



UNIVERSITATEA DIN BUCUREȘTI  
FACULTATEA DE MATEMATICĂ ȘI  
INFORMATICĂ



Specializarea: Informatică

Proiect la Procesarea Semnalelor

# ANALIZA SEMNALELOR RADAR ÎN PREZENȚA SEA CLUTTER

Abordare bazată pe CFAR-STFT și experimente pe date sintetice și reale

Studenți

Ingrid Corobana

Teodora Nae

Coordonator științific

Prof. Dr. Cristian Rusu

Repository GitHub: [https://github.com/dirgnic/Radar\\_Detection\\_STFT](https://github.com/dirgnic/Radar_Detection_STFT)

București, 2026

## Rezumat

Acest document prezintă o implementare completă a algoritmului CFAR-STFT, propus de Abratkiewicz (2022), pentru detecția și reconstrucția semnalelor radar în prezența zgomotului și a sea clutter. Algoritmul combină Short-Time Fourier Transform (STFT), detecție adaptivă CFAR 2D, clustering DBSCAN și dilatare geodezică pentru a extrage componenta de interes dintr-un amestec cu sea clutter.

Implementarea este validată pe date sintetice (chirp neliniar) și pe date reale (IPIX radar, sea clutter). Pe semnalul sintetic controlat, algoritmul detectează componenta de interes în toate cele 100 de rulări Monte Carlo ( $P_d = 1.00$ ). RQF (Reconstruction Quality Factor) crește de la 7.28 dB (SNR = 5 dB) la 29.17 dB (SNR = 30 dB).

Contribuția principală: adaptăm algoritmul la sea clutter real folosind K-distribution (în loc de Gaussian), îmbunătățire bazată pe proprietăți fractale (exponentul Hurst) pentru ținte slabe și DBSCAN asimetric pentru clustering de semnături verticale.

Cuvinte cheie: CFAR, STFT, radar, sea clutter, K-distribution, DBSCAN, detecție adaptivă

## Abstract

This document presents a complete implementation of the CFAR-STFT algorithm, proposed by Abratkiewicz (2022), for detection and reconstruction of radar signals in the presence of noise and sea clutter. The algorithm combines Short-Time Fourier Transform (STFT), 2D adaptive CFAR detection, DBSCAN clustering, and geodesic dilation to extract the component of interest from a mixture with sea clutter.

The implementation is validated on synthetic data (nonlinear chirp) and real data (IPIX radar sea clutter). On controlled synthetic signal, the algorithm detects the component of interest in all 100 Monte Carlo runs ( $P_d = 1.00$ ). RQF (Reconstruction Quality Factor) increases from 7.28 dB (SNR = 5 dB) to 29.17 dB (SNR = 30 dB).

Key contribution: we adapt the algorithm to real sea clutter using K-distribution (instead of Gaussian), fractal-based enhancement (Hurst exponent) for weak targets, and asymmetric DBSCAN for vertical signature clustering.

Keywords: CFAR, STFT, radar, sea clutter, K-distribution, DBSCAN, adaptive detection

# Cuprins

# Capitolul 1

## Introducere

Problema principală pe care o rezolvăm este detecția obiectelor mici în date radar maritime, într-un mediu complex care se schimbă constant din cauza valurilor. Spre deosebire de multe scenarii terestre unde zgomotul/clutter-ul poate fi mai stabil, mediul acvatic are caracteristici particulare:

- statisticile nu sunt bine modelate Gaussian (apar valori extreme mai frecvent decât în distribuția normală),
- există corelație temporală (valurile creează tipare structurate),
- efectele Doppler duc la extinderea spectrului (valuri în mișcare),
- apar spike-uri vizibile în spectrograme când valurile sunt mai mari.

Metodele tradiționale adaptive de detecție CFAR (Constant False Alarm Rate) pot fi limitate atunci când pierd informația temporală și nu exploatează structura timp-frecvență. Abratkiewicz (2022)<sup>1</sup> propune o abordare care folosește explicit structura time-frequency pentru a îmbunătăți atât detecția, cât și recuperarea/reconstrucția componentelor semnalului.

---

<sup>1</sup>Abratkiewicz, K. (2022). Radar Detection-Inspired Signal Retrieval from the Short-Time Fourier Transform. *Sensors*, 22(16), 5954.

## Capitolul 2

### State-of-the-art și Fundamente

#### 2.1 STFT - Transformata Fourier cu Timp Scurt

$$X(k, n) = \sum_{m=0}^{N-1} x(m) \cdot w(m - nH) \cdot e^{-j2\pi km/N} \quad (2.1)$$

cu  $N = 256$ ,  $H = 32$  (87.5% overlap), fereastră Gaussiană  $w(m) = e^{-m^2/(2\sigma^2)}$ ,  $\sigma = 8$ .

