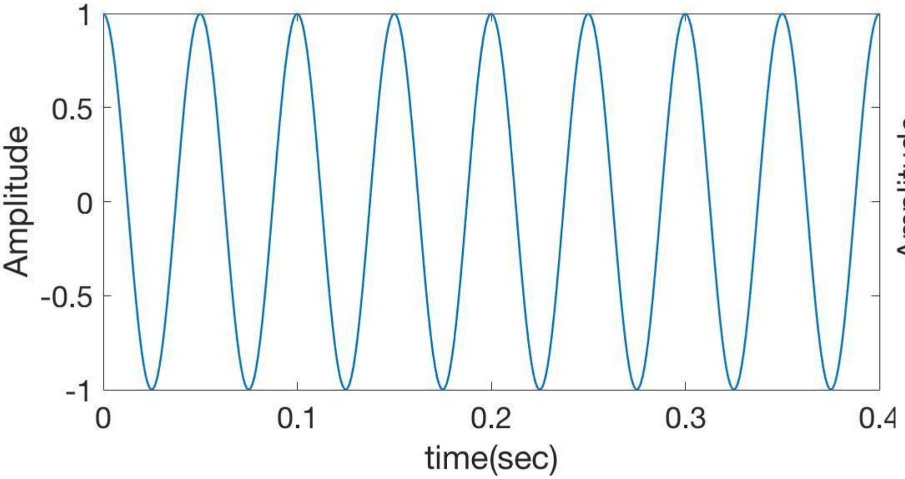
Study of Denoising of Sounds and Images

Yeonho Jung

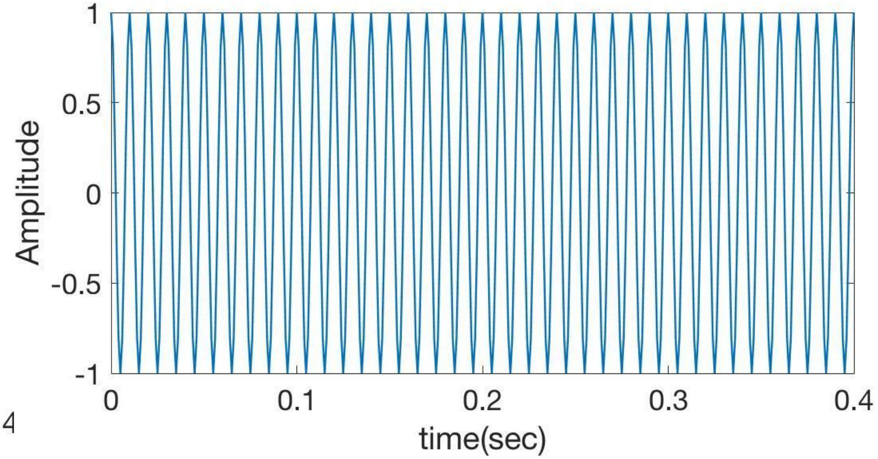
Denoising Sound and Images

- **Sound**
  - Purpose
    - Noise obscuring voice
    - Inaudible
  - Objective: Decrease **amplitude of noise** and increase **amplitude of voice**
  - Method: Different **Filters** and **Windows**
- **Images**
  - Purpose
    - Anecdote
  - Objective: Isolate the **k-space** to expedite the process
  - Method: Different **Filters**

