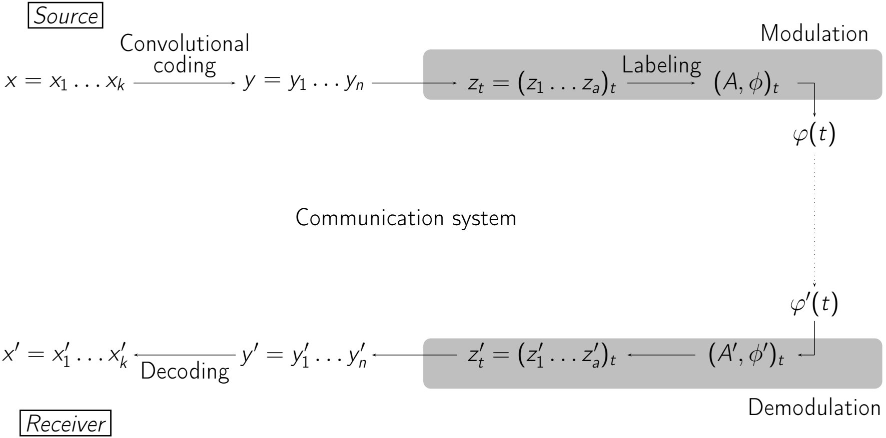
Reconstruction of Constellation Labeling with Convolutional Coded Data

Marion Bellard   Nicolas Sendrier

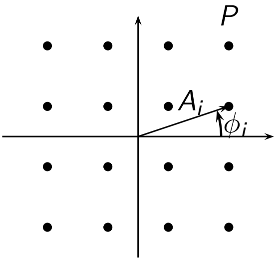
INRIA-Rocquencourt, SECRET Project-Team

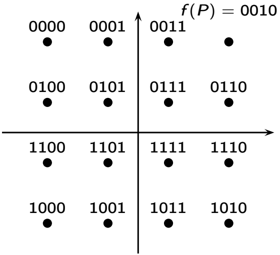
October 2012


Source
Convolutional coding
$x = x_1 \dots x_k$
$y = y_1 \dots y_n$
Modulation
$z_t = (z_1 \dots z_a)_t$
Labeling
$(A, \phi)_t$
$\varphi(t)$
Communication system
$\varphi'(t)$
$x' = x'_1 \dots x'_k$
Decoding
$y' = y'_1 \dots y'_n$
$z'_t = (z'_1 \dots z'_a)_t$
$(A', \phi')_t$
Demodulation
Receiver

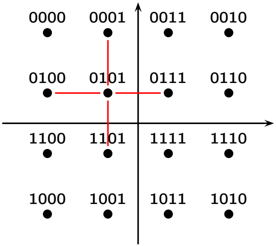
Labeling and Modulation

Labeling : $P \in \mathcal{C} \to f(P) \in \mathbb{F}_2^a$
Constellation : Representation of the labeling in the plane





Gray labeling

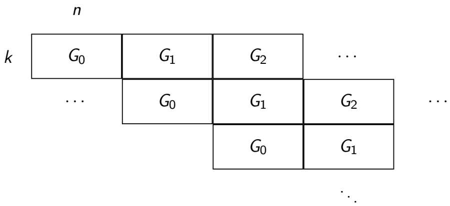


f is a Gray labeling if for all P_1 and $P_2 \in \mathcal{C}$ such as $d(P_1, P_2) = 1$,

$$d_{Hamming}(f(P_1), f(P_2)) = 1$$

1. Communication System
2. **Convolutional codes**
3. Reconstruction of Constellation Labeling
   - Method
   - Classes
   - Convolutional code reconstruction
   - Complexity
   - Further work