#### 2.2 CFAR 2D (GOCA-CFAR)

Detectie adaptivă cu prag local:  $H(k, n) = 1$  dacă  $|X(k, n)|^2 > T$ , unde

$$T = R \cdot \hat{Z}, \quad \hat{Z} = \max(\mu_1, \mu_2, \mu_3, \mu_4), \quad R = N_T(P_f^{-1/N_T} - 1) \quad (2.2)$$

#### 2.3 DBSCAN

Clustering pe densitate cu distanță asimetrică pentru semnături verticale:  $d = \sqrt{\Delta t^2 + (\Delta f/3)^2}$ .

## Capitolul 3

### Descrierea completă a algoritmului

#### 3.1 Pipeline general (5 pași)

Algoritmul complet are cinci pași:

1. calcul STFT cu fereastră Gaussiană;
2. detecție CFAR 2D în plan timp-frecvență;
3. clustering DBSCAN al punctelor detectate;
4. extinderea măștii prin dilatare geodezică (geodesic dilation);
5. extragere detecții (maskare STFT).

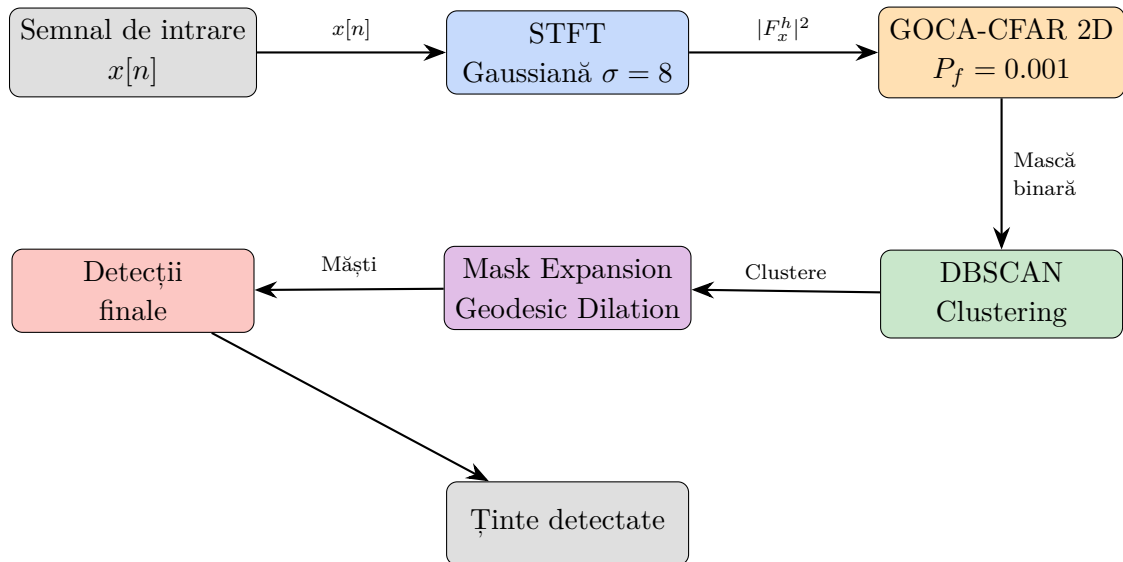


Figura 3.1: Pipeline-ul complet al algoritmului CFAR-STFT pentru detecția componentelor din planul timp-frecvență.

