(GAP) Generalized Adaptive Polynomial Window Function

João F. Justo and Wesley Beccaro

Departamento de Sistemas Eletrônicos, Escola Politécnica, Universidade de São Paulo, Brazil

IEEE Access 8, 187584 (2020). DOI: 10.1109/ACCESS.2020.3030903

Outline

- Motivation and state-of-the-art
- Proposition
- Results: Mimic other window functions
- Results: Optimize window functions
- Conclusions

Motivation

- Discrete Fourier Transform (DFT) is a powerful tool to perform Fourier analysis in discrete data, with several applications, such as astronomy, chemistry, acoustics, geophysics, and digital processing.

- The use of window functions affects the analysis in the frequency domain, sometimes introducing unwanted artifacts, such as signal leakage, scalloping loss, and intensity of sidelobes.

State-of-the-art Window Functions

- The current need for better signal processing methods motivates development of improved window functions, to provide superior spectral properties.
- Recent research on window functions has focused in improving windows with flexible temporal and spectral characteristics.
- There is a demand for a more systematic procedure to develop those functions.
- We propose a generalized functional form to describe windows combined with an optimization method to improve their spectral properties.

Generalized **A**daptive **P**olynomial (GAP)

- We present a generalized window function as a non-linear polynomial expansion in which **all the current windows could be mimic with the appropriate expansion coefficients**.
- This functional form is very flexible, which allows searching for sets of expansion coefficients that provide superior properties, considering a reference figure of merit associated to the property to be improved.
- This procedure paves the way for optimization and adaptive methods, such as machine learning and genetic algorithms, to adapt window functions to certain data sets and specific applications.

The GAP Window Function

- Flexible functional form for a window function, a non-linear polynomial expansion:

$$w(t) = \sum_{n=0}^{m} \overline{a}_n t^n$$

where $\overline{a}_n$ and m are the coefficients and the order of the polynomial expansion, respectively.

The GAP Window Function

- All windows are symmetrically constrained around their center.
- Considering the polynomial represented only in the time interval$-T/2$ to $+T/2$, we kept the form:

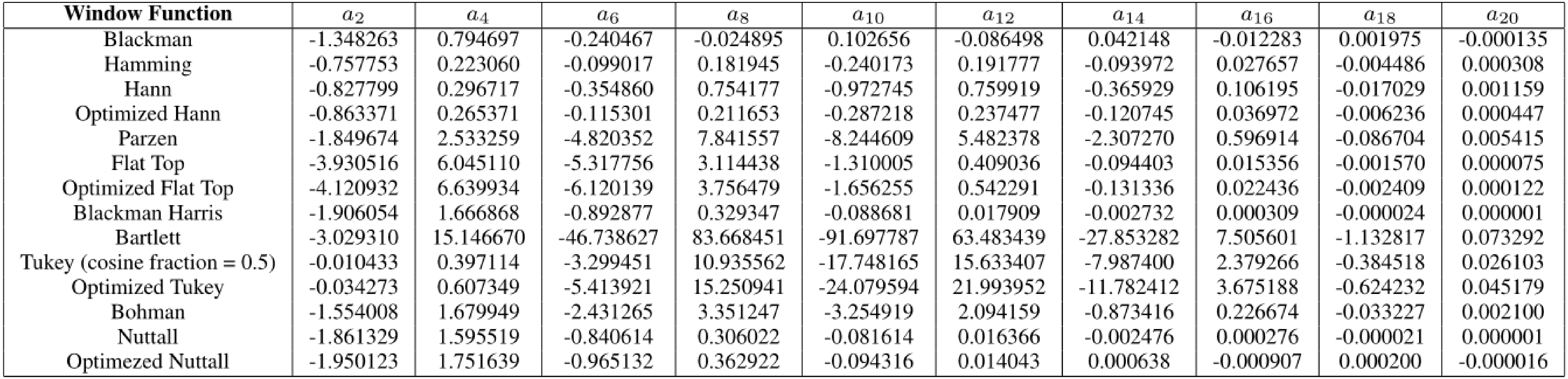
$$w(t) = a_0 + \sum_{n=1}^{m} a_{2n} \left(\frac{t}{T}\right)^{2n}, \quad \text{for } |t| \leq T/2$$

where $w(t) = 0$ for $|t| > T/2$.

- Here, we developed window functions constraining $a_0 = 1$, but this constrain could be lifted in future developments.

