Association of Non-Binary code and CCSK modulation

Quasi Cyclic Short Packets Project

26/01/2023

Emmanuel Boutillon
Prof. Lab-STICC, Université Bretagne Sud

Kassem Saied, Camille Monière, Christian Roland, Cédric Marchand, Leonardo Monteya Oboso and all the QCSP’s consortium.
Massive IoT: first paradigm shift

Classical cellular methods are simply not adapted for massive IoT.

Solution: just suppress coordination… and accept collision [1].

Second problem: Classical model of frame is inefficient for small payload,

=> Header, Data and Redundancy should be merged.

Big bet: **new waveform for IoT** for low cost sensors, unsupervised network
Cyclic-Code Shift Keying (CCSK) used in Quasi-Zenith Satellite system (Japanese GPS enhancement system). 2003 [1]

Non-binary error correcting codes (NB-ECC) used in BeiDou (Chinese GPS-like system) 2017 [2].

QCSP Approach: CCSK modulation and NB-code association

About QCSP project

- Project funded by ANR (French Research Funding Agency).
- 4 years project started in October 2019.

Objective: Evolution of IoT frames
Outline

- QCSP system model
- Receiver processing
- GNU Radio implementation
- Rate adaptative QCSP
- Conclusion and perspectives
Outline

- QCSP system model
- Receiver processing
- GNU Radio implementation
- Rate adaptative QCSP
- Conclusion and perspectives
Cyclic Code Shift Keying modulation

\( P_0 = 11101000 \) + BPSK modulation, roll-off factor 0.35, \( q = 8 \)

- CCSK modulation:
  - \( P_0 = 11101000 \)
  - \( P_1 = 01110100 \)
  - \( P_2 = 00111010 \)
  - \( P_3 = 00011101 \)
  - \( P_4 = 10001110 \)
  - \( P_5 = 01000111 \)
  - \( P_6 = 10100011 \)
  - \( P_7 = 11010001 \)

Binary message: 011001100

Make 3-uplet symbols: \((011)_2(001)_2(100)_2\)

Take decimal value: 3 1 4

Associate CCSK symbol \( P_3 P_1 P_4 \)

Send: 000111010111010010001110
QCSP frame structure \((q = 2^m)\)

- The frame is composed of \(N\) segments of CCSK sequence (or symbol)

```
3  1  4  7  1  2
```

```
[Diagram of the QCSP frame structure showing the flow from NB Encoder to NB Decoder, q=2^m-CCSK coder, CCSK ML decoder, and the channel.]
```
QCSP frame structure \( (q = 2^m) \)

- The frame is composed of \( N \) segments of CCSK sequence (or symbol)

\[
\begin{array}{cccccc}
3 & 1 & 4 & 7 & 1 & 2 \\
\end{array}
\]

\[
\begin{array}{cccccc}
b_0 = +1 & b_1 = -1 & b_2 = -1 & b_3 = +1 & b_4 = +1 & b_5 = -1 \\
\end{array}
\]
Demodulation of CCSK frame in complex noise

Correlation between each of the received symbols $Y$ and the $q$ CCSK sequences.

$$Y = P_{19} + W$$

$$\log(\text{Prob}(Y|P_a)) \sim \text{Real}\left(\sum_{k=0}^{q-1} Y(k)P_a(k)\right)$$

$$\log(\text{Prob}(Y|P_{19})) \sim \text{Real}\left(\sum_{k=0}^{q-1} (P_{19}(k) + W(k))P_{19}(k)\right)$$

$\sim q + \text{Noise}$
Demodulation with phase offset

Effect of Doppler of local oscillators mismatch

CCSK sequences \( P_{13} \) \( e^{j\varphi} \) Complex noise

\[
\log(\text{Prob}(Y|P_a)) \sim \left| \text{Real} \left( \sum_{k=0}^{q-1} Y(k)P_a(k) \right) \right|
\]

Phase and/or frequency offset require non-coherent demodulation.
Example of performance

Perfect synchronization.

**QCSP:**
NB code rate $r \approx \frac{1}{3}$
CCSK: $m = n \cdot 6$ bits, $q = 64$
$\Rightarrow$ rate $\frac{6}{64}$.
Global rate $\approx \frac{6}{64} \times \frac{1}{3} = 0.0313$

**5G:**
rate $\frac{1}{3}$ binary LDPC code + 11 repetitions.

Global rate: $\frac{1}{33} = 0.0303$
Outline

- QCSP system model
- Receiver algorithms
- GNU Radio implementation
- Work in progress on QCSP
- Conclusion and perspectives
Receiver principle...

Signal Processing factory

Messages & Noise & Time? & Freq?

Inner Code

Over Mod.

Outer Code

Used redundancy

Coherently demodulated messages

NB decoder engine

Decoded messages
Developing blind detection and self-synchronization algorithms for achieving correct preamble-less short packet reception at very low SNRs.