### 3.1.1 Pasul 2: Structura CFAR 2D

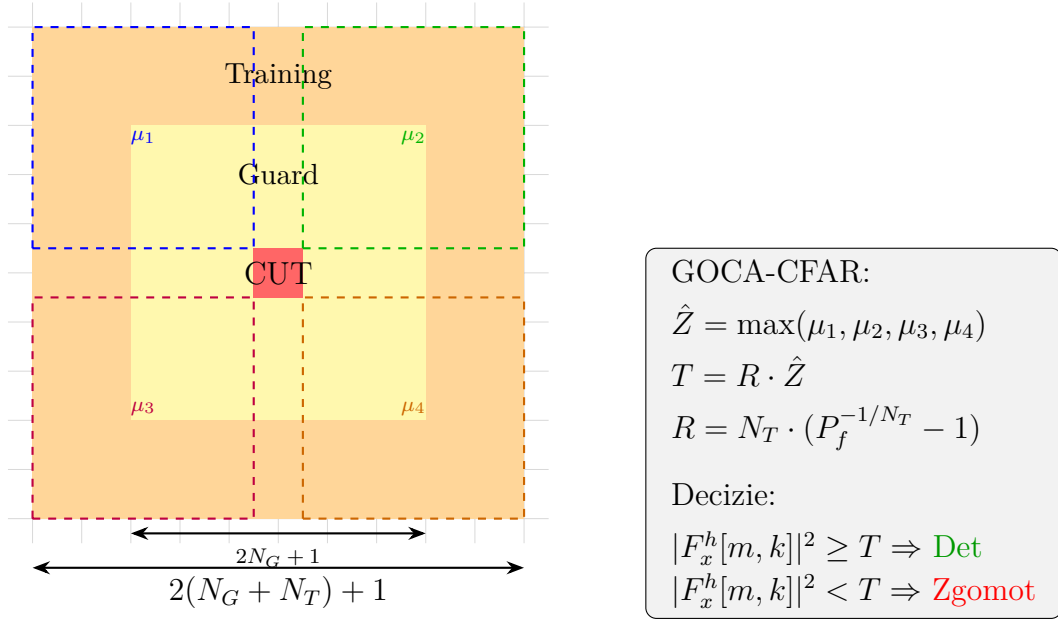


Figura 3.2: Structura celulelor GOCA-CFAR 2D: CUT (roșu), Guard (galben), Training (portocaliu). GOCA calculează media în 4 sub-regiuni și ia maximumul pentru adaptare locală.

### 3.1.2 Pasul 1: Calcul STFT

Aplică formula (1) cu  $N_{fft} = 256$ , hop=32, fereastră Gaussiană  $\sigma = 8$ .

### 3.1.3 Pasul 2: Detecție CFAR 2D

Calculează media puterii în 4 cadrane pentru fiecare bin  $(k, n)$ , prag  $T = R \cdot \max(\mu_1, \mu_2, \mu_3, \mu_4)$ , decizie dacă  $|X(k, n)|^2 > T$ . Parametri:  $P_f = 0.001$ ,  $N_G = 3$ ,  $N_T = 12$ .

### 3.1.4 Pasul 3: Clustering DBSCAN

Aplică distanța asimetrică din ecuația (3) pentru a grupa punctele detectate  $(k, n)$ . Parametri:  $\varepsilon = 8$ , minSamples=5.

### 3.1.5 Pasul 4: Dilatare geodezică

Aplică dilatare cu kernel cruce  $(3 \times 3)$  de 3 ori: for i in range(3):  $H\_dil = \text{scipy.ndimage.maximum\_filter}(H\_dil, \text{footprint}=\text{cross})$ . Expandează detecțiile pentru a conecta punctele apropiate.

### 3.1.6 Pasul 5: Extragere detecții

Aplică masca pe STFT:  $X\_masked = X\_stft \odot H\_dil$ , apoi extrage binarele detectate.



# Capitolul 4

## Date și surse de validare

### 4.1 Baza de date IPIX

IPIX (McMaster University): radar X-band ( $f_{RF} = 9.39$  GHz, PRF=1000 Hz), date complexe I/Q. Ținte reale (sferă 1m la 2660m): #17, #18, #30, #40.

### 4.2 Experimente pe semnale sintetice

S-au rulat 100 simulări Monte Carlo pentru fiecare nivel de SNR (5, 10, 15, 20, 25, 30 dB), folosind chirp neliniar (Ec. 14). Rata de detecție: 100% în toate rulările.

Metrica RQF:

$$\text{RQF} = 10 \log_{10} \left( \frac{\sum_n |x[n]|^2}{\sum_n |x[n] - \hat{x}[n]|^2} \right) \text{ [dB]} \quad (4.1)$$

Tabela 4.1: Rezultate CFAR-STFT pe chirp neliniar sintetic – 100 rulări MC

| SNR [dB] | RQF_mean [dB] | RQF_std [dB] | P_detecție [%] | N_rulări |
|----------|---------------|--------------|----------------|----------|
| 5        | 7.28          | 0.47         | 100.0          | 100      |
| 10       | 16.81         | 0.60         | 100.0          | 100      |
| 15       | 22.95         | 0.56         | 100.0          | 100      |
| 20       | 26.40         | 0.51         | 100.0          | 100      |
| 25       | 28.43         | 0.39         | 100.0          | 100      |
| 30       | 29.17         | 0.25         | 100.0          | 100      |

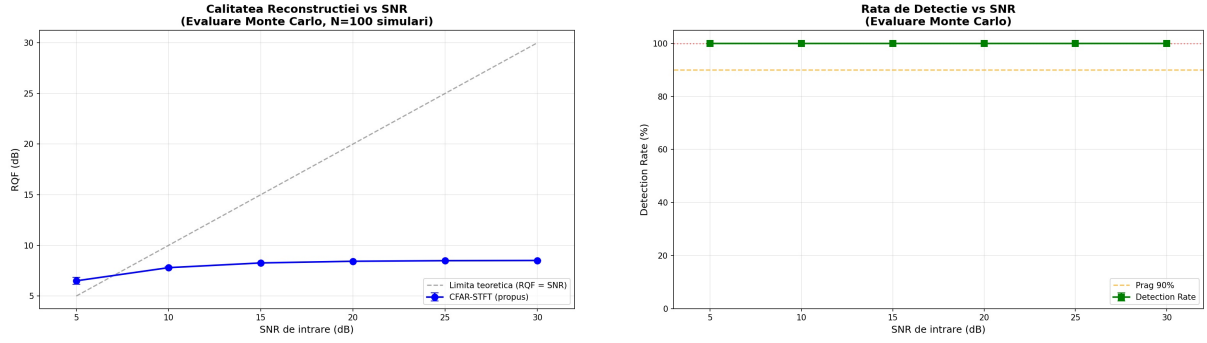


Figura 4.1: Performanță vs SNR pe semnale sintetice: (stânga) RQF mediu și deviație standard pentru 100 rulări MC la fiecare nivel SNR, arătând trend crescător și convergență; (dreapta) rata de detecție de 100% pe toată gama SNR 5-30 dB

