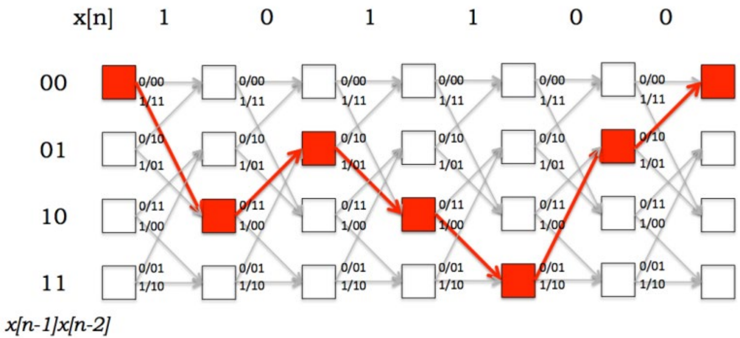


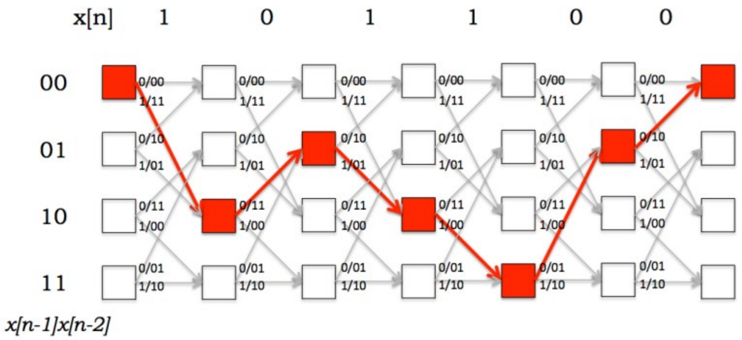
main (mit.edu)

Project 11

Backscatter Tag Design for High Concurrency (Software)

Group Members: Yuchen Ouyang, Haoyan Zhang

Teaching Assistant: Mingqi Xie



main (mit.edu)

Project 11
Backscatter Tag Design for High Concurrency

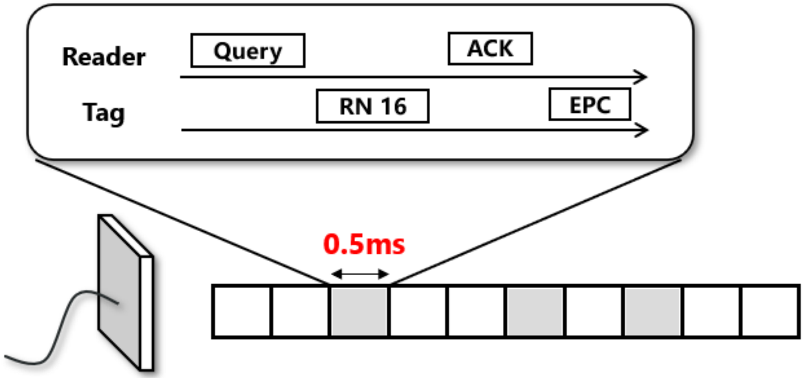
- **Background**
- Theory
- Algorithm & Optimization
- Performance
- Summary

1. Background

1. Background - Current Limitations

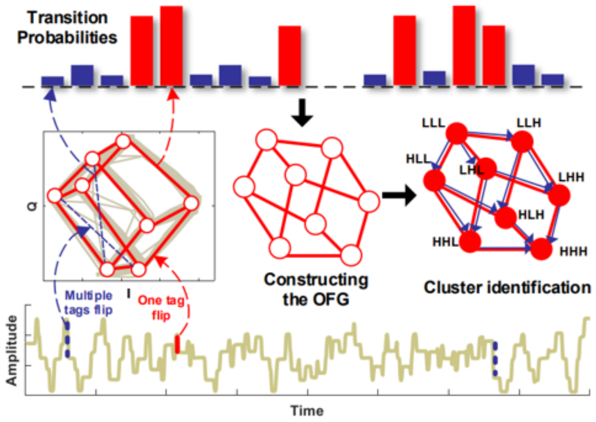
- Low efficiency
- Collision leads to a 70% waste of time.
- Low concurrency
- 5 tags at most.

TDMA Protocol:



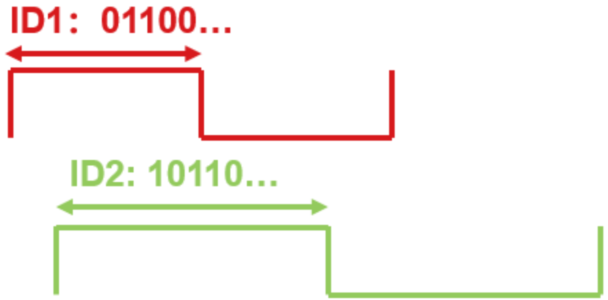
Parallel Decoding:

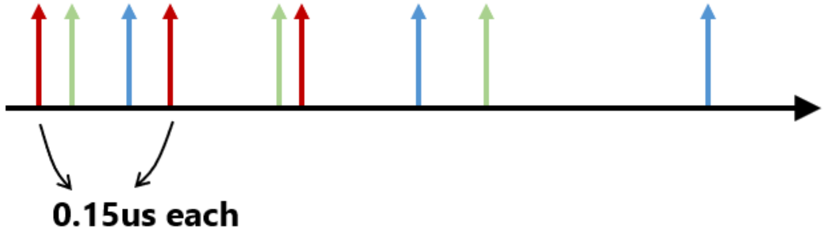
FilpTracer (Mobicom'17)

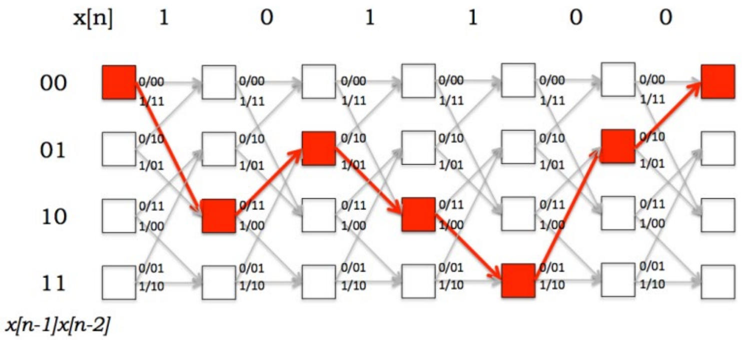


1. Background - Tag Design

- Tag ID
  - Rectangular waveforms.
  - Different frequency.
- Edge detection
  - 100x more than before, theoretically.







