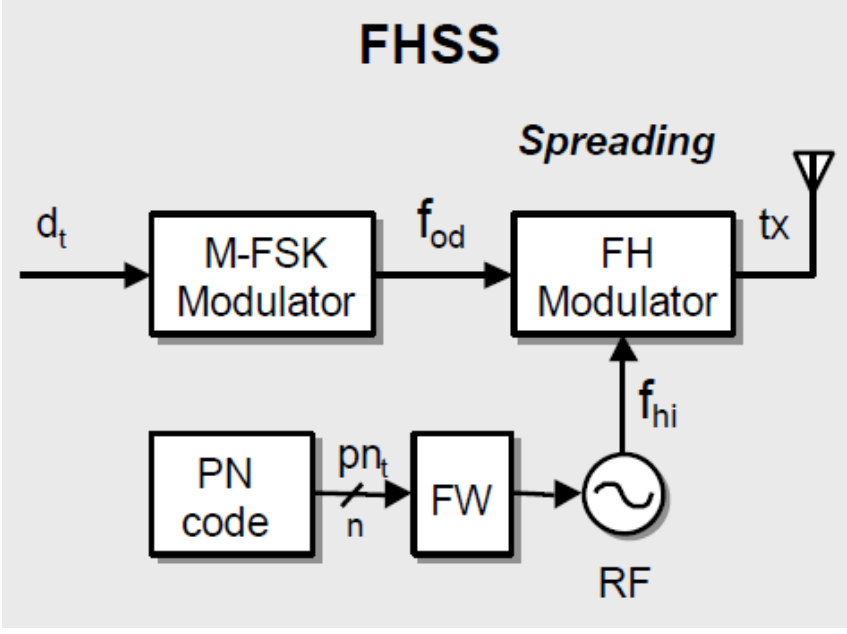
Architecture

- Object-oriented Matlab
- Modulation using Communications System Toolbox

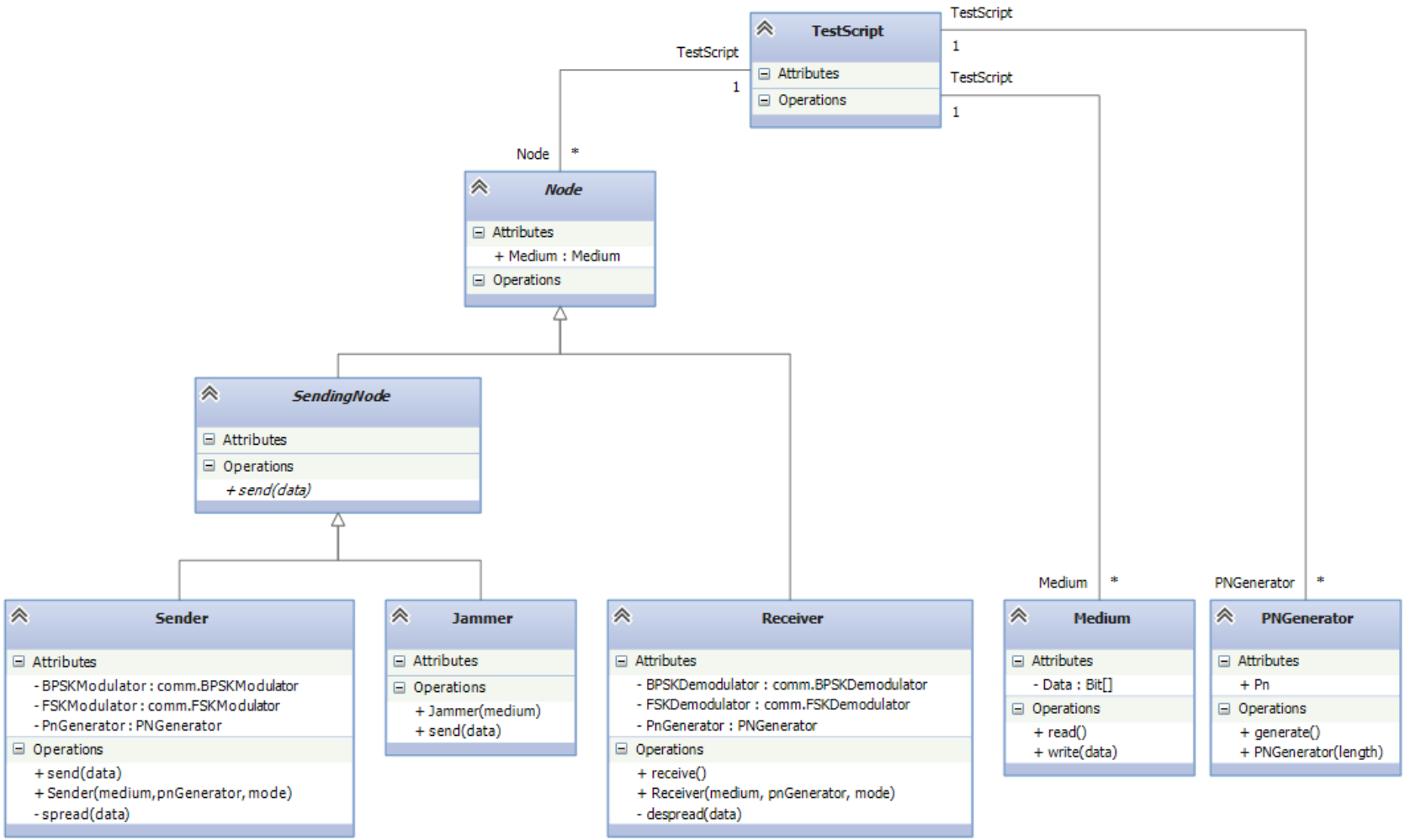
Architecture

- Object-oriented Matlab
- Modulation using Communications System Toolbox





UML


TestScript
Attributes
Operations
TestScript
1
TestScript
1
TestScript
1
Node
*
Node
Attributes
+ Medium : Medium
Operations
SendingNode
Attributes
Operations
+ send(data)
Medium
*
PNGenerator
*
Sender
Attributes
- BPSKModulator : comm.BPSKModulator
- FSKModulator : comm.FSKModulator
- PnGenerator : PNGenerator
Operations
+ send(data)
+ Sender(medium,pnGenerator, mode)
- spread(data)
Jammer
Attributes
Operations
+ Jammer(medium)
+ send(data)
Receiver
Attributes
- BPSKDemodulator : comm.BPSKDemodulator
- FSKDemodulator : comm.FSKDemodulator
- PnGenerator : PNGenerator
Operations
+ receive()
+ Receiver(medium, pnGenerator, mode)
- despread(data)
Medium
Attributes
- Data : Bit[]
Operations
+ read()
+ write(data)
PNGenerator
Attributes
+ Pn
Operations
+ generate()
+ PNGenerator(length)

Simulation

- DSSS:
  - Phase modulation - BPSK modulation scheme
- FHSS:
  - Frequency modulation - FSK modulation scheme
- Add interferences and noise on the medium
  - Gaussian noise
  - Broadband noise
  - Narrow band noise
- Different chip rates in FHSS - Fast & slow hopping
- Different chip sequence length

Simulation

- DSSS:
  - Phase modulation - BPSK modulation scheme
- FHSS:
  - Frequency modulation - FSK modulation scheme
- Add interferences and noise on the medium
  - Gaussian noise
  - Broadband noise
  - Narrow band noise
- Different chip rates in FHSS - Fast & slow hopping
- Different chip sequence length

Simulation

- DSSS:
  - Phase modulation - BPSK modulation scheme
- FHSS:
  - Frequency modulation - FSK modulation scheme
- Add interferences and noise on the medium
  - Gaussian noise
  - Broadband noise
  - Narrow band noise
- Different chip rates in FHSS - Fast & slow hopping
- Different chip sequence length

