LoRa receiver in MATLAB

LoRa receiver SDR implementation in MATLAB using the RTL-SDR dongle as a receiver.

SDL team:
Guillermo Ortas 4612957
Abdul Aziz Hamad 4621565
Kumar Navneet 4619110

TU Delft

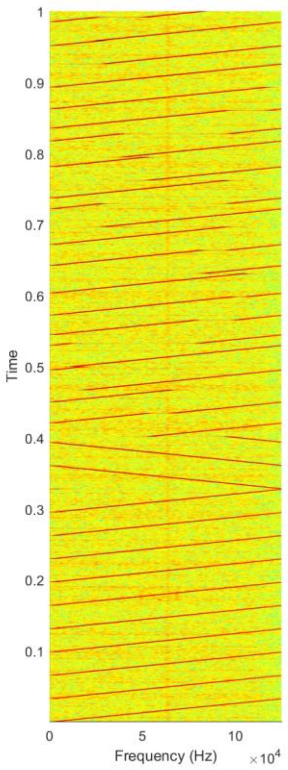
Implementation

Two main blocks:

- Demodulation
- Decoding

TU Delft

Signal modulation



- MFSK modulation on top of a chirp signal.
- The chirp rate is $\frac{BW}{2^{SF}}$, inverse of the symbol time.
- Signal bandwidth is either 125 kHz, 250 kHz or 500 kHz.
- Spreading Factor ranges from 6 to 12 bits (# bits/symbol)

- The preamble/training sequence: 10 up-chirps (the last two are a special sync word), corresponding to the value zero.
- Start of Frame Delimiter (SFD): 2.25 down-chirps, immediately followed by the data.

Channelization and re-sampling

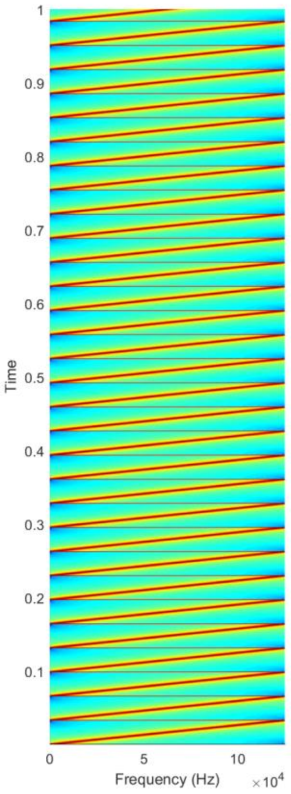
The signal is channelized using a Digital Down Conversion (DDC) technique.

1. Bring signal to base-band
2. Low pass filter
3. Resample to match the chirp bandwidth: $F_s' = BW$ (decimation)

This dramatically decreases the computational load required to process the signal (less samples).