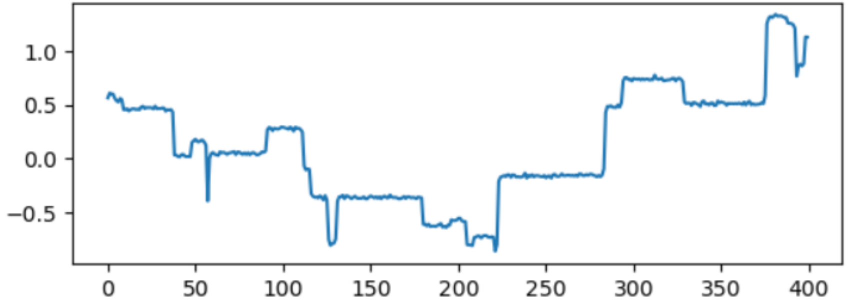
main (mit.edu)

Project 11
Backscatter Tag Design for High Concurrency

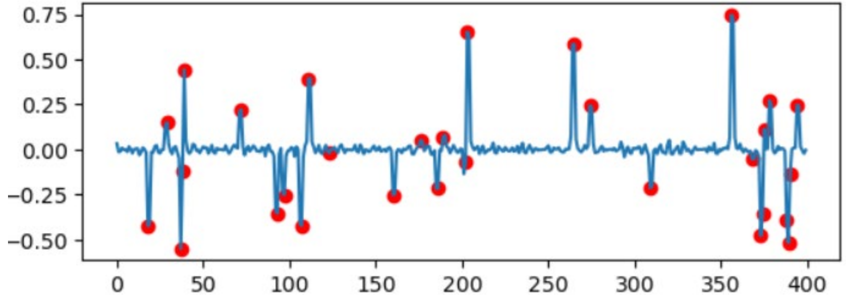
- Background
- **Theory**
- Algorithm & Optimization
- Performance
- Summary

2. Theory - Edge Extraction

- Received signal: Sum of square waveform + Gaussian noise



- Edges extraction: shift, subtraction



(real part)

What we get: **Index in original signal + its corresponding edge amplitude**

2. Theory - Classification

Goal: Given a sequence of extracted edges, determine the class of each edge.

Definition:

- class:

  which RFID tag generates this edge

- path:

  a sequence of numbers. Every number represents the class of its corresponding edge.

- branch:

  If multiple paths have the same true prefix, then they are branches of the prefix.

2. Theory - Classification

Goal: Given a sequence of extracted edges, determine the class of each edge.

Model: Max-likelihood estimation, find the path with the largest probability.

$$\max \ \mathbf{prob}(\mathrm{path}[: L])$$

- Require to check all combinations of classes in path. -> $\mathcal{NP}$
- How to avoid brute-force checking?

2. Theory - Classification

Greedy:

$$\max\ \mathbf{prob}(\text{path}[L]|\text{path}[:L-1]) \cdot \text{prob}(\text{path}[:L-1])$$

$$\max\ \mathbf{prob}(\text{previous path}) \cdot \mathbf{prob}(\text{path}[L]|\text{path}[:L-1])$$

- iterative
- Very similar to Viterbi decoding method.

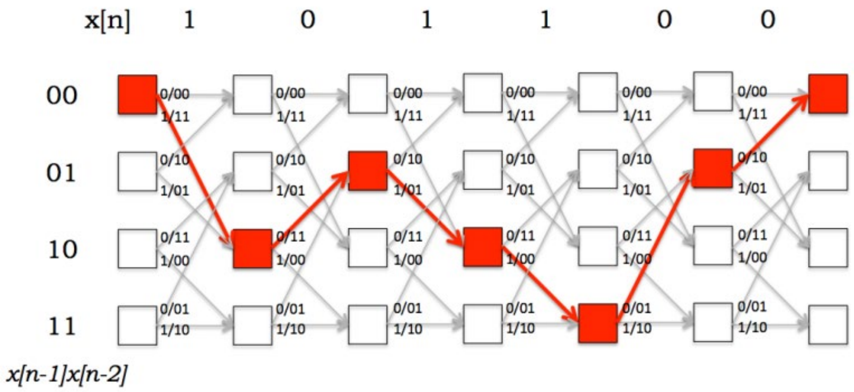
Optimality: cannot guarantee optimality

- We can add some tolerance of previous paths, i.e. retain K paths with the top K probabilities.

2. Theory - Comparison with Viterbi

Differences:

1. optimal path may not exist in the final stages.
2. elimination doesn't work.



main (mit.edu)

Analogy:

1. ML criterion: hamming distance & euclidean distance
2. not elimination, but find the branches with large probabilities.

2. Theory - Model

$$\max \ \mathbf{prob}(\text{previous path}) \cdot \mathbf{prob}(\text{path}[L] | \text{path}[: L-1])$$

Define the model for conditional probability:

- If the current point in I-Q domain is adjacent to some point in the previous path, it tends to be classified to the same tag (denoted as $class$) as its neighbor.

$$M_1[class] = \frac{\alpha}{|r - a[class]|^2} \cdot \prod_{i \neq class} |r - a[i]|^2$$

- If the current point in I-Q domain is far from any other points in the previous path, it tends to be classified to a new tag.

$$M_2 = \prod_i |a[i] - r|^2$$

2. Theory - Model

Simplify:

$$\log M_1 = -\log |r - a[class]|^2 + \sum_{i \neq class} \log |r - a[i]|^2 + \log \alpha$$

$$\log M_2 = \sum_i \log |a[i] - r|^2$$

If several branches derive from the same prefix, then the largest probability:

$$\log M_1[i] > \log M_1[j] \Leftrightarrow \log |r - a[i]|^2 < \log |r - a[j]|^2$$

$$\log M_2 > \log M_1 \Leftrightarrow \log |r - a[class]|^2 > \frac{\log \alpha}{2}$$

We tend to choose the nearest existing tag, what about new class?

