ULB

WIRELESS COMMUNICATIONS
RESEARCH GROUP

OPERA.ULB.AC.BE

Modulation & Coding: Lab 1

Q&A session room number: S.UB4.222

Trung Hien Nguyen (trung-hien.nguyen@ulb.ac.be)
Jean-François Determe (jdeterme@ulb.ac.be)

Outline

- Preliminary
- Major parts of the lab. project
- Miscellaneous tips

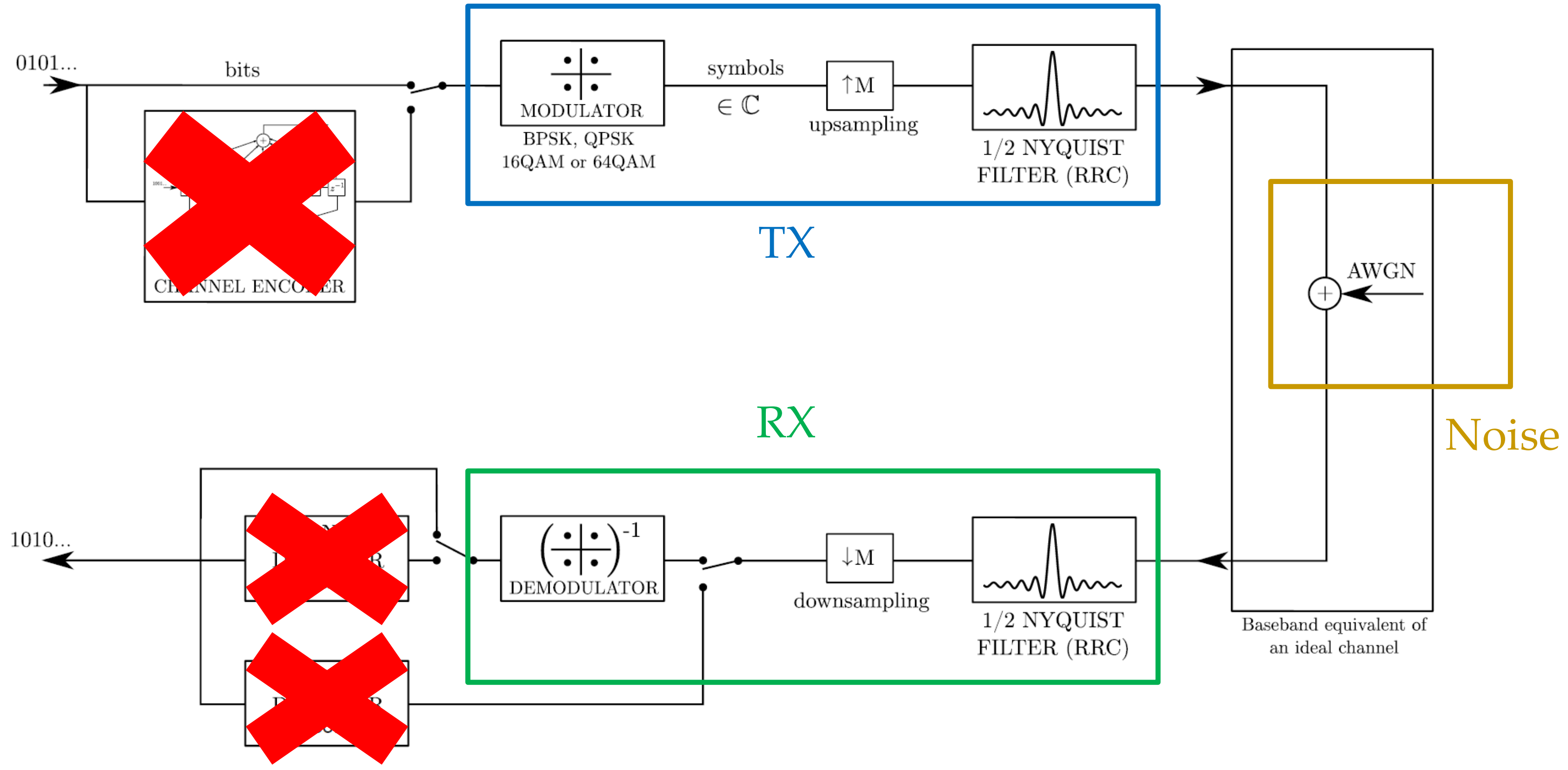
Preliminary

- Simulation tool: Matlab
- Digital Video Broadcasting-Satellite (DVB-S) communication chain
- Groups of 2 to 3 students
- Evaluations for the three parts of the project (report of max. 5 pages + oral defense for each group)
  - Figures/results + explanations and answers to questions
- Final report (20 pages) + *.zip matlab code => Mail to: fhorlin@ulb.ac.be BEFORE the deadline