Mimic other window functions

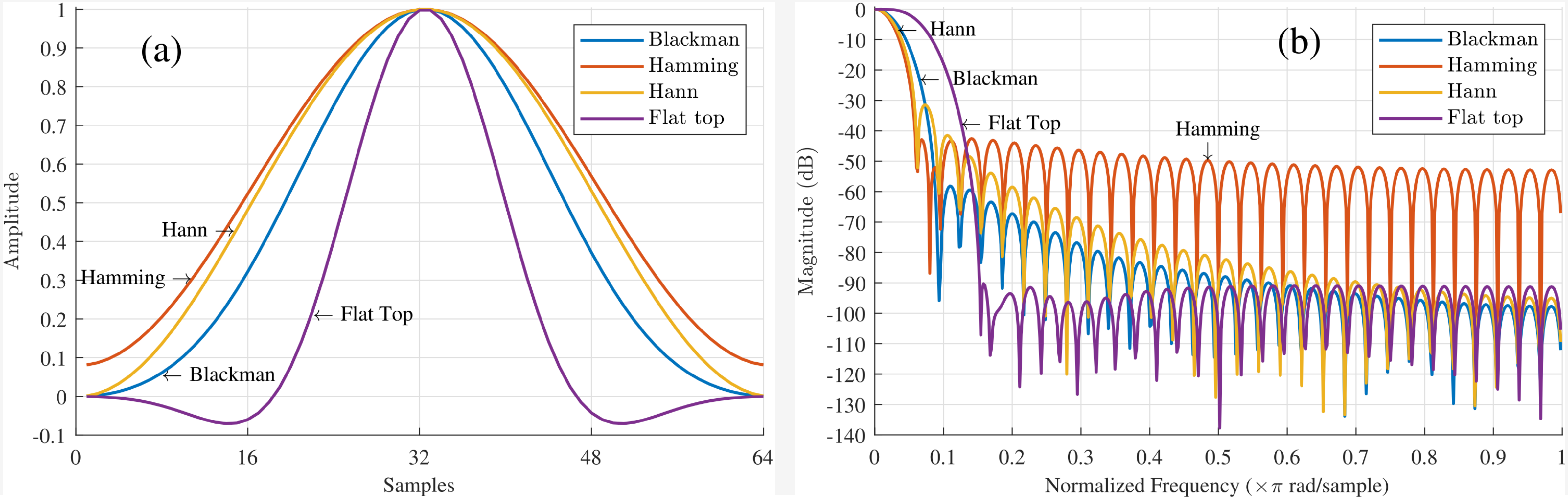
- An expansion with $m = 10$ (only ten coefficients a_2 to a_{20}) can describe well most of the traditional window functions.
- The expansion, with coefficients presented in the table, mimic any of the well-established windows and their properties.



Window Function	a_2	a_4	a_6	a_8	a_{10}	a_{12}	a_{14}	a_{16}	a_{18}	a_{20}
Blackman	-1.348263	0.794697	-0.240467	-0.024895	0.102656	-0.086498	0.042148	-0.012283	0.001975	-0.000135
Hamming	-0.757753	0.223060	-0.099017	0.181945	-0.240173	0.191777	-0.093972	0.027657	-0.004486	0.000308
Hann	-0.827799	0.296717	-0.354860	0.754177	-0.972745	0.759919	-0.365929	0.106195	-0.017029	0.001159
Optimized Hann	-0.863371	0.265371	-0.115301	0.211653	-0.287218	0.237477	-0.120745	0.036972	-0.006236	0.000447
Parzen	-1.849674	2.533259	-4.820352	7.841557	-8.244609	5.482378	-2.307270	0.596914	-0.086704	0.005415
Flat Top	-3.930516	6.045110	-5.317756	3.114438	-1.310005	0.409036	-0.094403	0.015356	-0.001570	0.000075
Optimized Flat Top	-4.120932	6.639934	-6.120139	3.756479	-1.656255	0.542291	-0.131336	0.022436	-0.002409	0.000122
Blackman Harris	-1.906054	1.666868	-0.892877	0.329347	-0.088681	0.017909	-0.002732	0.000309	-0.000024	0.000001
Bartlett	-3.029310	15.146670	-46.738627	83.668451	-91.697787	63.483439	-27.853282	7.505601	-1.132817	0.073292
Tukey (cosine fraction = 0.5)	-0.010433	0.397114	-3.299451	10.935562	-17.748165	15.633407	-7.987400	2.379266	-0.384518	0.026103
Optimized Tukey	-0.034273	0.607349	-5.413921	15.250941	-24.079594	21.993952	-11.782412	3.675188	-0.624232	0.045179
Bohman	-1.554008	1.679949	-2.431265	3.351247	-3.254919	2.094159	-0.873416	0.226674	-0.033227	0.002100
Nuttall	-1.861329	1.595519	-0.840614	0.306022	-0.081614	0.016366	-0.002476	0.000276	-0.000021	0.000001
Optimezed Nuttall	-1.950123	1.751639	-0.965132	0.362922	-0.094316	0.014043	0.000638	-0.000907	0.000200	-0.000016

Mimic other window functions

- This generalized window function could mimic (as shown in the figure) any of the well know window functions.