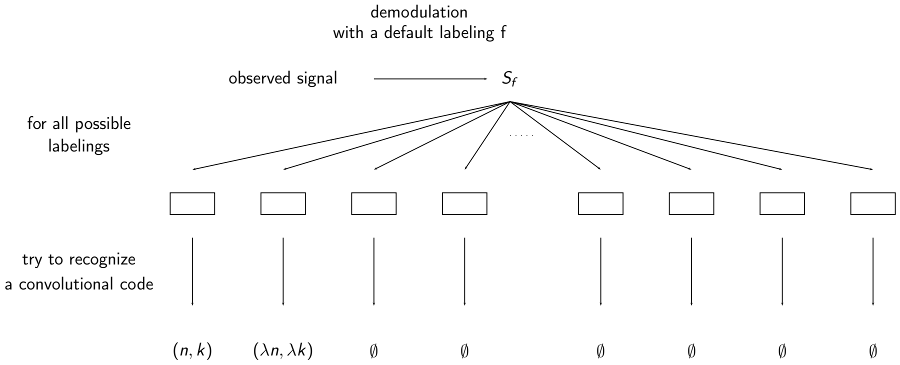
Convolutional codes



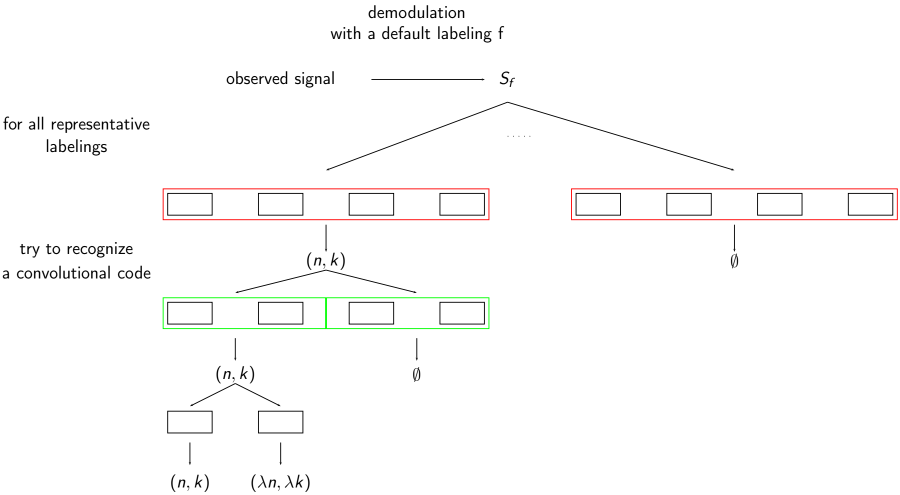
k : number of inputs
n : number of outputs

The output $(y_1 \dots y_n)_t$ depends on $(x_1 \dots x_k)_t$, $(x_1 \dots x_k)_{t-1}$, $\dots$, $(x_1 \dots x_k)_{t-M}$

Exhaustive method



Our method


demodulation
with a default labeling f
observed signal
S_f
for all representative labelings
try to recognize a convolutional code
(n, k)
$\emptyset$
(n, k)
$\emptyset$
(n, k)
$(\lambda n, \lambda k)$

Linear and affine classes

- Notation :
  - $\mathcal{C}$ a constellation, a the number of bits per symbol.
  - $C_L(f)$ the linear class of f, $C_A(f)$ the affine class of f

Definitions

- We say that two labelings f_1 and f_2 from $\mathcal{C}$ to $\{0,1\}^a$ are linearly equivalent if and only if there exists $\mathcal{L}$ a binary invertible $a \times a$ matrix such as for all $P \in \mathcal{C}$, $f_1(P) = f_2(P) \cdot \mathcal{L}$.
- We say that two labelings f_1 and f_2 from $\mathcal{C}$ to $\{0,1\}^a$ are affine equivalent if and only if there exists $\mathcal{L}$ a binary invertible $a \times a$ matrix and v a binary vector of length a such as for all $P \in \mathcal{C}$, $f_1(P) = f_2(P) \cdot \mathcal{L} + v$.

Labeling distribution

Table: Number of labelings $a = 2, 3, 4$

a	Number of labelings
2	24
3	40 320
4	$> 2 * 10^{13}$

Table: Labelings distribution for $a = 2, 3, 4$

a	(1)	(2)	(3)
2	1	4	6
3	30	8	168
4	64 864 800	16	20 160

(1) Number of affine classes
(2) Number of linear classes per affine class
(3) Number of labelings per linear class

Gray labelings

The "Grayness" of the labelings is compatible to an extent with the linear and affine equivalence relations

Property

Let f_1 be a Gray labeling and let $f_2 \in \mathcal{C}_A(f_1)$, then f_2 is a Gray labeling if and only if there exists $\mathcal{L}$ a permutation matrix and v a vector of $\{0,1\}^a$ such as $\forall P \in \mathcal{C}$, $f_2(P) = f_1(P) \cdot \mathcal{L} + v$.

