

# Analiza semnalelor radar în prezența ecourilor marine

Abordare CFAR-STFT cu adaptări pentru sea clutter real

Ingrid Corobana    Teodora Nae

Universitatea din București  
Facultatea de Matematică și Informatică  
Coordonator: Conf. Dr. Cristian Rusu

Procesarea Semnalelor – 2026

# Cuprins

- 1 Introducere și Motivație
- 2 Fundamente Teoretice
- 3 Adaptări pentru Sea Clutter
- 4 Rezultate Experimentale
- 5 Comparatie Parametri
- 6 Concluzii

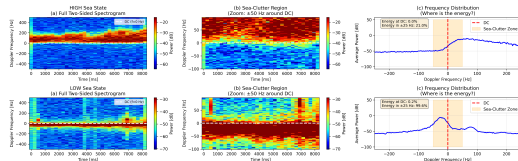
# Problema: Detecția în Sea Clutter

## Sea Clutter:

- Ecouri de la suprafața mării
- Non-Gaussian cu spike-uri frecvente
- Ascunde ținte mici

## Provocări:

- K-distribution statistics
- Corelație temporală
- Ținte slabe sub clutter



SEA CLUTTER EXPLANATION:

- Peșe pe rețea DC (0 Hz) = ecosisteme statice de la suprafața mării
- Gelber/Wende la jurul DC = clutter-ul marin în mișcare (valuri, spume)
- Albăstru/Lila = zgomot termic, sunet, radio
- "Clutter" = energie slabă concentrată în anumite zone temporale/frecvențiale
- Tintele reale se află cu bariere cauze de DC la 100-400 Hz

Figura: Spectrogramă IPIX

# Soluția: CFAR-STFT (Abratkiewicz 2022)

**Ideea cheie:** Exploatăm structura **timp-frecvență**!

- ① **STFT** → Reprezentare 2D (spectrogramă)
- ② **CFAR 2D** → Detecție adaptivă local
- ③ **DBSCAN** → Clustering puncte detectate
- ④ **Dilatare geodezică** → Extindere mască
- ⑤ **iSTFT** → Reconstrucție semnal

**Contribuțiile noastre:**

- Implementare completă în Python
- **Adaptări pentru sea clutter real:** K-distribution, Hurst boost, DBSCAN asimetric
- Validare pe date IPIX reale

# STFT – Transformata Fourier cu Timp Scurt

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

## Parametri:

- $N_{fft} = 256$  (pentru IPIX)
- Hop  $H = 32$  (87.5% overlap)
- Fereastră Gaussiană  $\sigma = 8$

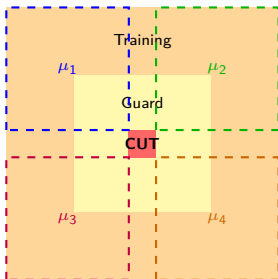
## De ce STFT?

- Localizare timp-frecvență
- Semnalele radar = ridges în TF
- Clutter-ul = energie difuză

Two-sided pentru date complexe I/Q

Date IPIX = I + jQ  $\Rightarrow$  Frecvențe Doppler pozitive (apropiere) și negative (depărtare)

# GOCA-CFAR 2D – Detecție Adaptivă



**GOCA = Greatest-Of Cell Averaging**

$$\hat{Z} = \max(\mu_1, \mu_2, \mu_3, \mu_4)$$

$$T = R \cdot \hat{Z}$$

$$R = N_T(P_f^{-1/N_T} - 1)$$

**Decizie:**

$$|X(k, n)|^2 \geq T \Rightarrow \text{Detectat}$$

## Parametri IPIX

$N_G = 3$ ,  $N_T = 12$ ,  $P_f = 0.001$  (mult mai mic decât paper!)

# De ce CFAR nu detectează „toată zona albă”?

**Observație:** De ce doar puncte localizate și nu regiuni uniforme?

**Prag global (naiv):**

- Detectează tot peste -20 dB
- $\Rightarrow$  Mii de alarme false!

**CFAR local-adaptiv:**

- Compară cu **vecinii locali**
- Zonă uniformă  $\Rightarrow$  nu detectează
- Doar **tranziții bruște** de putere

**Consecință**

CFAR detectează: margini, ridges, ținte reale – NU zone omogene de clutter!