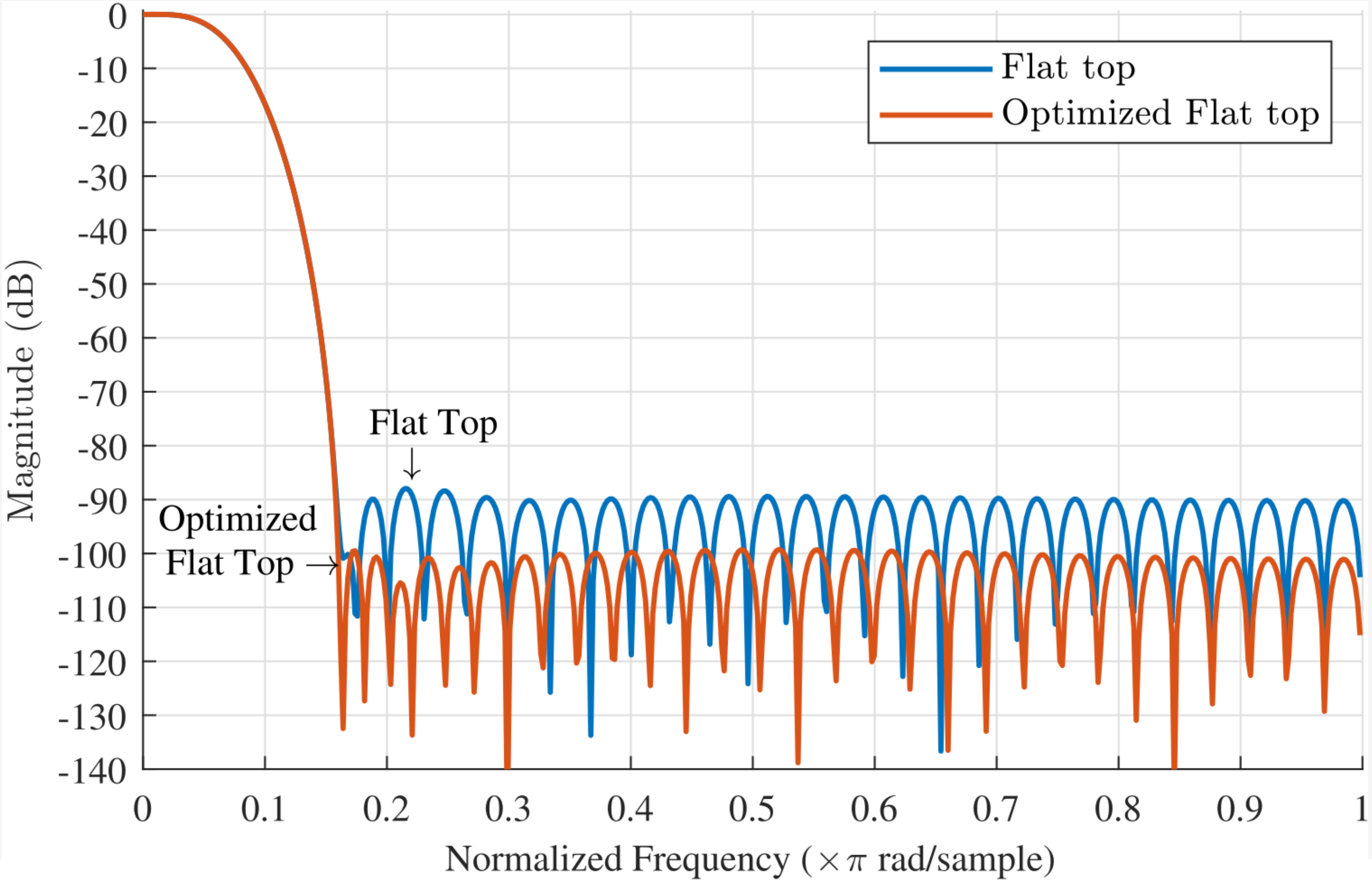


Optimization Algorithms

- Starting with a set of expansion coefficients a_i that mimics a certain window function, one can find a new window by varying those coefficients, searching iteratively to minimize a certain cost function up to a pre-determined convergence value.
- Using the a_i variables as input of the Nelder–Mead (NM) algorithm (simplex method), it is possible to find a local minimum of a side lobe measurement function (or other spectral properties).