Simulation

- DSSS:
  - Phase modulation - BPSK modulation scheme
- FHSS:
  - Frequency modulation - FSK modulation scheme
- Add interferences and noise on the medium
  - Gaussian noise
  - Broadband noise
  - Narrow band noise
- Different chip rates in FHSS - Fast & slow hopping
- Different chip sequence length

Simulation

- DSSS:
  - Phase modulation - BPSK modulation scheme
- FHSS:
  - Frequency modulation - FSK modulation scheme
- Add interferences and noise on the medium
  - Gaussian noise
  - Broadband noise
  - Narrow band noise
- Different chip rates in FHSS - Fast & slow hopping
- Different chip sequence length

Metrics

- Bit-error rate
- Packet-error rate

Metrics

- Bit-error rate
- Packet-error rate

Optional goals

- Error distribution
- Multiple sender/receiver
- Free space loss

Optional goals

- Error distribution
- Multiple sender/receiver
- Free space loss

Optional goals

- Error distribution
- Multiple sender/receiver
- Free space loss

Roadmap

- Basic structure is done
- Next steps:
  - Implement spreading & despreading
  - Add noise and inteferences
  - Collect & analyze results

GitHub:
https://github.com/MobileComGroup6/MobileCom

Roadmap

- Basic structure is done
- Next steps:
  - Implement spreading & despreading
  - Add noise and inteferences
  - Collect & analyze results

GitHub:
https://github.com/MobileComGroup6/MobileCom

Roadmap

- Basic structure is done
- Next steps:
  - Implement spreading & despreading
  - Add noise and inteferences
  - Collect & analyze results

GitHub:
https://github.com/MobileComGroup6/MobileCom

Roadmap

- Basic structure is done
- Next steps:
  - Implement spreading & despreading
  - Add noise and inteferences
  - Collect & analyze results

GitHub:
https://github.com/MobileComGroup6/MobileCom

Roadmap

- Basic structure is done
- Next steps:
  - Implement spreading & despreading
  - Add noise and inteferences
  - Collect & analyze results

GitHub:
https://github.com/MobileComGroup6/MobileCom