# Adaptarea 1: K-Distribution

**Sea clutter**  $\neq$  **Gaussian** – modelează spike-uri cu K-distribution:

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

**Ajustare prag:**  $\nu$  mic  $\Rightarrow$  mai spiky. Estimare  $\nu = \mu^2 / (\sigma^2 - \mu^2)$ , ajustare multiplicator prag CFAR.



# Adaptarea 2: Fractal Boost (Hurst)

**Problema:** CFAR poate rata ținte slabe sub prag

**Observație:** Sea clutter are proprietate **fractală** (self-similar)

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

- Sea clutter:  $H \approx 0.75\text{--}0.85$  (persistent)
- Când apare țintă: structura se perturbă,  $H$  scade sub  $\sim 0.6$

## Soluție

Calculăm  $H$  per bin de frecvență. Dacă  $H < 0.6$  și putere mare  $\Rightarrow$  potențială țintă!

$$\text{mască} = \text{CFAR} \vee (\text{anomalie Hurst} \wedge \text{putere mare})$$

## Adaptarea 3: DBSCAN Asimetric

**Problema:** Țintele apar ca semnături aproape verticale în spectrogramă (energie pe multe binuri de frecvență, pe puține cadre temporale)

DBSCAN standard tinde să fragmenteze aceste semnături în mai multe cluster atunci când există discontinuități.

**Soluție:** Metrica de distanță asimetrică cu scalare pe frecvență ( $s_f = 3.0$ ):

$$d = \sqrt{(\Delta t)^2 + \left(\frac{\Delta f}{s_f}\right)^2} \quad \text{cu } s_f = 3.0$$

**Impact:** Ținta întinsă pe aproximativ 50 binuri de frecvență  $\Rightarrow$  un singur cluster coerent

**Alte adaptări:**

- Mascare componentă DC:  $\pm 8$  binuri ( $\approx \pm 31$  Hz)

# Rezultate pe Date Sintetice (Chirp Neliniar)

SNR [dB]	RQF_mean [dB]	RQF_std	Detecție [%]
5	7.28	0.47	100
10	16.81	0.60	100
15	22.95	0.56	100
20	26.40	0.51	100
25	28.43	0.39	100
30	<b>29.17</b>	0.25	100

- 100 rulări Monte Carlo per nivel SNR
- Rata de detecție: 100% în toate cazurile experimentale
- RQF crește monoton cu SNR pentru semnalul sintetic studiat

# Detecție pe Date IPIX Reale

## Baza de date IPIX (McMaster University, 1993):

- Radar X-band:  $f_{RF} = 9.39$  GHz
- PRF = 1000 Hz
- Semnale complexe I/Q
- Țintă de calibrare: sferă 1 m la distanța de 2660 m

## Relație Doppler – Viteză radială:

$$v_r = \frac{f_d \cdot c}{2f_{RF}}$$

Exemplu:  $f_d = 100$  Hz  $\Rightarrow v_r \approx 1.6$  m/s

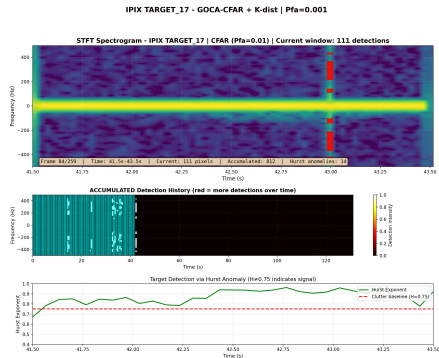
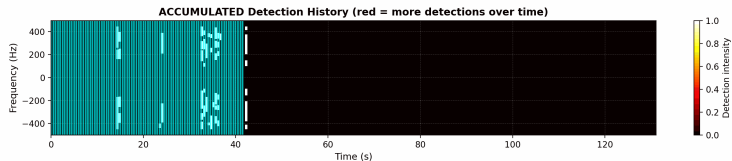
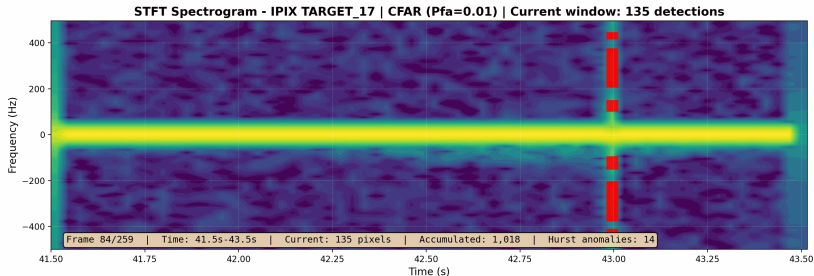