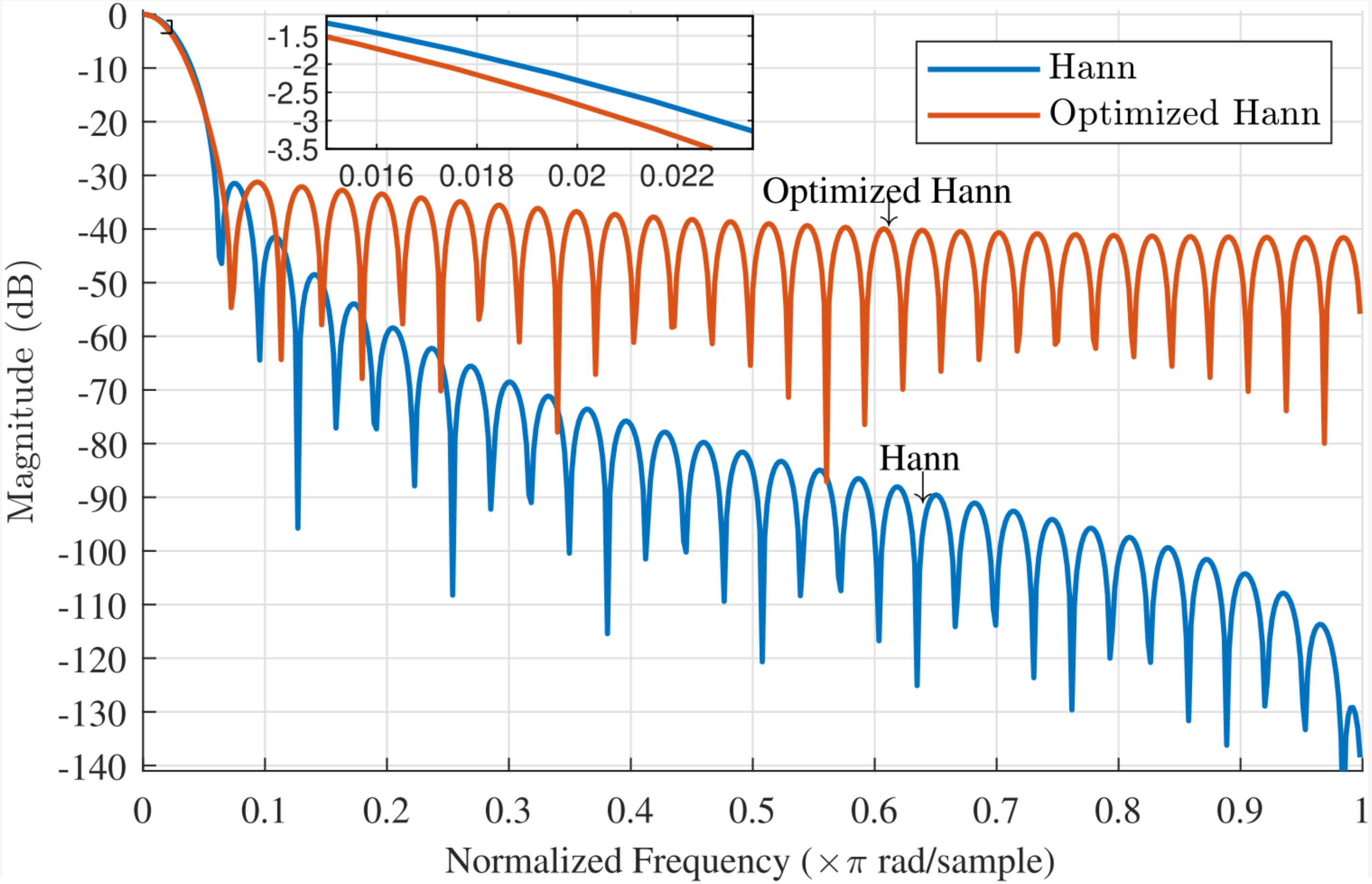
Optimization (Flat Top Window)

- Flat Top window is optimized to improve the side lobe attenuation.