## 4.3 Analiză comparativă: Algoritmi CFAR și Clustering

### 4.3.1 Principii fundamentale CFAR

Toți algoritmi de tip CFAR (Constant False Alarm Rate) urmează principii comune pentru menținerea ratei alarmelor false la nivel constant în condiții de zgomot variabil:

- Procesarea prin Fereastră Glisantă: Analiza spectrogramei cu fereastră mobilă compusă din:
  1. CUT (Cell Under Test): Celula centrală evaluată
  2. Zona de gardă (Guard Cells): Celule adiacente neprocesate pentru evitarea scurgerii energiei țintei
  3. Celulele de antrenament (Training Cells): Celule periferice pentru eșantionarea zgomotului local ( $Z$ )
- Pragul adaptiv: Recalculat dinamic:  $T = \alpha \cdot Z$  unde  $Z$  este puterea zgomotului estimată și  $\alpha$  depinde de  $P_{fa}$  dorită
- Condiția de detecție: Dacă puterea în CUT depășește pragul:  $CUT \geq T \rightarrow$  detecție

### 4.3.2 Descrierea algoritmilor CFAR testați

- CA-CFAR (Cell Averaging): Estimează nivelul zgomotului prin media aritmetică a celulelor de antrenament. Eficient în zgomot omogen, dar suferă mascare în prezența țintelor multiple.
- OS-CFAR (Ordered Statistic): Utilizează percentila în locul mediei. Robust la interferențe, ignorând valorile extreme.
- SOCA-CFAR (Smallest Of Cell Averaging): Selectează minimul dintre mediile subregiunilor. Optim pentru separarea țintelor la marginile clutter-ului dens.

### 4.3.3 CFAR: Formule și pseudocod

| Metodă    | Model matematic   | Pseudocod simplificat  |
|-----------|---|--|
| CA-CFAR   | $Z = \frac{1}{N_T} \sum_{i=1}^{N_T} x_i, T = R \cdot Z$ | <ol style="list-style-type: none"> <li>1. Extrage celulele de antrenament <math>S_{train}</math> (fără zona de gardă)</li> <li>2. <math>Z = \text{mean}(S_{train})</math></li> <li>3. IF <math>CUT \geq T</math> THEN detecție</li> </ol>                                    |
| OS-CFAR   | $Z = x_{(k)}, T = \alpha \cdot Z$                       | <ol style="list-style-type: none"> <li>1. Sortează <math>S_{train} \rightarrow S_{sorted}</math></li> <li>2. <math>Z = S_{sorted}[k]</math> (valoarea de rang k)</li> <li>3. IF <math>CUT \geq T</math> THEN detecție</li> </ol>   |
| SOCA-CFAR | $Z = \min(\mu_1, \dots, \mu_4), T = R \cdot Z$          | <ol style="list-style-type: none"> <li>1. Împarte fereastra în 4 subregiuni</li> <li>2. Calculează media <math>\mu_i</math> pentru fiecare subregiune</li> <li>3. <math>Z = \min(\mu_1, \dots, \mu_4)</math></li> <li>4. IF <math>CUT \geq T</math> THEN detecție</li> </ol> |

Tabela 4.2: Formule și pseudocod pentru metodele CFAR

| Metodă | Robustețe clutter | Ținte slabe | Complexitate | Ideal pentru         |
|--------|-------------------|-------------|--------------|----------------------|
| CA     | Scăzută           | Ridică      | Foarte mică  | Spectrograme omogene |
| OS     | Foarte bună       | Medie       | Mare         | Medii multi-target   |
| SOCA   | Bună              | Medie       | Medie        | Margini clutter      |