Joint overall probability:
\[ P \approx P_d \times P_s \times P_c \]

Aiming to maximize the overall probability is achieved by maximizing the weakest probability:
\[ \text{Max}(\text{min}(P_d, P_s, P_c)). \]
Test in real time different time/frequency hypothesis on frame arrival.

Accumulate maximum absolute correlation between received symbols and codebook over the frame size.

Declare detection if score above a threshold.
**Toy example**

\[ P_0 = \text{ABAABB} \]

Message \( = \{0,0,0\} \ (N = 3) \)

\{\text{ABAABB, ABAABB, ABAABB}\}

\[ P_0 \quad P_0 \quad P_0 \]

Message \( = \{3,0,5\} \ (N = 3) \)

\{\text{ABBABA, ABAABB, BABAAB}\}

\[ P_3 \quad P_0 \quad P_5 \]

Channel delay

\[ \text{ABBABAABAABBBABABAAB} \rightarrow \text{XXXXABBABAAABABBBBABAABXX} \]
Toy example

\[ Y_0 \]

\[ XXXXABBAABABAABBBABAABXXX \]

\[ ABAABBB \]

\[ BAABBA \]

\[ AABBAB \]

\[ ABBABA \]

\[ BBABA \]

\[ BABAAB \]

\[ 2 \]

\[ L_0 = \{ <Y_0, P_a> \}_{a=0,1,...,5} \]

\[ \text{Max}(L_0) = 2 \]
Toy example

\[ Y_1 \]

\[
XXXABBAABAABBAABBAABABXXX
\]

- ABAABB
- BAABBA
- AABBBAB
- ABBABA
- BBABAA
- BABAAB

\[ \text{Max}(L_1) = 3 \]
Toy example

\[ Y_2 \]

\[ XXXXABBAABABAABBBABAABXXX \]

\[ \]

\[ \]

\[ --ABAABBB-- \]

\[ 2 \]

\[ --BAABBA-- \]

\[ 4 \]

\[ --AABBAB-- \]

\[ 1 \]

\[ --ABBAAB-- \]

\[ 2 \]

\[ --BBABAA-- \]

\[ 3 \]

\[ --BABAAB-- \]

\[ 0 \]

\[ \]

\[ \]

Max(\(L_2\))=4
Toy example

\[ Y_3 \]

XXX X ABBABA A ABA ABBB A B A B X X X

--- ABAABB --- ABBA BA BA AB --- ABBAB --- ABBA BA --- ABBA BA --- BA BA BA ---

1 2 5 1 3 3

\[ \max(L_3) = 5 \]
Toy example

\[ Y_4 \]

[Diagram showing various sequences and their lengths, with \( \text{Max}(L_4) = 6 \)]
Toy example

\[ Y_5 \]

XXXAABBABAABAABBBABAABXXX

----- ABAABB ---------------------------------------- 2
----- BAABBA ---------------------------------------- 4
----- AABBBAB --------------------------------------- 1
----- ABBABA --------------------------------------- 2
----- BBABAA --------------------------------------- 6
----- BABAAB --------------------------------------- 1

Max(\(L_5\))=6

234566
Toy example

\[ Y_6 \]

XXXAABBABAABAAABBBABAABXXX

\[ \begin{align*}
    &\text{ABAAB} &\quad 2 \\
    &\text{BAABBA} &\quad 2 \\
    &\text{AABBAB} &\quad 4 \\
    &\text{ABBAAB} &\quad 2 \\
    &\text{BABAAB} &\quad 2 \\
    &\text{BABAAB} &\quad 6 \\
\end{align*} \]

\[ \text{Max}(L_6) = 6 \]
Toy example

\[ Y_7 \]

\[
\begin{align*}
XXX\text{ABB}A\text{BAABAABBB}\text{BAABXXX} & \quad \text{Max}(L_7) = 5 \\
\hline
\text{ABAAB} & \quad \text{5} \\
\text{BAABBA} & \quad \text{2} \\
\text{AABBBAB} & \quad \text{1} \\
\text{ABBBABA} & \quad \text{4} \\
\text{BBABAA} & \quad \text{2} \\
\text{BABAAAB} & \quad \text{1} \\
\hline
23456665
\end{align*}
\]
Toy example

\[ Y_{13} \]

\[
\begin{align*}
XXX & \quad \text{ABBA}\text{ABAABAABBBABAABXXX} \\
\text{--} & \quad \text{ABAABB} \quad \text{----} & \quad 3 \\
\text{--} & \quad \text{BAABBA} \quad \text{----} & \quad 1 \\
\text{--} & \quad \text{AABBAB} \quad \text{----} & \quad 5 \\
\text{--} & \quad \text{ABBBAB} \quad \text{----} & \quad 3 \\
\text{--} & \quad \text{BBABAA} \quad \text{----} & \quad 3 \\
\text{--} & \quad \text{BAABAB} \quad \text{----} & \quad 3 \\
\text{2345666555654556543210000} \\
\text{Max}(L_{13}) = 5
\end{align*}
\]
Toy example