2. Theory - alpha formula

Recall the wave generation: $x(t) = x_0(t) + n(t)$

- Suppose $\Re[n(t)] \sim \mathcal{N}(0, \sigma^2)$ and $\Im[n(t)] \sim \mathcal{N}(0, \sigma^2)$, mutually and time independent.
- If $r = x_0 + n_1$, $a[class] = x_0 + n_2$, generated by same RFID tag.
- then

$$|r - a[class]|^2 = (\Re[n_1] - \Re[n_2])^2 + (\Im[n_1] - \Im[n_2])^2 \sim 2\sigma^2\chi^2(2)$$

Chi-distribution credibility:

p-value	0.3	0.2	0.1	0.05
$\chi^2(2)$	2.41	3.22	4.61	5.99

2. Theory - alpha formula

Recall the wave generation: $x(t) = x_0(t) + n(t)$

- Suppose $\Re[n(t)] \sim \mathcal{N}(0, \sigma^2)$ and $\Im[n(t)] \sim \mathcal{N}(0, \sigma^2)$, mutually and time independent.
- If $r = x_0 + n_1$, $a[class] = x_0 + n_2$, generated by same RFID tag.
- then

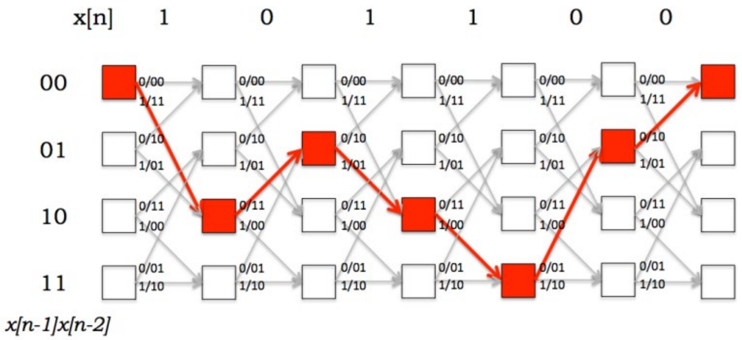
$$|r - a[class]|^2 = (\Re[n_1] - \Re[n_2])^2 + (\Im[n_1] - \Im[n_2])^2 \sim 2\sigma^2\chi^2(2)$$

Set p = 0.05, we have

$$\sqrt{\alpha} = 5.99 \cdot 2\sigma^2$$

$$\alpha = 144\sigma^4$$

(σ can be estimated by the shift-subtraction sequence)



main (mit.edu)

Project 11
Backscatter Tag Design for High Concurrency

- Background
- Theory
- **Algorithm & Optimization**
- Performance
- Summary

3.1 Extraction Algorithm

Algorithm 1: Extraction Algorithm

Input: *signal*, *thres*, *duration*
Output: *rx_signal*

```
shifted_signal := shift signal by C;
temp := shifted_signal − signal;
for edge in temp do
    if |edge| ≥ thres then
        add edge to edges;
for i in len(edges) do
    if len(rx_signal) = 0 or i − rx_signal[−1] ≥ duration then
        add i to rx_signal;
    else
        if |angle(edges[i]) − angle(edges[rx_signal[−1]])| > 0.15 then
            add i to rx_signal;
        else
            if |edges[i]| > |edges[rx_signal[−1]]| then
                rx_signal[−1] = i;
```

3.2 Classification Algorithm

Algorithm 2: Classification Algorithm

Input: rx_signal, K, α

Output: $result\_path$

$paths :=$ zeros(K,L), $path\_prob :=$ zeros(K);

$paths[0][0] := 1$;

$path\_prob[0] := 1$;

$path\_num := 1$;

for $edge$ ***in*** rx_signal **do**

    $prob\_matrix :=$ zeros$(path_num, \max\{paths\} + 1)$;

    for j ***in*** $len(path\_num)$ **do**

        $tag\_num := \max\{paths[j][:]\}$;

        for m ***in*** $tag\_num + 1$ **do**

            $prob\_matrix[j][m] = prob\_path[j] \cdot \mathbf{F}(paths[j], m, \alpha)$;

    $path\_num = \min\{\mathrm{sum}(prob\_matrix \neq 0), K\}$;

    find $path\_num$ branches with largest probabilities in $prob\_matrix$;

    update $paths$ to the $path\_num$ branches;

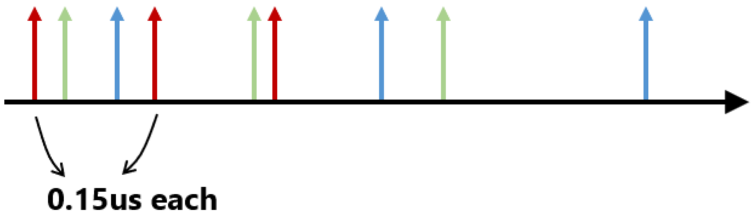
    $prob\_paths = path\_num$ largest values in $prob\_matrix$;

$res := paths[0][:]$;

$$O(LKR) \sim O(L)$$

3.3 Optimizations

- Time complexity:
  - avoid redundant computation in F: $O(R^2) \rightarrow O(R)$
  - allocate memory once + block assignment
- Period information:
  - check the index difference with the previous distance
  - if the periods are close to each other, we multiply a reward (>1) term to conditional probability; otherwise we multiply a penalty (<1) term.
- Averaging:
  - too random
  - take the ave s 3 points in the same class into fc



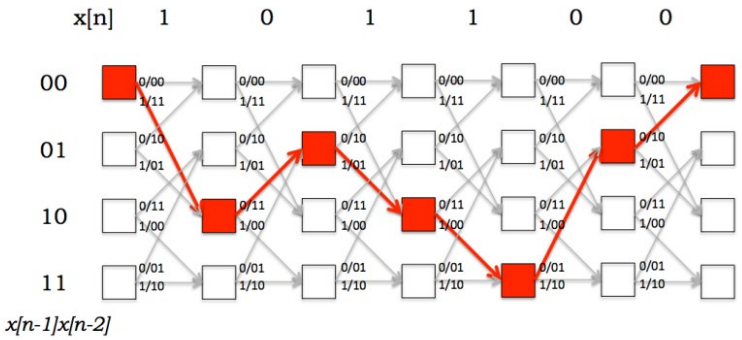
3.4 Postprocessing

- filtering invalid tags: leach out the tags in classified array that are few.

- frequency estimation: get the average Δn in the original signal for each valid tag. Therefore we can estimate the frequency of this tag:

$$2\pi f \frac{\Delta n}{f_s} = \pi \Leftrightarrow f = \frac{f_s}{2\Delta n}$$

- frequencies match test: match the estimated frequency with the actual frequency of tags.
  - How many tags are detected?
  - How many edges of each tag are classified correctly?