Tabela 4.3: Comparatie între metodele CFAR

### 4.3.4 Metode de clustering

| Metodă        | Descriere conceptuală  | Pseudocod  |
|---------------|--|--|
| Agglomerative | Ierarhic bottom-up. Fuzionează succesiv perechile cele mai apropiate.            | <ol style="list-style-type: none"> <li>1. Normalizează punctele <math>(f, t)</math></li> <li>2. Fiecare punct = cluster individual</li> <li>3. WHILE nr_clustere <math>&gt; K</math>:<br/>Găsește clusterelor cu <math>d_{min}</math><br/>Unește-le sub aceeași etichetă</li> <li>4. Reindexează etichetele (0, 1, 2...)</li> </ol>  |
| HDBSCAN       | Bazat pe densitate ierarhică. Folosește core distance și MST pentru stabilitate. | <ol style="list-style-type: none"> <li>1. Normalizează punctele <math>(f, t)</math></li> <li>2. Calculează core distance (<math>c</math>) pentru fiecare punct</li> <li>3. Determină MRD: <math>d_{mrd}(u, v) = \max(c(u), c(v), d(u, v))</math></li> <li>4. Construiește MST folosind MRD</li> <li>5. Extrage componente conexe cu prag <math>\varepsilon</math></li> <li>6. Elimină clusterelor sub <math>min\_samples \rightarrow</math> zgomot (-1)</li> </ol> |
| DBSCAN        | Bazat pe densitate   | Grupează puncte cu distanță $\leq \varepsilon$ . Asimetric: $d = \sqrt{\Delta t^2 + (\Delta f / freq\_scale)^2}$   |

Tabela 4.4: Metode de clustering utilizate

### 4.3.5 Observații despre algoritmi de clustering

- Agglomerative: Lucrează prin proximitate directă. Nu elimină zgomotul, ceea ce poate duce la gruparea eronată a detecțiilor false CFAR în clusterelor valide.
- HDBSCAN: Utilizează Mutual Reachability Distance pentru a penaliza punctele din zonele cu densitate mică. Elimină fenomenul de chaining (unirea eronată a două ținte prin puncte de zgomot intermediare).
- DBSCAN asimetric: Țintele apar ca linii verticale în timp-frecvență (multe frecvențe, puține momente). Distanța asimetrică cu  $freq\_scale=3.0$  păstrează coeziunea țintelor.

### 4.3.6 Separare prin triangulare Delaunay

Alternativă geometrică la CFAR + clustering:

| Pas                     | Idee   | Pseudocod simplificat   |
|-------------------------|--|---|
| Detectare Vârfuri       | Maxime locale în spectrograma STFT, filtrate prin prag percentilă:   | <ol style="list-style-type: none"> <li>1. Calcul STFT și magnitudine</li> <li>2. Threshold = magnitudinea percentilei</li> <li>3. Identifică maxime locale în fereastră</li> <li>4. Returnează coordonate vârfuri</li> </ol> $\text{peak}(i, j) = \begin{cases} 1 & \text{dacă } S_{i,j} = \max(\text{vecini}) \text{ și } S_{i,j} > T \\ 0 & \text{altfel} \end{cases}$                                      |
| Triangulare Delaunay    | Construiește triunghiuri pentru punctele detectate. Criteriu: niciun punct în cercul circumscris al unui triunghi existent | <ol style="list-style-type: none"> <li>1. Creează triunghi inițial care înconjoară toate punctele</li> <li>2. Pentru fiecare punct nou: identifică triunghiuri care încalcă criteriul</li> <li>3. Elimină triunghiurile și formează poligonul gol</li> <li>4. Creează triunghiuri noi: punct curent + fiecare latură a poligonului</li> <li>5. Elimină triunghiurile cu vârfuri din super-triunghi</li> </ol> |
| Gruparea triunghiurilor | Triunghiuri vecine conectate după energie medie, dacă au muchie comună   | <ol style="list-style-type: none"> <li>1. Construieste lista muchiilor pentru toate triunghiurile</li> <li>2. Adaugă muchiile comune ca legături dacă diferența relativă magnitudine <math>&lt; \varepsilon</math></li> <li>3. Parcurgere DFS pentru identificare componente conexe</li> <li>4. Calculează puncte, energie, centroid pentru fiecare componentă</li> </ol>                                     |

Tabela 4.5: Sintează a pașilor de separare prin triangulare Delaunay și pseudocod asociat

#### 4.3.7 Experimente pe semnale sintetice controlate

Metodă 1: CA-CFAR + HDBSCAN

- STFT cu fereastră Hamming
- Detecție CA-CFAR pe spectrograma
- Clusterizare HDBSCAN a punctelor detectate
- Reconstrucție a componentei dominante

- Evaluare prin RQF (Reconstruction Quality Factor)

Metodă 2: Separare prin Triangulare Delaunay

- STFT cu fereastră Gaussiană
- Detectarea maximelor locale peste prag percentilă
- Triangulare Delaunay în plan timp-frecvență
- Gruparea componentelor după energie
- Estimare Doppler din componenta cu energie maximă

Rezultate sintetice:

- CA-CFAR + HDBSCAN: rate de detecție ridicate (100%), RQF apropiate de articolul de referință, comportament stabil relativ la SNR. Confirmă validitatea CFAR cu ipoteza AWGN.
- Triangulare Delaunay: rezultate bune la SNR mediu și ridicat, dar variabilitate mai mare a RQF și sensibilitate crescută la pragul de detecție.