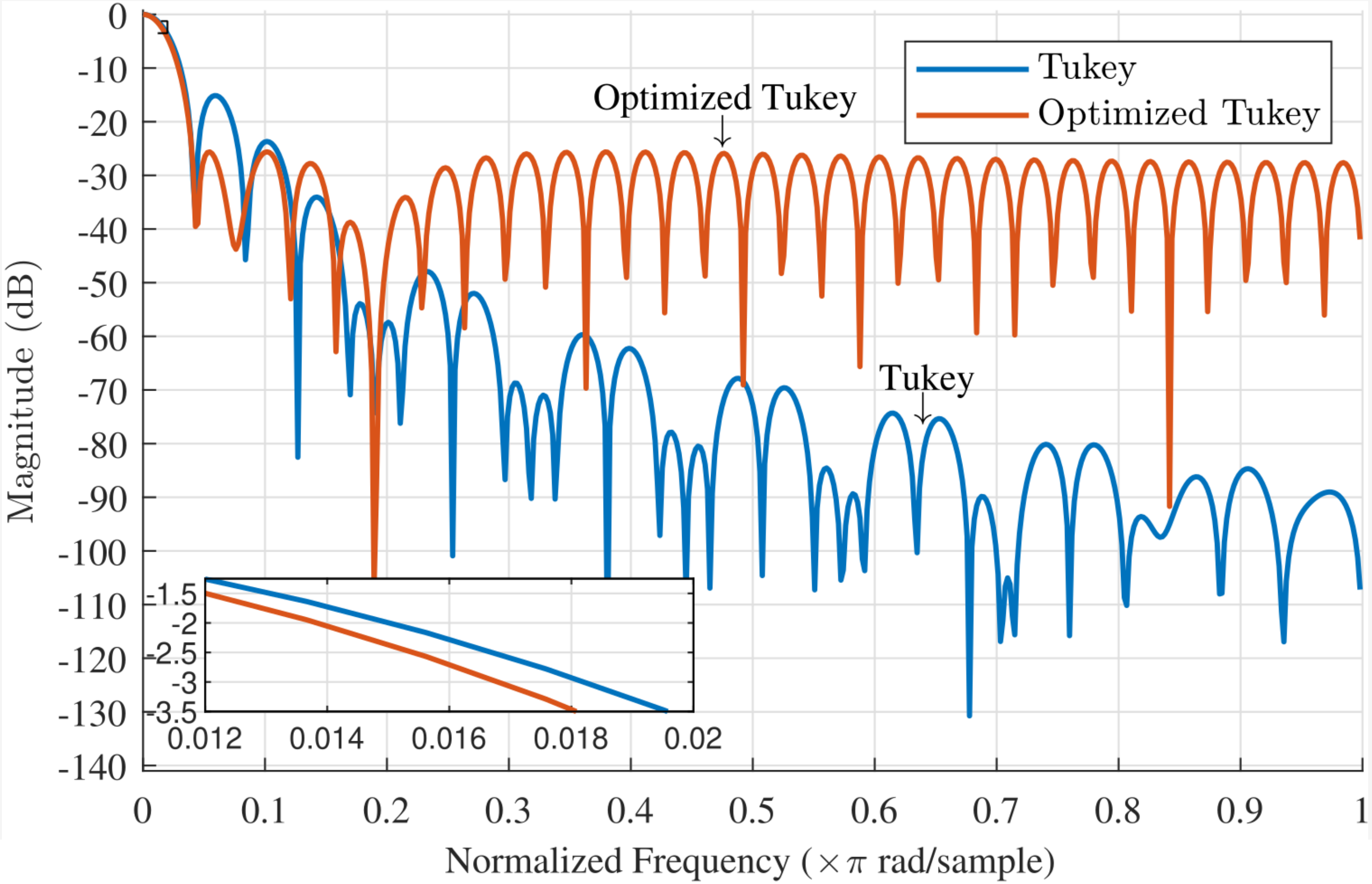
Optimization (Hann Window)

- Hann window is optimized to improve the main lobe width.