[main (mit.edu)](#)

Project 11
Backscatter Tag Design for High Concurrency

- Background
- Theory
- Algorithm & Optimization
- **Performance**
- Summary

4. Scalarized Criterion

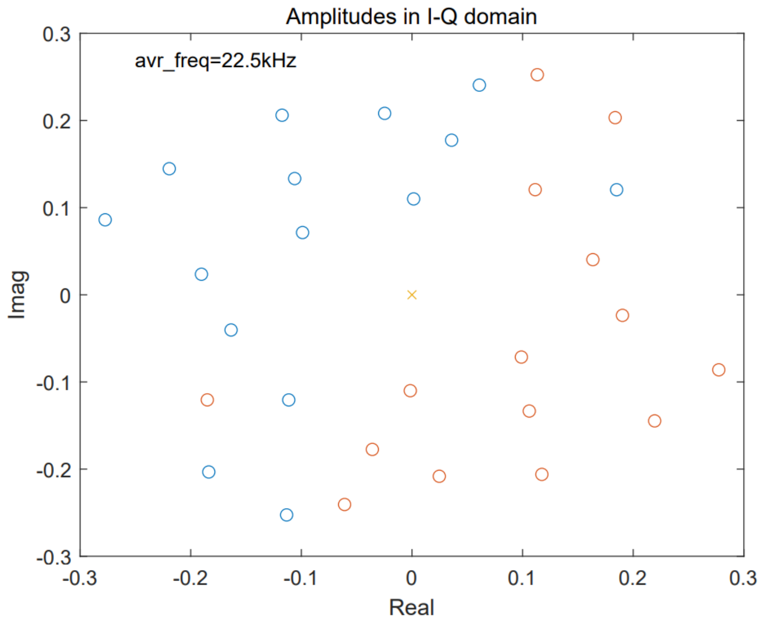
From frequencies match test, we get the edges extraction ratio for each tag.

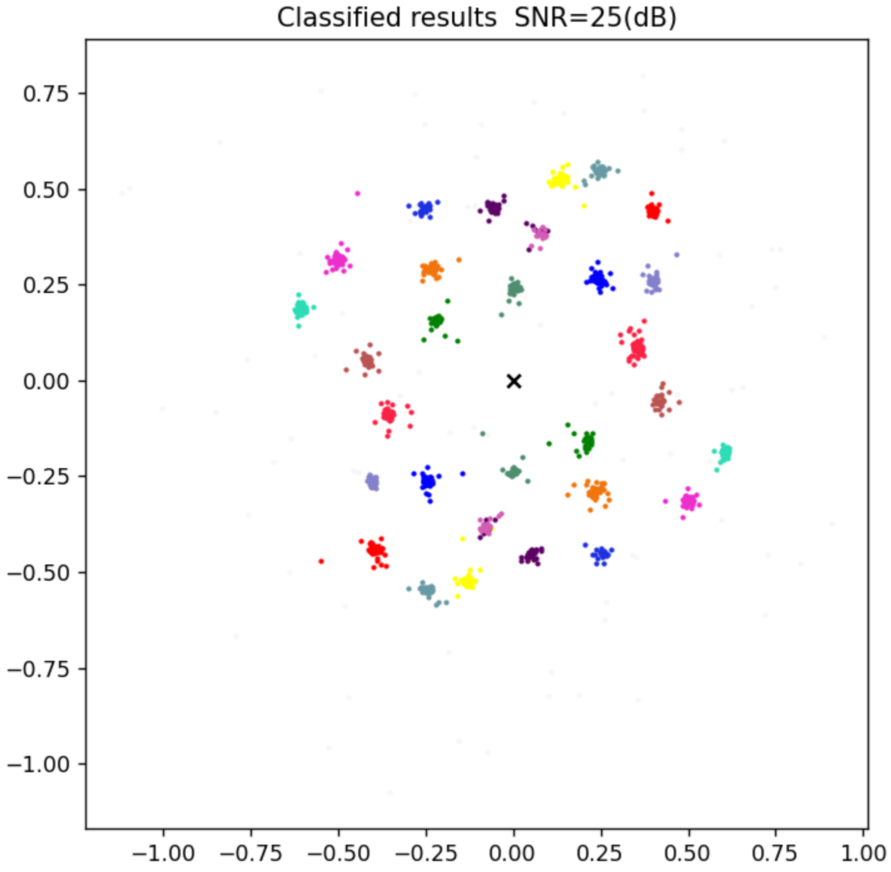
Give a vector $r \in \mathbb{R}_n^+$, we define the criterion $\varphi(r)$ to be:

$$\varphi(r) = \text{average}(\min\{r, 5 - 4r\})$$

- What's its meaning?
  - accept if we detected fewer. (misclassification, overlap, …)
  - partially accept if we detected a little more. (misclassification)
  - decline if too more.