Score(k) = \max(L_k) + \max(L_{k+6}) + \max(L_{k+12})

Score(4) = \max(L_4) + \max(L_{10}) + \max(L_{16}) = 18
Toy example

Score(k) = \text{max}(L_k) + \text{max}(L_{k+6}) + \text{max}(L_{k+12})

Score(6) = \text{max}(L_6) + \text{max}(L_{12}) + \text{max}(L_{18}) = 14
Detection problem: general case

\[ S_k(Y) = \sum_{n=0}^{4} \max(|L_{k+64n}|) \]

\( N = 5, \ q = 64, \) no noise

\[ \max(|L_k|) \]
Detection problem: general case

\[
S_k(Y) = \sum_{n=0}^{4} \max(|L_{k+64n}|)
\]

\[S_k(Y) = \sum_{n=0}^{4} \max(|L_{k+64n}|)\]

N = 5, q = 64, with noise (-4 dB)

max(|\(L_k|\))

Detection

Threshold
Score values in 3D grid where $N = 60$, $q = 64$ and a frequency offset.

Impact of the frequency offset

Requires a 2D dimensionnal search
Overall performance

The QCSP parameters we choose to work on: $N = 120$ symbols, $q = 64$, $R_c = 1/2$

Asynchronous AWGN channel

![Diagram of signal processing](image)

Theoretical Lower bound [1,2]


Time, Frequency & Phase synchronization

- Use information given by overmodulation to mitigate time ambiguity at CCSK symbol level [1]
- Use redundancy of outer code to mitigate time ambiguity at the chip level [1]
- Use soft information given by inner code decoding + redundancy of the outer code to estimate residual Frequency&Phase ambiguity [2].
- Perform coherent demodulation before NB-Decoder

The QCSP parameters we choose to work on: $N = 120$ symbols, $q = 64$, $R_c = 1/2$

Asynchronous AWGN channel

---


Outline

1. QCSP system model
2. Receiver processing
3. GNU Radio implementation
4. Work in progress
5. Conclusion and perspectives
Offline experimental set-up

Note: Indoor test from 2 different rooms

- GNU Radio parameters:
  - $F = 433.950$ MHz
  - Chip rate = 500 kHz
  - Bit information rate $0.5 \times 6/64 \times 1/3 = 15.6$ Kbit/s
Super-Frame structure

- Super Frame (SF) composed of 61 QCSP frames
- -1 dB of energy between two consecutive frames.
- QCSP frame: $k = 120$ bits.
Received signal

Theoretical limit:
FER = $10^{-4}$ at -11.5 dB

$Es/N_0 = 29.5$ dB
$\Delta F = 4006$ Hz
No error before NB-decoder

$Es/N_0 \approx -12.5$ dB
$\Delta F = 4012$ Hz
35% of uncorrectly detected symbols before NB-decoder.
Real-time experimental set-up: Emission

Payload: $k = 120$ bits that contains
- Localization given by GPS module
- + Time, RPi4 Temperature and a frame number.

New message every 4 seconds
Radio Settings in Transmission
- Measured RF emission Power: 4.5 dBm
- Carrier Frequency: $F_c = 433.950$ MHz

Bit information rate: 3.9 Kbit/s
Message duration: 32 ms.
Path in Lorient

Except for residents
Result of the experimentation

Interactive map: https://qcsp.univ-ubs.fr/events/
Dec. 2022: Proof of communication at 433 Mhz from a buoy to a boat with significant distance. March 2023: Next round of experimentation.
**Multi-users detection: first results**

- 2 orthogonal CCSK sequences PN1, PN2.
- Random time of emission, random frequency of emission within \([F_{\text{min}}, F_{\text{max}}]\), random position of emission.

\[
F = F_0 \pm 12.5\% \text{ of signal bandwidth}
\]
Successive Interference Cancellation

- Layer by layer, you get the very center of the onion.
- SIC decoder is (almost) the same principle
Input signal in time

\[ y(n) = s(n) + w(n) \]

Noise \( w(n) \)

Signal \( s(n) \)

45 frames received
Output of detection match filter

CCSK1

8 trames decoded

CCSK2

6 trames decoded

Detected/corrected

Not detected

Detected but not decoded
SIC receiver: second iteration

\[ y(n) = s(n) + w(n) \]

- Detected trames are subtracted to the incoming signal
  => Then detection done again.
SIC receiver: second iteration

+8 trames decoded

+7 trames decoded

Total: 14 (first iteration) + 15 (second iteration) = 29 decoded frames among 45
State at the end of fourth iteration

Only 3 remaining frames undecoded among 45
=> 93 % of decoding success!
QCSP for rate-adaptive code

- Variation on channel conditions
  - Optimized Modulation/code for each channel.
  - Complexity => Cost.