Major parts of project

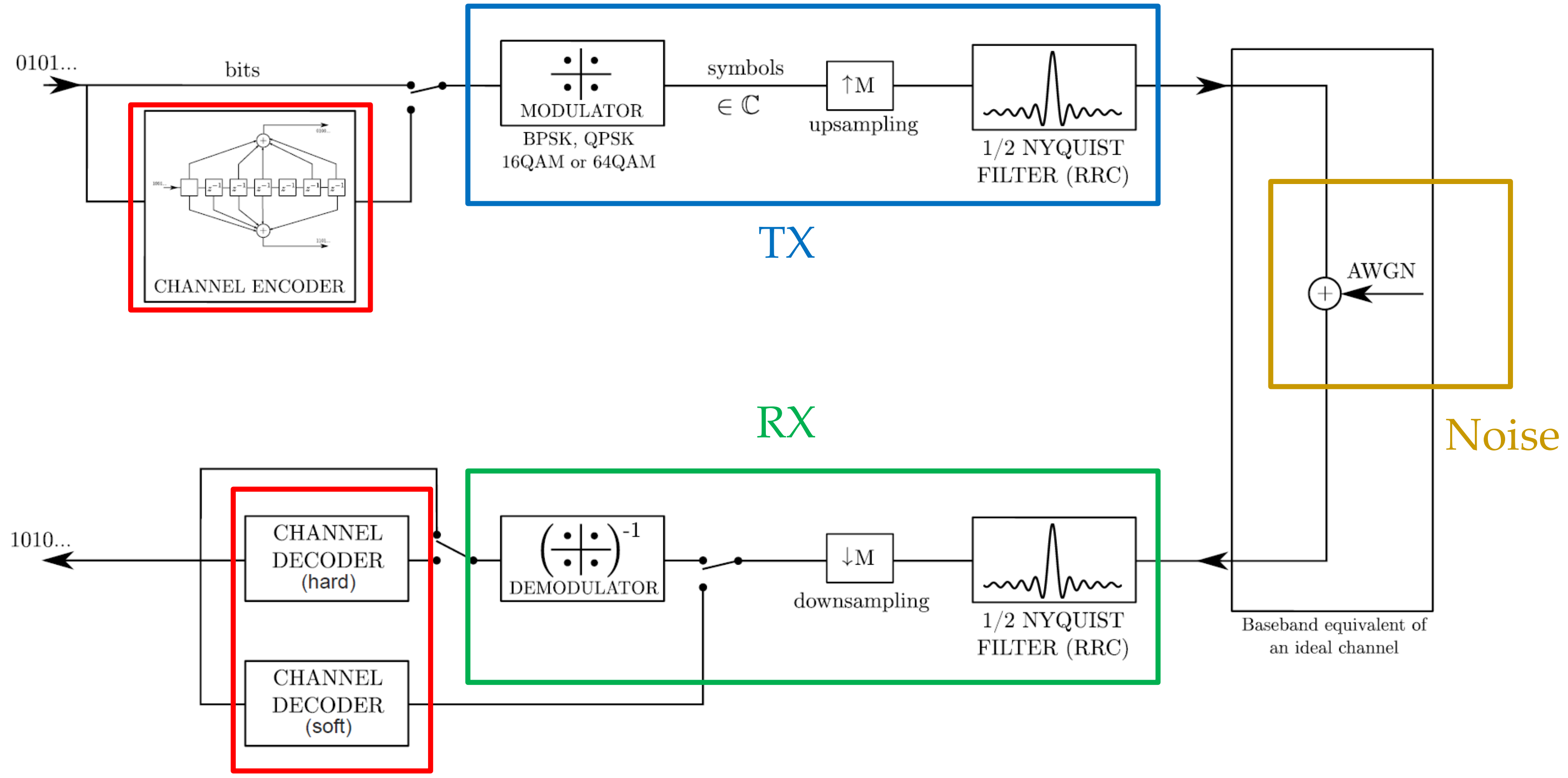
- Task 1: simulation of the optimal communication chain over idea channel
- Task 2: simulation of time/frequency synchronization algorithm
- Task 3: simulation of LDPC channel encoder and decoder
  - Hard decoding for all modulations is required
  - BPSK soft decoding is a bonus

Task 1



- **Task 1:** Implement in Matlab: TX, RX, and noise addition.

Task 2 & 3



- **Task 2:** Synchronization -> additional blocks at TX and RX.
- **Task 3:** Implement LDPC channel coding and decoding.

Tips (1)

- Functions `mapping` and `demapping` for converting bits to symbols and symbols to bits, respectively.
- The functions are provided by us and should be included in the folder from which the simulations are run

```
if (bitsPerSymbol > 1)
    txSymbols = mapping(txBits.', bitsPerSymbol, 'qam'); % Symbols at the tx
else % BPSK case
    txSymbols = mapping(txBits.', bitsPerSymbol, 'pam');
end
```

```
if (bitsPerSymbol > 1)
    rxBits = demapping(rxSymbols, bitsPerSymbol, 'qam').';
else % BPSK case
    rxBits = demapping(real(rxSymbols), bitsPerSymbol, 'pam').';
end
```

  - The input vectors for both functions should be column vectors.
  - In the BPSK case, the demapping function requires a **real** column vector.

Tips (2)

- Tips:
  - Never manually inject numbers depending on simulation parameters into your code, use variables instead (e.g. `nbBits, RRCTaps, bitsPerSymbol, EbN0`). Otherwise:
    - Difficult/Time-consuming to change the parameters afterwards in all the functions
    - Debugging made easier if parameters can be easily changed
  - Use `1i` instead of `i` or `j` to represent the complex number 0+1*i.
  - `variable.’` = transpose while `variable’` = Hermitian transpose.
  - In your reports, explicitly answer the questions of the project statement.
- Meet us at the Q&A sessions!
  - Explanations (PowerPoint) about the difficult parts of each lab.
  - More tips & practical demo on how to produce nice looking PDF figures from Matlab.