TU Delft

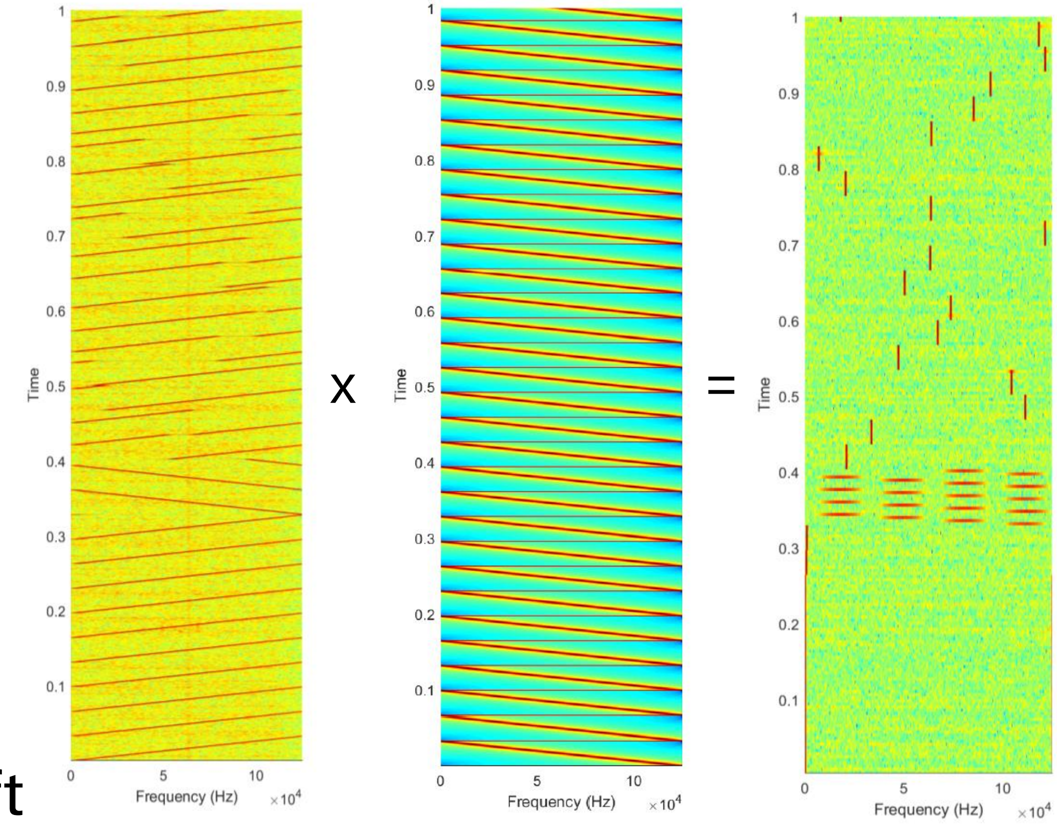
De-chirping process



- Locally generating chirp:
  - Chirp rate matching the incoming signal
- Multiplying the incoming signal with the complex conjugate of the locally generated chirp.
- The resulting signal is equivalent to a MFSK modulation with $M = 2^{SF}$ levels.

TUDelft

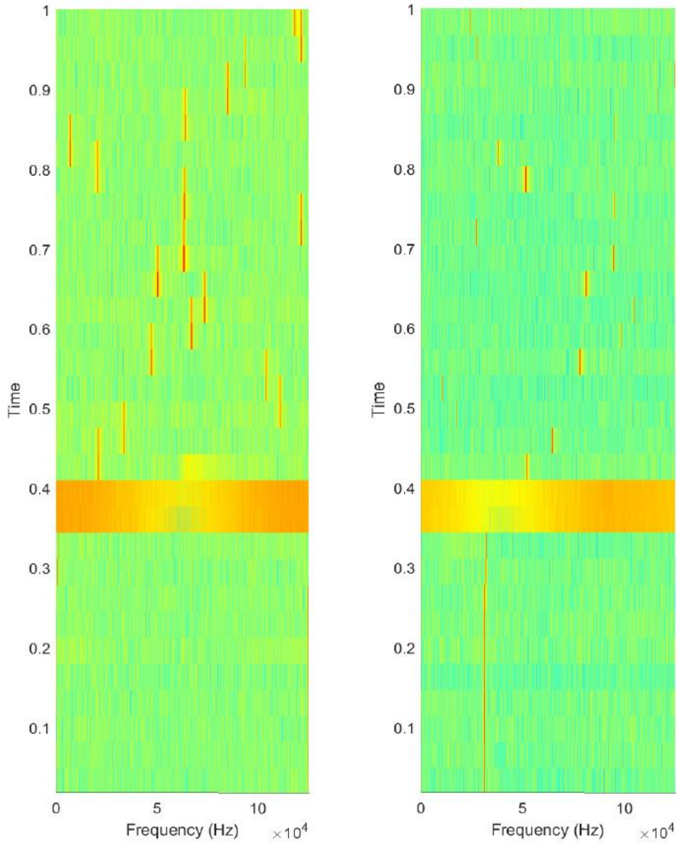
De-chirped signal



- Spectrogram with adjusted parameters to make the symbols easier to see
- The horizontal lines correspond to the SFD (more spread)

Synchronization and alignment

- The start of the signal is identified with a correlation function
  - Good correlation properties of chirps
- Symbols are aligned to the data segment



Extraction of symbols and bits

- Spectrogram set up to return one sample per symbol per frequency bin (as in last slide)
- A symbol corresponds to the frequency index of the strongest spectral component for each symbol time

The script now returns a string of raw bits

Decoding block

- The next step is to decode the raw bits to get actual data.
- There are four different stages for decoding:
  - Grey Coding
  - Whitening sequence
  - De-interleaving
  - Forward error correction

Grey indexing

- It is equivalent to Grey code
- Convert grey code block to binary code.
- Binary output (B) given by:

$B_1 = D_1$

$B_2 = B_1$ XOR D_1

$B_3 = B_2$ XOR $D_2 \ldots B_n = B_{n-1}$ XOR D_n

TU Delft

Removing Data Whitening

- It’s needed to know the random number used for whitening the signal
- XOR that random number with the bits B

Deinterleaving and FEC

- Hamming(N,4) encoder is used in LoRa:
  - Every 4 data bit is concatenated with N bit, with N = SF.
- Knowing two thing before we perform FEC:
  - Deinterleaving matrix
  - Error correcting code
- Fill in the bits and flip the matrix to get the data
- Use error correcting code to correct the bit

TU Delft

Thank You

TU Delft