4. Performance (simulation, tags=15)



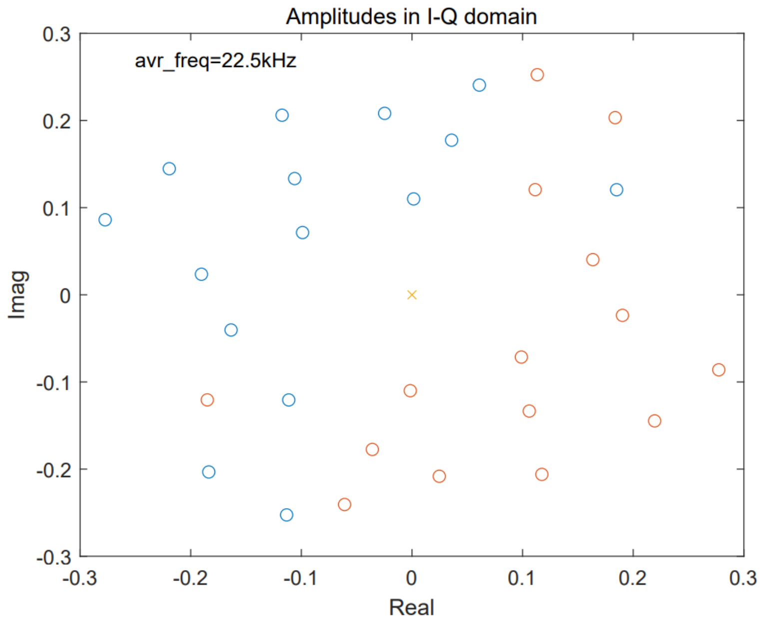


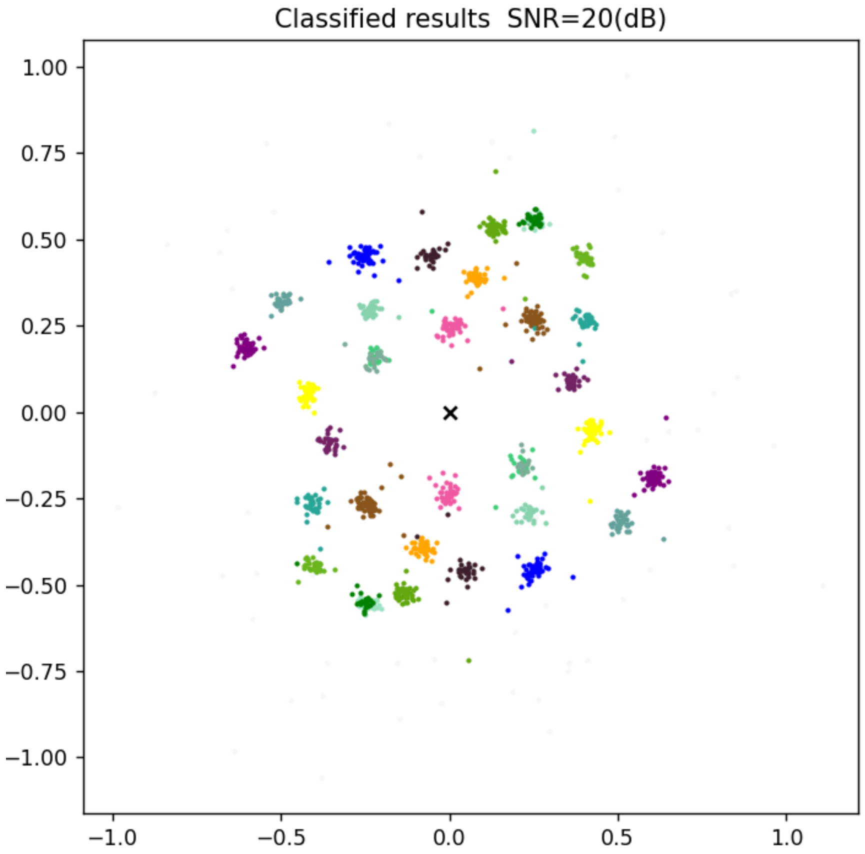
16 tags filtered out

15 tags matched (?)