Gray labelings

Table: Number of Gray labelings for square constellations $16-QAM$, $64-QAM$ and $256-QAM$

a	Number of Gray labelings
4	384
6	414 720
8	584 674 836 480

Table: Gray labelings distribution for $a = 4, 6, 8$

a	(1)	(2)	(3)
4	1	16	24
6	9	64	720
8	56 644	256	40 320

(1) Number of affine classes containing Gray labelings
(2) Number of linear classes per affine class
(3) Number of Gray labelings per linear class

Representatives of classes

- Representatives of affine classes : There exist $f_1, \ldots, f_N$ such as

$$\bigcup_{i=1..N} C_A(f_i) = \mathcal{M}_a$$

and

$$f_i \notin C_A(f_j)$$

- We find them :
  - Not Gray : by fixing the values of several points of $\mathcal{C}$ and carry out a backtrack search
  - Gray : with the direct product of two Gray codes (Wesel, Liu, Cioffi, Komminakis)
- Representatives of linear classes :

$$\bigcup_{v \in \mathbb{F}_2^a} C_L(f_i + v) = C_A(f_i)$$

Convolutional code reconstruction

n : code length

k : code dimension

S_f : the binary sequence observed with labeling f

- $R(S_f) = (n, k, G)$ is the result of convolutional code reconstruction

Test on linear and affine classes

- $(f \ \in \ C_L(f')) \Rightarrow (R(S_f) = \emptyset \Leftrightarrow R(S_{f'}) = \emptyset)$
- $S \ll \delta$ the sequence S shifted by $\delta \geq 0$ left positions and $D_\delta(S) = S - (S \ll \delta)$.

  if δ is a multiple of $\mathrm{lcm}(n, a)$ we have :
  $(f' \ \in \ C_A(f)) \Rightarrow (R(D_\delta(S_f)) = \emptyset \Leftrightarrow R(D_\delta(S_{f'})) = \emptyset)$

Test on linear class

Property

$$(f \in C_L(f')) \Rightarrow (R(S_f) = \emptyset \Leftrightarrow R(S_{f'}) = \emptyset).$$

Let S_f be a binary sequence produced by an (n, k, m) convolutional encoder using the labeling f

Let $\mathrm{lcm}(a, n) = \lambda n = \mu a$.

We have $\forall P \in \mathcal{C},\ f'(P) = f(P) \cdot \mathcal{L}$

Note that $R(S_f) = (\lambda n, \lambda k, G^{[\lambda]}) \Rightarrow R(S_{f'}) = (\lambda n, \lambda k, G^{[\lambda]}\mathcal{L}^{[\mu]})$

Test on affine class

- $(f' \in C_A(f)) \not\Rightarrow (R(S_f) = \emptyset \Leftrightarrow R(S_{f'}) = \emptyset)$
- $S \ll \delta$ the sequence S shifted by $\delta \geq 0$ left positions and $D_\delta(S) = S - (S \ll \delta)$.

Property

If δ is a multiple of $\mathrm{lcm}(n, a)$ we have

$$(f' \in C_A(f)) \Rightarrow (R(D_\delta(S_f)) = \emptyset \Leftrightarrow R(D_\delta(S_{f'})) = \emptyset).$$

Proof : $\forall P \in \mathcal{C}, f'(P) = f(P) \cdot \mathcal{L} + v$
δ multiple of a : the affine part v cancels with the difference.
δ multiple of n : the sequence $D_\delta(S) = S - (S \ll \delta)$ remains a sequence from a convolutional encoder

Complexity

N : the number of affine classes
We denote $C_A(x_1), \ldots, C_A(x_s)$ the affine classes selected, n_i the number of linear classes selected in $C_A(x_i)$.

Number of code reconstruction :

- Non gray : $N + s2^a + (n_1 + n_2 + \cdots + n_s)\#GL(a, \mathbb{F}_2)$
- Gray : $N + s2^a + (n_1 + n_2 + \cdots + n_s)a!$

Further work

- Use the relation between linear classes :
  Let $f_1 \in C_A(f)$ be a linear representative such as $R(S_{f_1}) = (G^{[\lambda]}, \lambda n, \lambda k)$. Then
  $R(S_{(f_1+v)\cdot\mathcal{L}}) = R(S_{f_1\cdot\mathcal{L}}) = (G^{[\lambda]}\mathcal{L}^{[\mu]}, \lambda n, \lambda k)$
- Use the ratio $\frac{k}{n}$ to select the best affine classes
  Example : $(n, k) = (3, 1)$ we can find $(3, 2)$ codes

Thanks for your attention
Any questions ?