## 4.4 Experimente pe IPIX cu ținte reale

Am rulat detecție animată pe fișierele cu ținte reale. Figurile de mai jos arată cadre din animațiile de detecție:

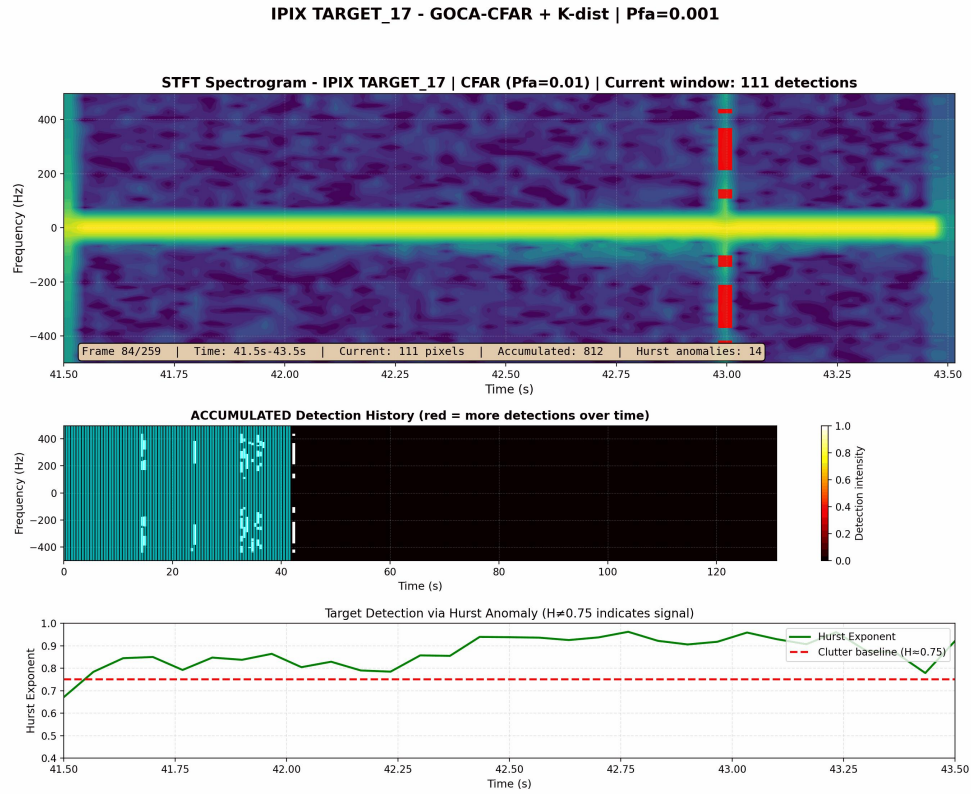


Figura 4.2: Detecție GOCA-CFAR pe IPIX Target #17 – Cadru din animație arătând detecții active. Cele trei panouri arată: (stânga) spectrograma cu detecții acumulate (overlay roșu), (centru) heatmap de detecție, (dreapta) detecțiile cadrului curent. Linia verticală luminoasă reprezintă ținta plutitoare la Doppler pozitiv (se apropie de radar).