Figura: Detecție GOCA-CFAR pe Target #17

# Detecție cu Fractal Boost

IPIX TARGET\_17 - GOCA-CFAR + K-dist + Fractal |  $P_{fa}=0.001$



# Animații de Detecție (GIF-uri)

## Disponibile în repository:

- `ipix_target_17_goca_*.gif` – Detecție cu metoda GOCA standard
- `ipix_target_17_fractal_boost_*.gif` – Cu amplificare Hurst (anomalie fractală)
- `ipix_target_17_asymmetric_*.gif` – Cu DBSCAN asimetric
- `ipix_target_30_detection.gif` – Target #30 (sea state ridicat)

## Vizualizare animații

Animațiile arată evoluția detecțiilor în timp, cu suprapunerea măștilor CFAR și semnalelor detectate pe spectrograma STFT.

[https://github.com/dirgnic/Radar\\_Detection\\_STFT/tree/main/results/animations](https://github.com/dirgnic/Radar_Detection_STFT/tree/main/results/animations)

# Parametri: Paper vs. Implementare

Parametru	Paper	Sintetic	IPIX (real)
$f_s$	12.5 MSa/s	12.5 MSa/s	1000 Hz (PRF)
$N_{fft}$	512	512	256
Fereastră STFT	Gauss $\sigma = 8$	Gauss $\sigma = 8$	Gauss $\sigma = 8$
Hop	–	256	32
$N_G$	16	16	3
$N_T$	16	16	12
$P_f$	0.4	0.4	0.001
Model clutter	Gaussian	Gaussian	K-distribution
DBSCAN $\varepsilon$	–	8	8
freq_scale	–	1.0	3.0

**Modificări cheie pentru sea clutter:**  $P_f$  mult mai mic, K-distribution, DBSCAN asimetric

# Concluzii și Contribuții

- ❶ **Implementare completă** CFAR-STFT în Python cu API flexibil
- ❷ **Validare pe date sintetice:** RQF = 29.17 dB la SNR=30 dB; rata detecție 100% în 100 rulări Monte Carlo
- ❸ **Validare pe date reale:** Detecții consistente pe secvențe IPIX (Target #17, #30, #40)
- ❹ **Adaptări pentru sea clutter real:**
  - K-distribution pentru modelarea statisticilor non-Gaussiene
  - Amplificare prin anomalie Hurst pentru sensibilitate crescută la ținte slabe
  - DBSCAN asimetric pentru coerența semnăturilor verticale în timp-frecvență
- ❺ **Performanță:** aproximativ 75 ms/segment pe CPU (13 FPS pentru fereastră 2s, hop 0.5s)

## Reproducibilitate

Cod și date: [https://github.com/dirgnic/Radar\\_Detection\\_STFT](https://github.com/dirgnic/Radar_Detection_STFT)



- **Accelerare GPU:** CUDA/OpenCL pentru timp real
- **Optimizare automată:** ML pentru calibrare parametri
- **Multi-target tracking:** Predicție traiectorii
- **Integrare sistem real:** Radar operațional

**Mulțumim pentru atenție!**

Întrebări?

## Reconstruction Quality Factor:

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

- $x[n]$  = semnal original (curat)
- $\hat{x}[n]$  = semnal reconstruit
- RQF mai mare  $\Rightarrow$  reconstrucție mai bună

# Backup: Interpretare Doppler IPIX

## Two-sided spectrum:

- $f_d > 0 \Rightarrow$  țintă se apropie
- $f_d < 0 \Rightarrow$  țintă se depărtează
- $f_d \approx 0 \Rightarrow$  sea clutter

## Pentru IPIX:

$$v_{max} = \frac{PRF \cdot c}{4f_{RF}}$$
$$\approx \pm 8 \text{ m/s}$$