- State of the art solution
  - Probabilistic shaping

- Other Solution
  - One NB decoder optimized for a single code rate
  - One Inner code responsible of rate matching

> The Truncated CCSK (presented in ISTC’2021)
> The Truncated Non Binary-CCSK (new results)
**Inner Truncated CCSK code: principle**

- **k symbols**
  - $k.m$ bits

- **n symbols**
  - $n.m$ bits

- **q chips**
  - $00011, 11010110010$

- **p chips**
  - $00011$

**CCSK**

**NB code**

- $\alpha^0, \alpha^3, \alpha^7, \alpha^2$

**Outer code**

- $r_o = k/n$

**Spreading**

- $r_s = m/q$

**Truncation**

- $r_t = q/p$

**Inner code**

- $r_i = m/p$

**Spectral efficiency**

- $R = r_o \cdot r_i$ bit/s/Hz

---

Inner TCCSK code: performance, $k = 120$ bits, GF(64)

Performance with 64-APSK-CCSK sequence, \( q = 64, k = 120 \) bits.
Outline

QCSP system model

Receiver algorithms

GNU Radio implementation

Work in progress

Conclusion and perspectives
Conclusion

- This work has required...
Signal processing theory

Each output $O_m$ of parity check can be calculated as:

$$O_m^z = \frac{I_1^x(x) \cdot h_m \cdot v_x + I_1^y(y) \cdot h_m \cdot v_y}{h_m \cdot v_z}$$

where $m \in [0, N - K - 1]$.

$$P_0/R\text{-code} = \left(-f_0 \Psi_0 + Z_k\right)^q$$

No parity checks fulfilled and $I_k \neq \Xi \neq O_k \neq \Omega_k$.

$$H_2(U|Y) = \mathbb{E}_Y \left[ -\sum_{s \in \mathbb{Z}} L(s) \left( \log_q (L(s))^2 \right) \right], \quad (37)$$

$$L_k \Psi_k(\Delta k - \Delta) = e^{j\Psi_k} \sum_{n=0}^{\Delta - 1} e^{j2\pi f_0 n} = e^{j\Psi_k} \left( \frac{\sin(\pi f_0 \Delta n)}{\sin(\pi f_0)} \right), \quad (21)$$

$f_{M_k}(x) = \frac{2x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \left[ 1 - e^{-\frac{x^2}{2\sigma^2}} \right]^{q-1}.$ \quad (33)
```
function ToRun2(searchmethod, n_searchsPW, PNs_group, alllambdaetsnrs, threshold, top_notDecoded, n_simulations)

% Fixed lambda : ToRun2(1, 1, 1, [], 12, -10], 145, 100, 1);
% All lambdas  : ToRun2(1, 1, 1, 0, 145, 100, 1);

global LAMBDA

% Case for fixed SNR range and lambda values
if ~isequal(alllambdaetsnrs(1), 0)
    OneLambda2iterate(searchmethod, n_searchsPW, PNs_group, alllambdaetsnrs, threshold, top_notDecoded, n_simulations);
    return;
end
```
Software: take maximum profit of all feature of the CPU (SIMD, multi-thread).

Hardware: efficient VHDL implementation on hardware.
Heterogeneous system integration

Hybrid real time receiver with hardware&software resources.

(LINUX)

Legend

- RFNoC Blocks
- C/C++ Code
- MATLAB/Octave
- On the USRP
- On the PC

Step 3 — 25/01/2022

- RF Tasks
- FIR
- Detection
- Time Sync.
- Freq. Sync.
- Phase Sync.
- ECC Decoder

Step 4 — Work In Progress

- RF Tasks
- FIR
- Detection
- Time Sync.
- Freq. Sync.
- Phase Sync.
- ECC Decoder

Step 7 (within a year?)

- RF Tasks
- FIR
- Detection
- Time Sync.
- Freq. Sync.
- Phase Sync.
- ECC Decoder
Experimentations
Conclusion

- Take away on QCSP frames:
  - Very low complexity at emitter side
  - Close to theoretical limit in Gaussian channel
  - Proved efficient in several channels
Perspectives

- Improved hardware/software implementation of the receiver

- Multi-users detection with SIC
  - Ongoing work for an average of 8 users in parallel.
  - Extension to 64 users in parallel.

- Application for adaptive coding rate.

- From academics work to applications...
Potential applications

◊ Space (IoT – Low Earth Orbiter Satellite)

◊ Sea (Buoy-Boat)

◊ Gas/water Meters network

◊ Many others...
Thank you!

qcsp.univ-ubs.fr/