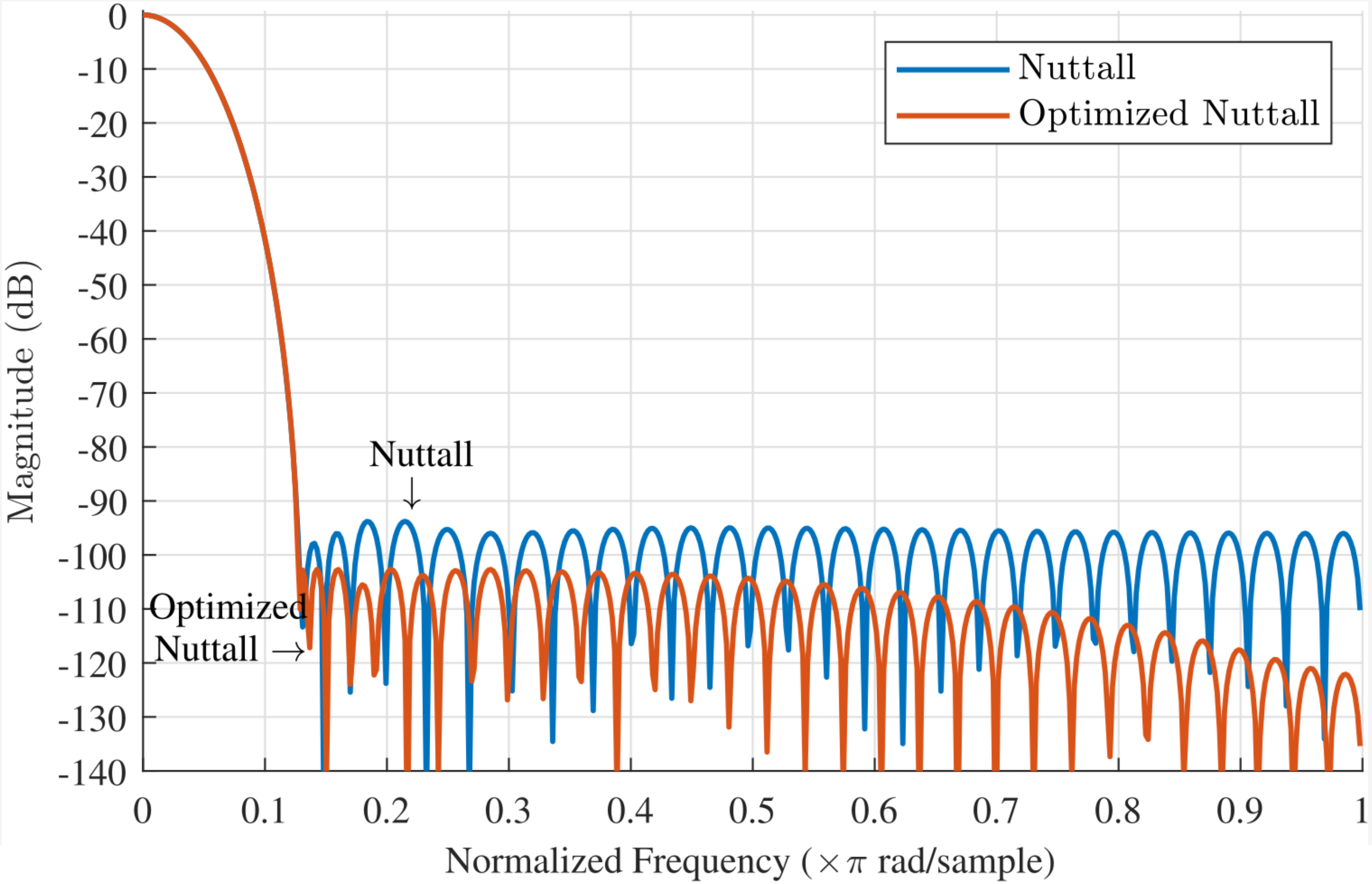
Optimization (Tukey Window)

- Tukey window is optimized to improve simultaneously the side lobe attenuation and main lobe width.