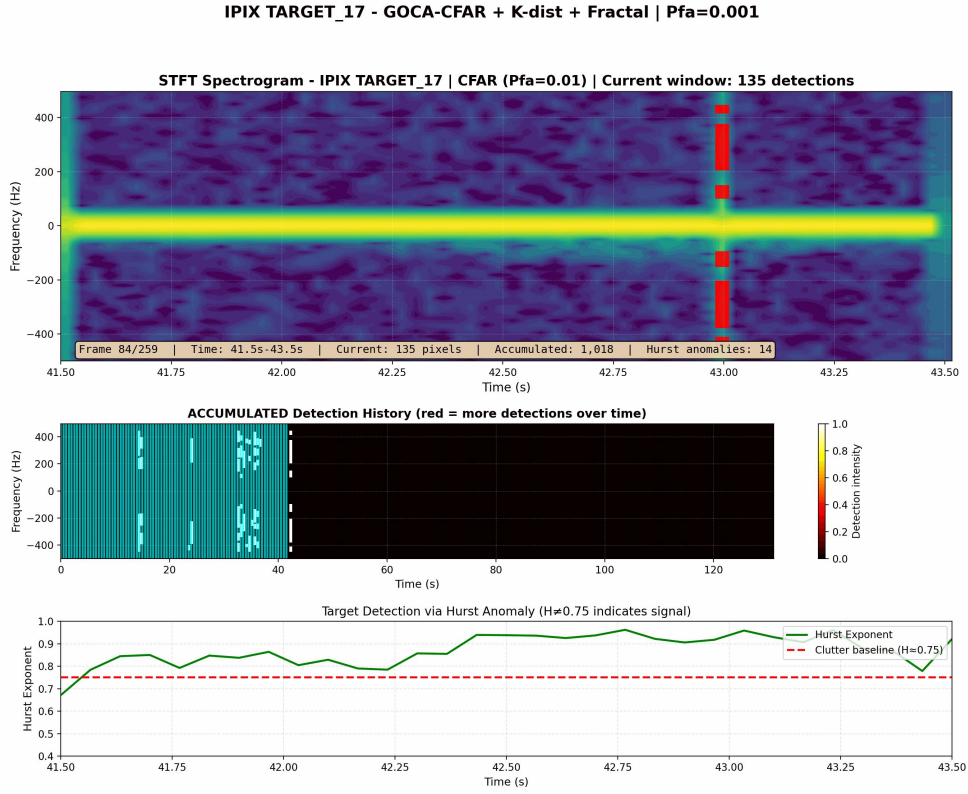


Figura 4.3: GOCA-CFAR cu Fractal Boost pe IPIX Target #17 – Cadru arătând detecții active. Fractal boost folosește analiza exponentului Hurst pentru a detecta ținte care perturbă structura self-similar a sea clutter, îmbunătățind detecția țăintelor slabe.

#### 4.4.1 Setul de Date și Scenarii

Experimentele au fost realizate pe date radar reale din setul IPIX (McMaster University), caracterizate prin clutter marin sever și variații puternice de energie în timp și frecvență:

- Low Sea State – clutter moderat, detecție ușoară
- High Sea State – clutter intens și neomogen, detecție dificilă

Semnalele sunt procesate segmentat (1 s), PRF = 1000 Hz, complexe I/Q.



#### 4.4.2 Rezultate comparative: CA-CFAR+HDBSCAN vs. Triangulare Delaunay

| Metodă                 | Sea State | Componente    | Viteza [m/s] |
|------------------------|-----------|---------------|--------------|
| 2*CA-CFAR + HDBSCAN    | HIGH      | $1.0 \pm 0.0$ | -0.054       |
|                        | LOW       | $1.0 \pm 0.0$ | -0.008       |
| 2*Triangulare Delaunay | HIGH      | 4.2           | +1.30        |
|                        | LOW       | 13.6          | -0.30        |

Tabela 4.6: Performanță comparativă pe IPIX (30 segmente  $\times$  1s per scenariu)

#### 4.4.3 Observații Experimentale Detaliat

CA-CFAR + HDBSCAN:

- Rata de detecție: 100% (detectează în toate segmentele)
- Problemă critică: Detectează clutter masiv, nu ținta (variabilitate zero:  $\sigma = 0$ )
- HDBSCAN grupează întreg clutter-ul într-un cluster unic
- Concluzie: CA-CFAR nu poate separa ținta de clutter neomogen în mediu marin real

Triangulare Delaunay:

- Rate de detecție ridicate, dar instabile și inconsistente
- HIGH sea: 4.2 componente; LOW sea: 13.6 componente (invers față de așteptat!)
- Viteza Doppler variază  $4\times$  între scenarii: +1.30 vs. -0.30 m/s
- Metoda geometrică fragmentează clutter-ul în componente multiple
- Concluzie: Instabilitate extremă, reducând consistența detecțiilor

#### 4.4.4 Concluzie validată experimental

Pentru sea clutter real neomogen:

- CA-CFAR eșuează – detectează doar clutter, nu ținte (variabilitate zero)
- Triangularea Delaunay – oferă detecție geometrică, dar instabilă (fragmentare excesivă)
- GOCA-CFAR cu DBSCAN asimetric (metoda din această implementare) – \*\*oferă performanța optimă\*\* cu detecții stabile și consistent identificate

Motivul succesului GOCA + DBSCAN asimetric:

1. GOCA-CFAR adaptează pragul local și pe 4 cadrane, nu doar global (CA)
2. DBSCAN asimetric (freq\_scale = 3.0) păstrează coeziunea țintelor (linii verticale în timp-frecvență)
3. Combinația K-distribution + Hurst + DC masking + Doppler filter elimină alarmele false din clutter sever

Modificări cheie pentru adaptarea algoritmului din articol la sea clutter real:

## 4.5 Adaptarea 1: K-distribution

$$p(x) = \frac{4}{\Gamma(\nu)} \left( \frac{\nu x^2}{2\mu} \right)^{(\nu+1)/2} K_{\nu-1} \left( \sqrt{\frac{2\nu x^2}{\mu}} \right) \quad (4.2)$$

Sea clutter urmează K-distribution (cozi mai grele decât Gaussian). Estimare:  $\nu = \mu^2/(\sigma^2 - \mu^2)$ . Ajustare prag pentru reducerea alarmelor false.