$\varphi(r) = 0.913$

4. Performance (simulation, tags=15)



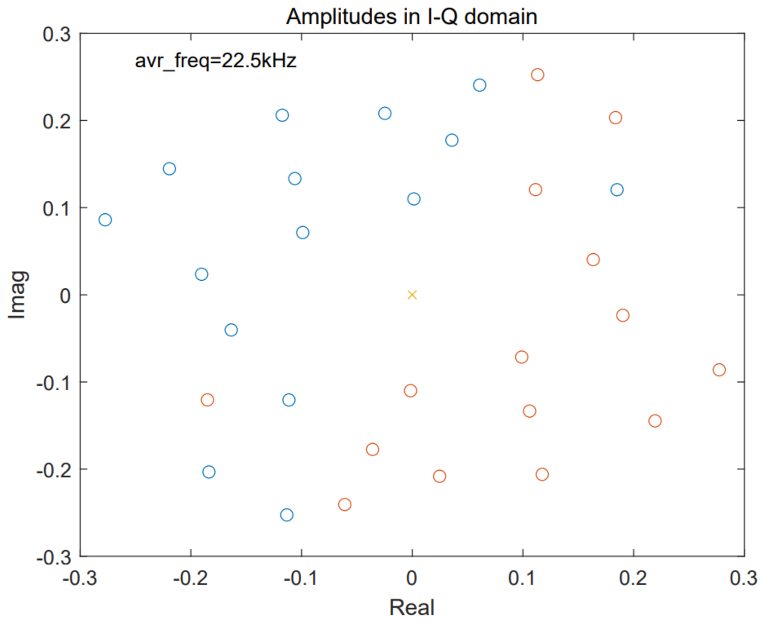


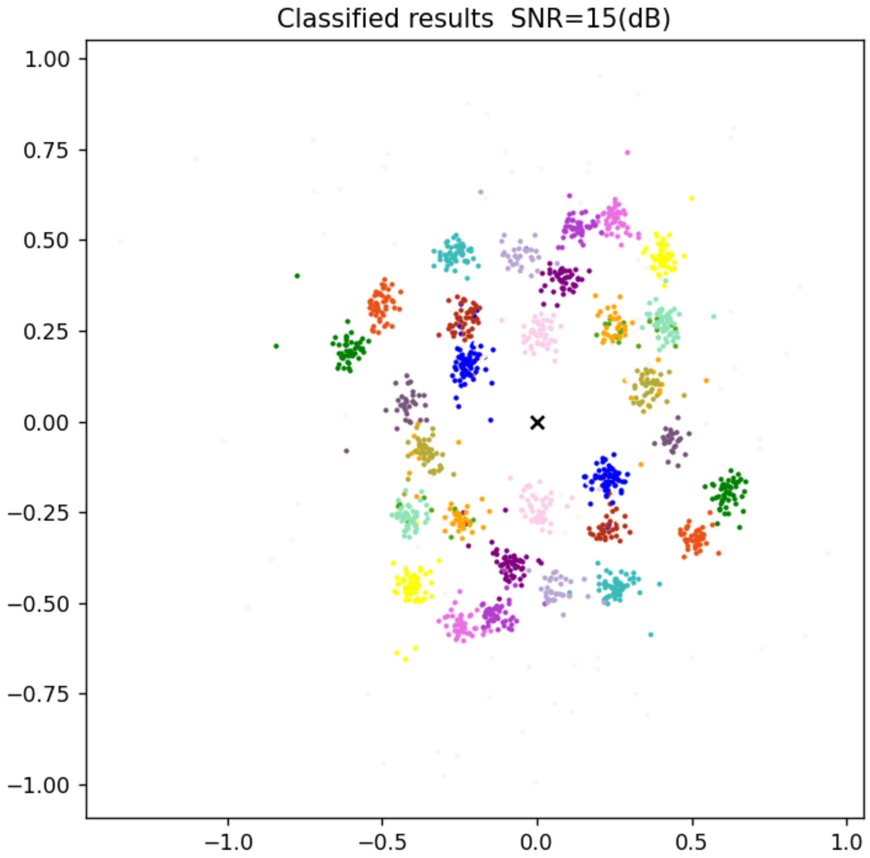
15 tags filtered out

15 tags matched

$\varphi(r) = 0.912$

4. Performance (simulation, tags=15)



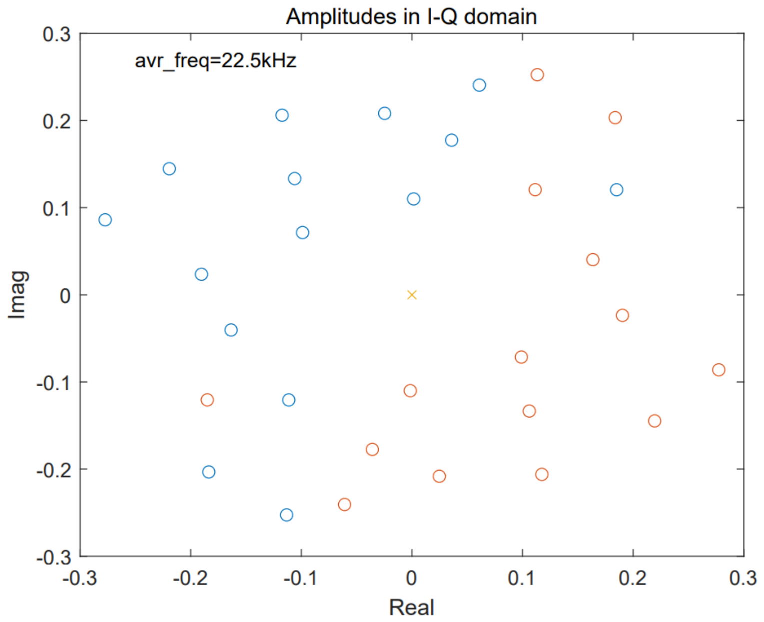


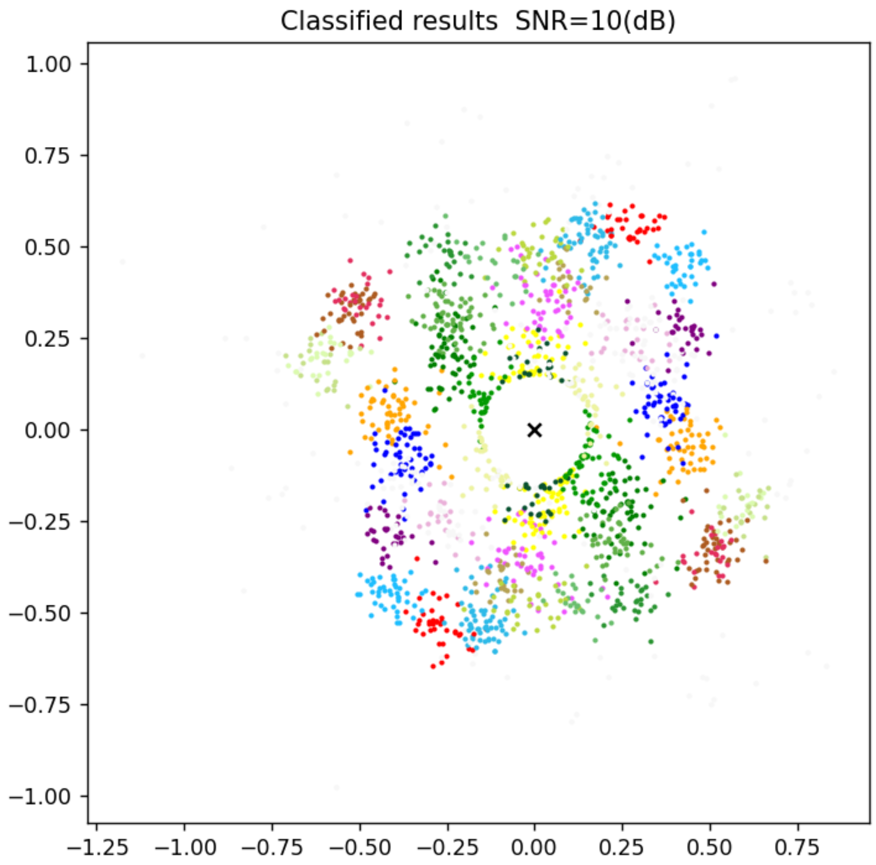
21 tags filtered out

14 tags matched

$\varphi(r) = 0.696$

4. Performance (simulation, tags=15)