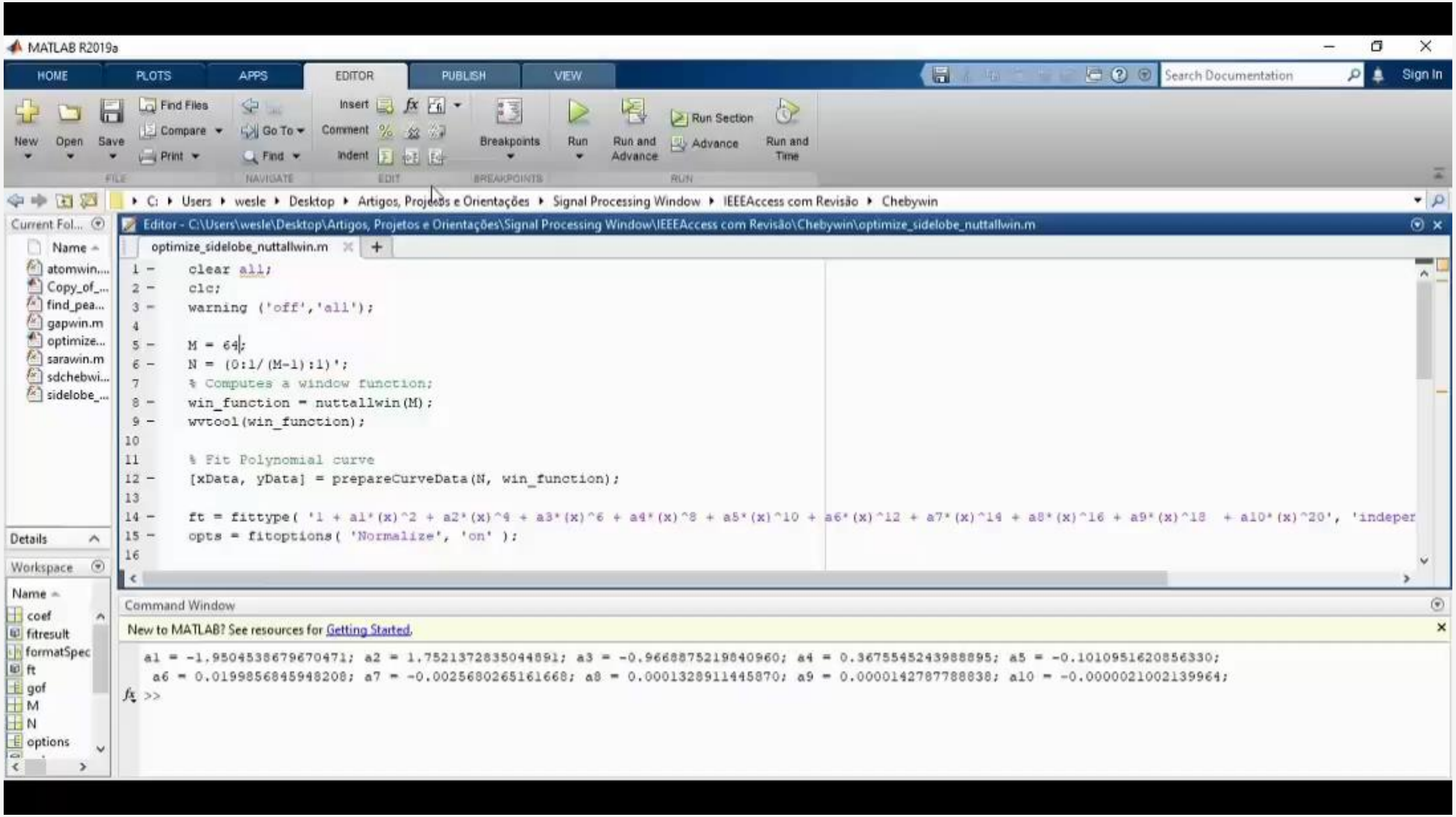


Optimization (Nuttall Window)

- Nuttall window is optimized to improve both the side lobe attenuation and main lobe width.



Optimization with MATLAB®



Editor - C:\Users\wesle\Desktop\Artigos, Projetos e Orientações\Signal Processing Window\IEEEAccess com Revisão\Chebywin\optimize_sidelobe_nuttallwin.m