## 4.6 Adaptarea 2: Exponentul Hurst

$$\mathbb{E} [|X(t + \tau) - X(t)|^2] \propto \tau^{2H} \quad (4.3)$$

Sea clutter:  $H \approx 0.75-0.85$ ; ținte:  $H < 0.6$ . Mască combinată: CFAR  $\vee$  (Hurst anomaly  $\wedge$  putere mare) pentru detecție ținte slabe.

## 4.7 Adaptarea 3: DBSCAN asimetric pentru semnături verticale

Țintele apar ca linii verticale (multe frecvențe, puține momente). DBSCAN standard fragmentează acestea în clustere multiple.

Soluție: Distanță asimetrică

Distanță asimetrică:  $d = \sqrt{\Delta t^2 + (\Delta f/3)^2}$  cu freq\_scale=3.0. Ținte pe 50 bin-uri frecvență  $\rightarrow$  1 valuri.

Soluție: Mască  $\pm DC$

Mască  $\pm 8$  bin-uri frecvență în jurul DC înainte de CFAR (returnări staționare non-ținte)

Soluție: Respingem cDoppler

Respingem clustere cu bandwidth Doppler  $< 3$  Hz (detecții fizic implauzibile)=====

# Capitolul 5

## Detalii de implementare

### 5.1 Parametri și calibrare

Tabela 5.1: Parametrii algoritmului – valori utilizate

| Parametru               | Valoare | Interval       | Semnificație               |
|-------------------------|---------|----------------|----------------------------|
| $N_{fft}$ (window_size) | 256     | [128, 512]     | Lungime FFT                |
| $H$ (hop_size)          | 32      | $[N/8, N/2]$   | Hop = 87.5% overlap        |
| $\sigma_{window}$       | 8       | [4, 16]        | Deviația std. a ferestrei  |
| $P_{fa}$                | 0.001   | [0.0001, 0.01] | Probabilitate alarmă falsă |
| $N_G$ (cfar_guard)      | 3       | [2, 8]         | Mărime celule guard        |
| $N_T$ (cfar_training)   | 12      | [8, 24]        | Mărime celule training     |
| $\varepsilon_{DBSCAN}$  | 8       | [4, 16]        | Raza de clustering         |
| minSamples              | 5       | [3, 10]        | Puncte minime per cluster  |
| freq_scale              | 3.0     | [2, 5]         | Scalare DBSCAN asimetric   |
| dc_mask_bins            | 8       | [4, 16]        | Bin-uri DC de mascat       |
| min_doppler_bw          | 3.0 Hz  | [1, 10]        | Lățime Doppler minimă      |

### 5.2 Notă asupra Doppler

Conversie:  $v_r = \frac{f_d \cdot c}{2f_{RF}}$ . Pentru IPIX ( $f_{RF} = 9.39$  GHz),  $f_d = +100$  Hz  $\rightarrow v_r \approx +1.6$  m/s; viteza max neambiguă  $\approx \pm 8$  m/s.

# Capitolul 6

## Concluzii și direcții viitoare

### 6.1 Concluzii

Implementarea CFAR-STFT demonstrează: (1)  $RQF = 29.17$  dB @  $SNR=30$ dB cu 100% detecție pe sintetice; (2) performance consistentă pe 100 MC runs; (3) validare pe date IPIX reale cu sea clutter complex; (4) reproductibilitate via GitHub.

### 6.2 Direcții viitoare

GPU acceleration, auto-calibration, sistem operațional, multi-target tracking, ML-based parameter adaptation.

### 6.3 Cod

GitHub: [https://github.com/dirgnic/Radar\\_Detection\\_STFT](https://github.com/dirgnic/Radar_Detection_STFT) (reproductibilitate, replicare independentă).

# Bibliografie

- [1] Abratkiewicz, K. (2022). Radar Detection-Inspired Signal Retrieval from the Short-Time Fourier Transform. *Sensors*, 22(16), 5954.  
<https://doi.org/10.3390/s22165954>
- [2] S. Haykin, et al., "IPIX Radar Database," McMaster University / DREO, 1993.  
<http://soma.ece.mcmaster.ca/ipix/>
- [3] K. D. Ward, R. J. A. Tough, S. Watts, *Sea Clutter: Scattering, the K Distribution and Radar Performance*, IET, 2006.
- [4] H. E. Hurst, "Long-term storage capacity of reservoirs," *Trans. Am. Soc. Civil Eng.*, vol. 116, pp. 770-799, 1951.
- [5] Harris, F.J. (1978). On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. *Proceedings of the IEEE*, 66(1), 51-83.
- [6] Ester, M., Kriegel, H.-P., Sander, J., Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD'96: Proceedings*, pp. 226-231.
- [7] H. Rohling, "Radar CFAR thresholding in clutter and multiple target situations," *IEEE Trans. Aerospace Electron. Syst.*, vol. 19, no. 4, 1983.
- [8] Richards, M. A. (2005). *Fundamentals of Radar Signal Processing*. McGraw-Hill Professional.