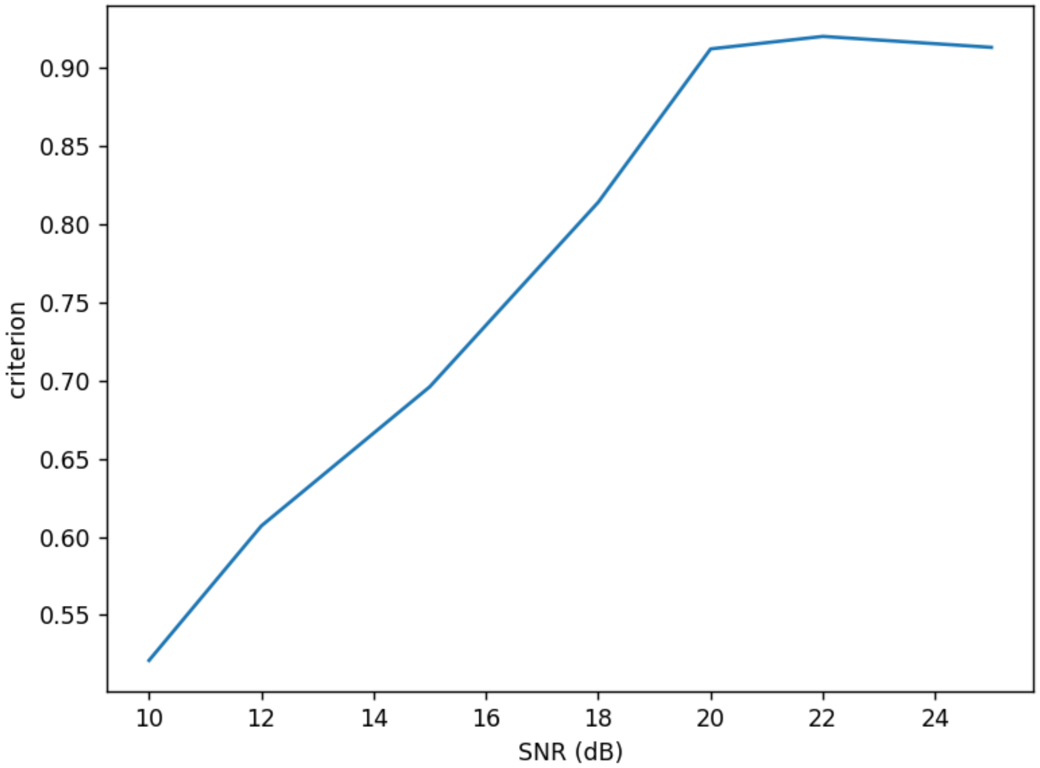


28 tags filtered out

14 tags matched

$\varphi(r) = 0.521$

4. Performance (simulation)

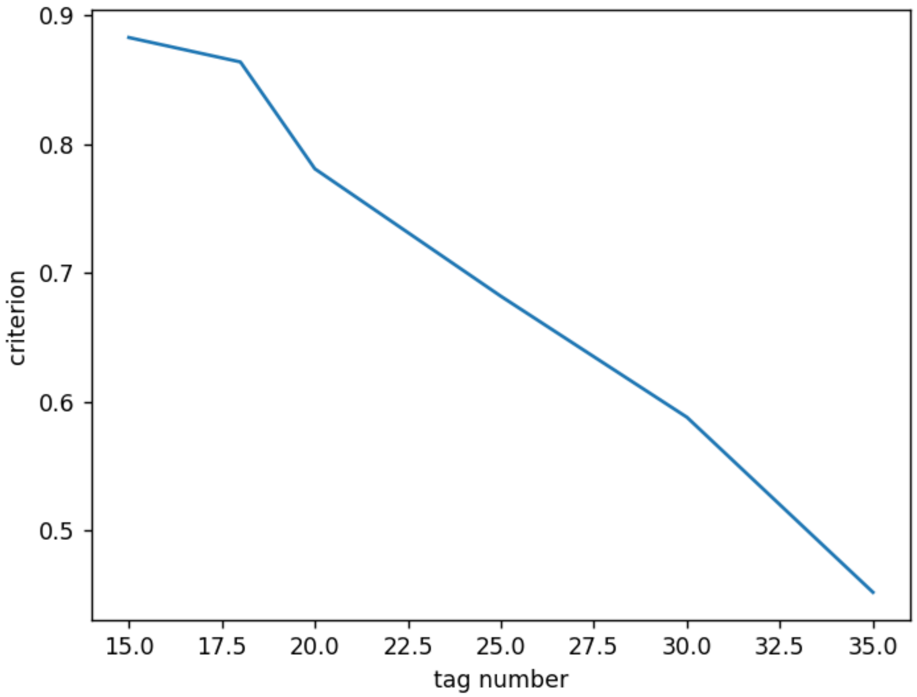


Performance - SNR curve

4. Performance (simulation)

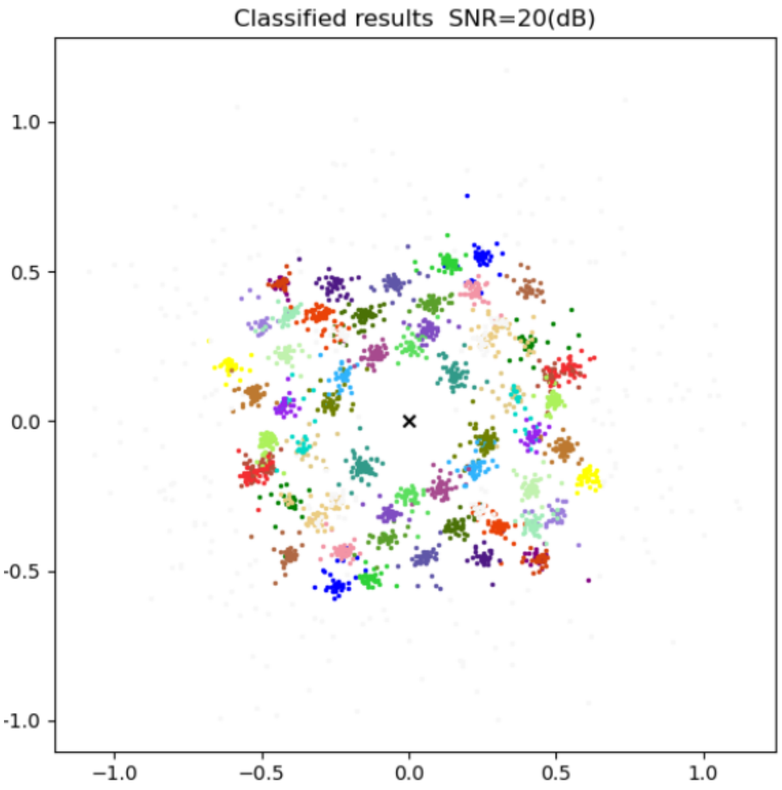
Fix SNR = 20dB

tags number	filtered number	matched number	criterion
15	16	15	0.883
18	18	18	0.864
20	20	18	0.781
25	24	21	0.682
30	30	22	0.588
35	31	20	0.452