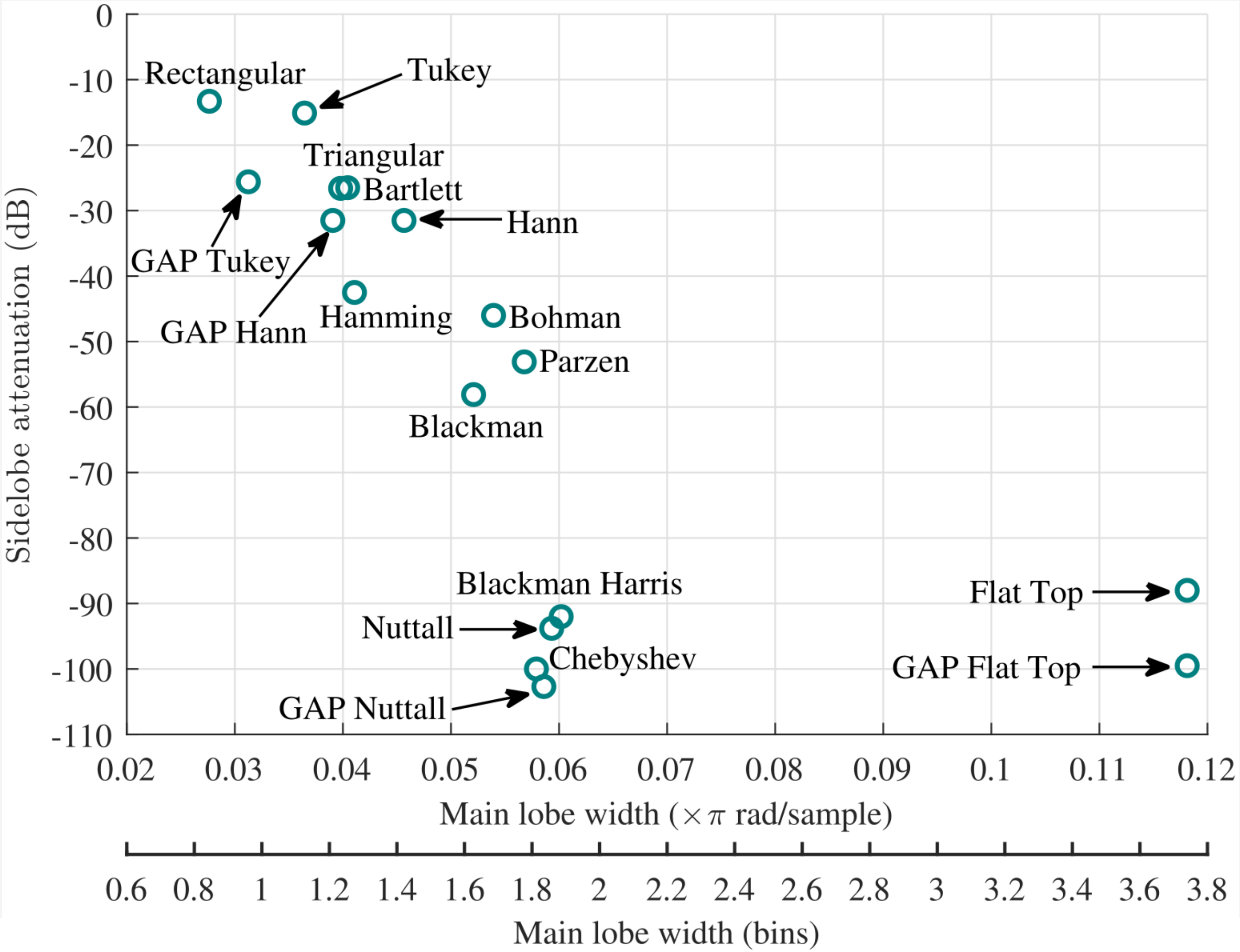
optimize_sidelobe_nuttallwin.m

```
clear all;
clc;
warning ('off','all');

M = 64;
N = (0:1/(M-1):1)';
% Computes a window function;
win_function = nuttallwin(M);
wvtool(win_function);

% Fit Polynomial curve
[xData, yData] = prepareCurveData(N, win_function);

ft = fittype( '1 + a1*(x)^2 + a2*(x)^4 + a3*(x)^6 + a4*(x)^8 + a5*(x)^10 + a6*(x)^12 + a7*(x)^14 + a8*(x)^16 + a9*(x)^18  + a10*(x)^20', 'indeper
opts = fitoptions( 'Normalize', 'on' );

```

New to MATLAB? See resources for Getting Started.

```
 a1 = -1,9504538679670471; a2 = 1.7521372835044891; a3 = -0.9668875219840960; a4 = 0.3675545243988895; a5 = -0.1010951620856330;
  a6 = 0.0199856845948208; a7 = -0.0025680265161668; a8 = 0.0001328911445870; a9 = 0.0000142787788838; a10 = -0.0000021002139964;
>>
```

Optimization of Spectral Properties (General Results)



Conclusions

- This method to obtain windows is quite general, allowing the use of several optimization methods, such as global optimization (genetic algorithms and simulated annealing) or local optimization (Newton and gradient-based methods) techniques, or even machine learning.
- Any new window obtained by optimization procedures represents an improvement of the properties in the frequency domain, when compared to that initial window function guess.
- This method allows to improve several spectral properties simultaneously.

Contact and link to software (MATLAB® and Python)

- IEEExplore:

https://ieeexplore.ieee.org/document/9223641

- GITHUB:

https://github.com/EmbDSP/GAP/

- MATLAB File Exchange:

https://www.mathworks.com/matlabcentral/fileexchange/81658-gap-generalized-adaptive-polynomial-window-function

- Contact: wesley@lme.usp.br