Performance - tags number curve

4. Performance (simulation)



tags = 30

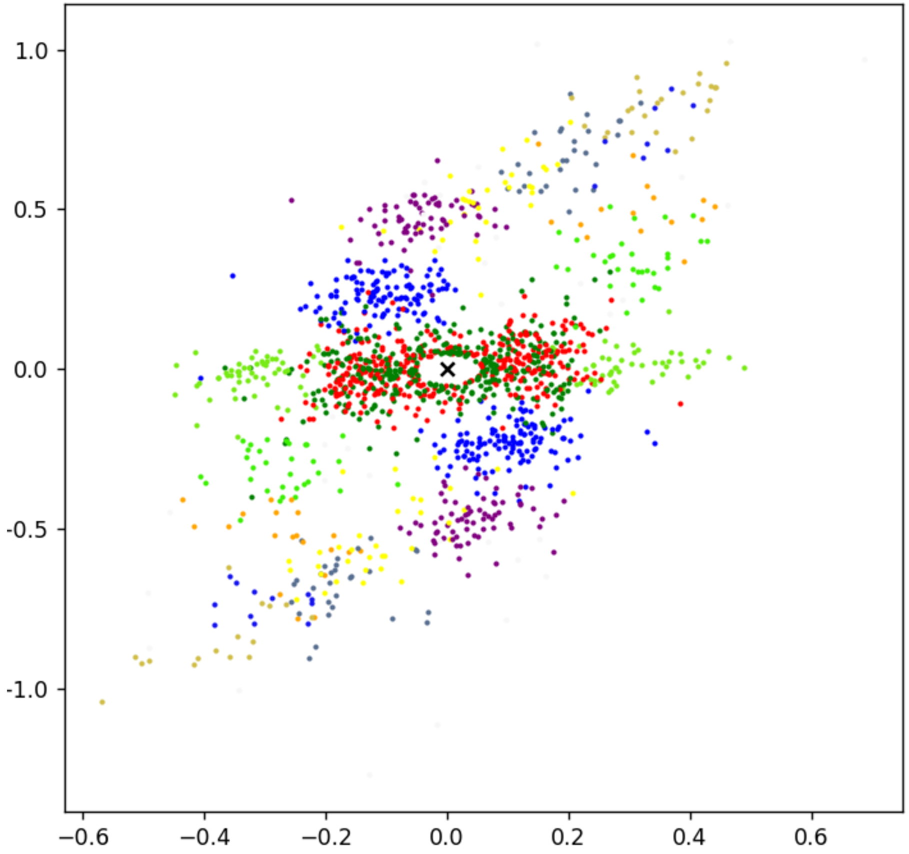
4. Performance (Real Data)

tags = 10

4 tags matched

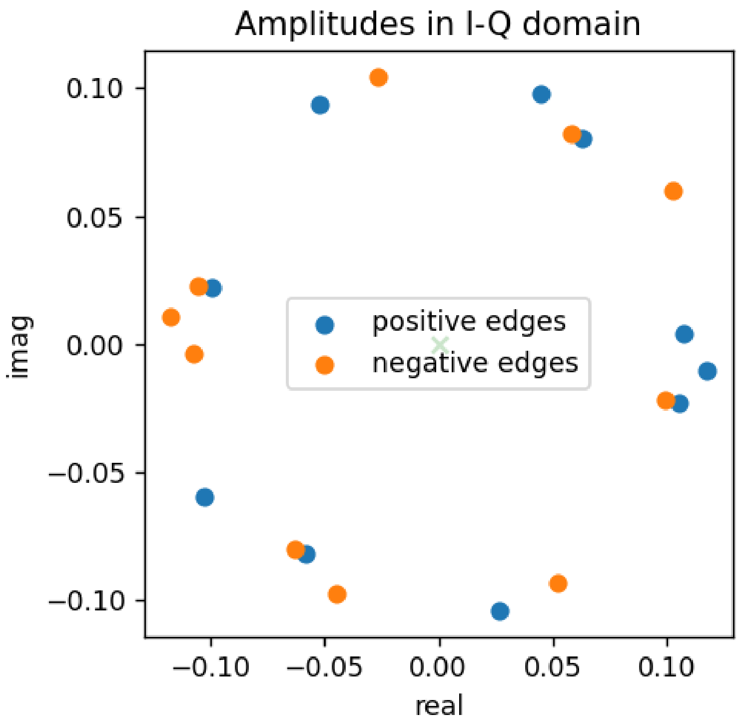
criterion < 0.3

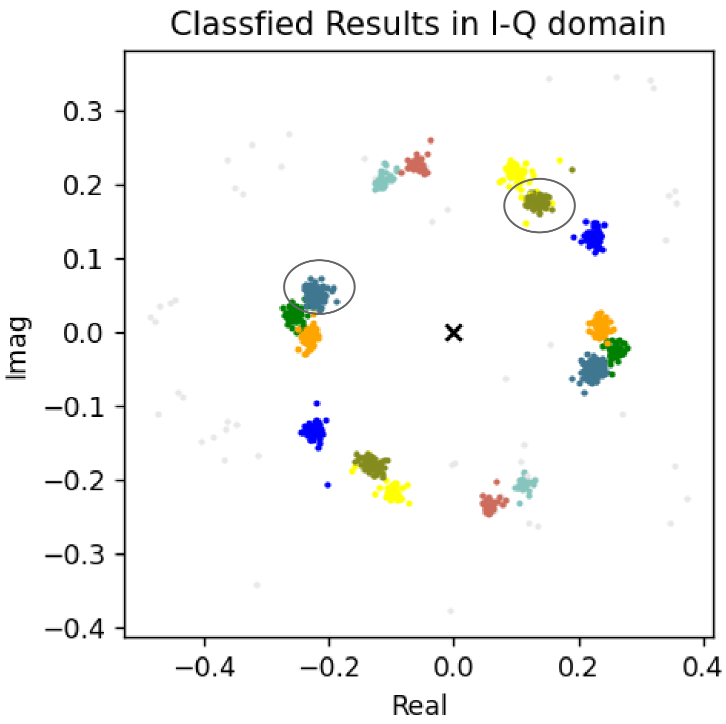
too many tags are classified to be same class -> delta n ↓, not uniform -> frequencies mismatch

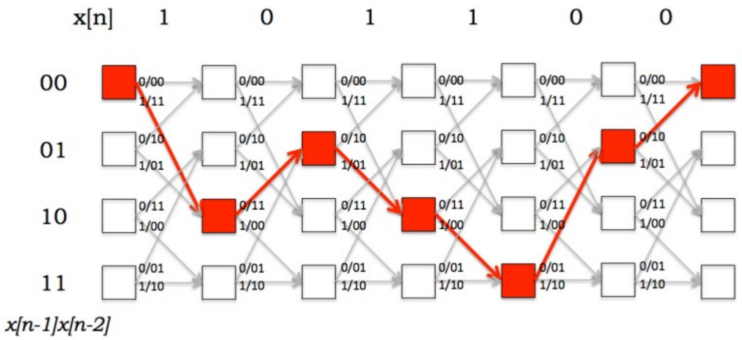


4. Performance (Real Data)

Verification of constellation problem: tags = 10, SNR = 20dB







main (mit.edu)

Project 11
Backscatter Tag Design for High Concurrency

- Background
- Theory
- Algorithm & Optimization
- Performance
- **Summary**

5. Summary

- Greedy classification algorithm:
  a. processing algorithm in the time domain $O(L)$
  b. focus on amplitude similarity (constellation requirement)
  c. not global optimal (increase K)

- Performance:
  a. concurrency $\uparrow$, performance $\downarrow$
  b. SNR $\downarrow$, performance $\downarrow$

- Paradox?

  ID is in frequency, but we care about amplitudes?