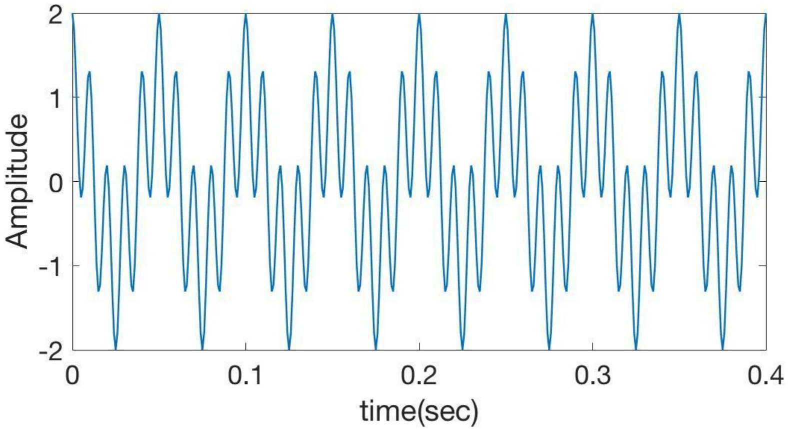
Denoising Sound - Original Sound Clip



Voice



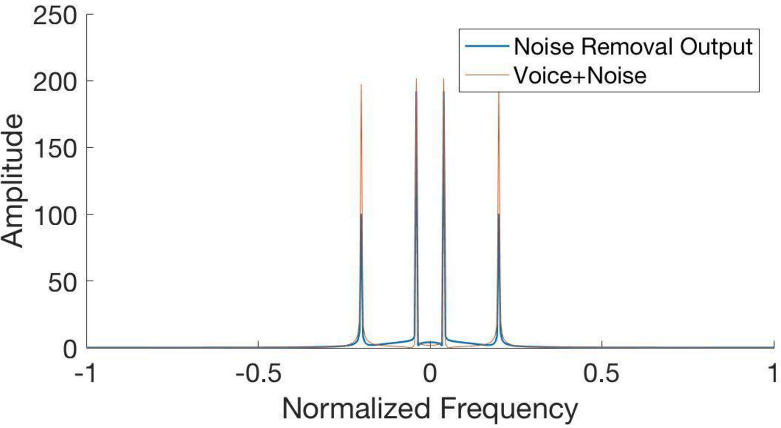
Noise



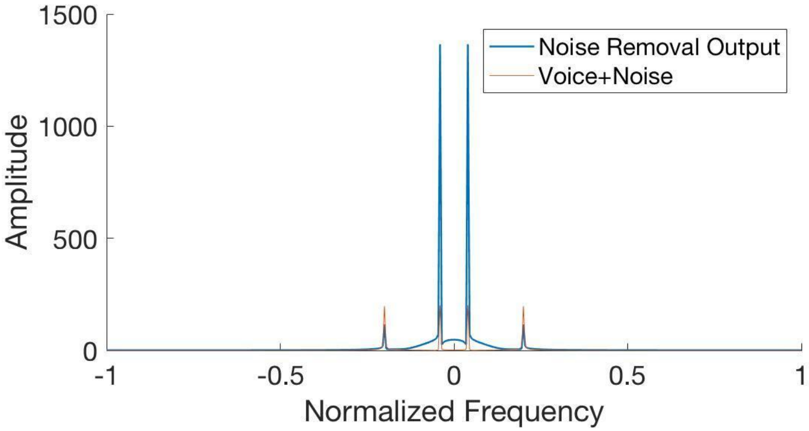
Voice + Noise

*Images are produced by student - by using MatLab

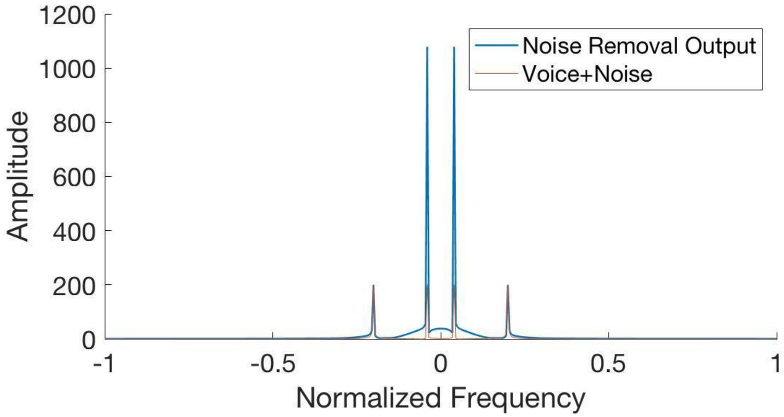
Denoising Sound - Transformed



Finite Impulse Response (FIR) Filter Only



Hamming Window Only



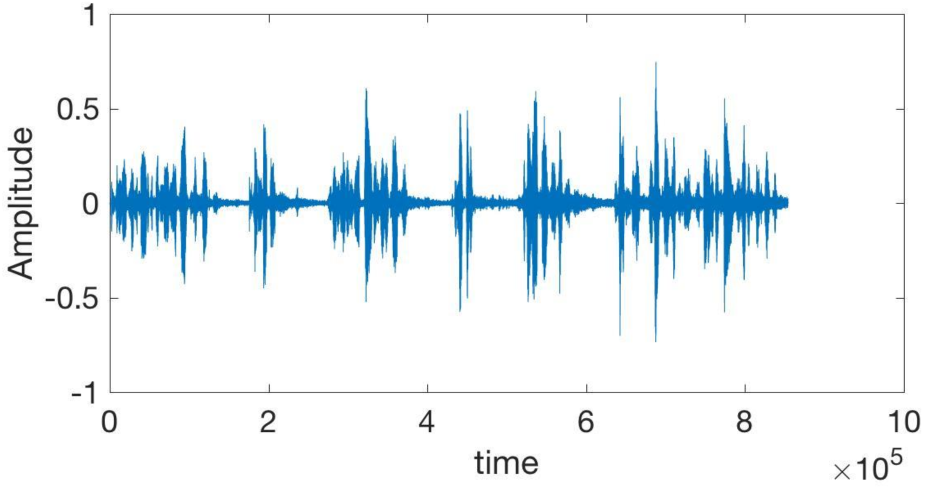
FIR Filter + Hamming Window

Key Terms: (1) **Fourier Transform;** (2) **Finite Impulse Response Filter;** (3) **Hamming Window**

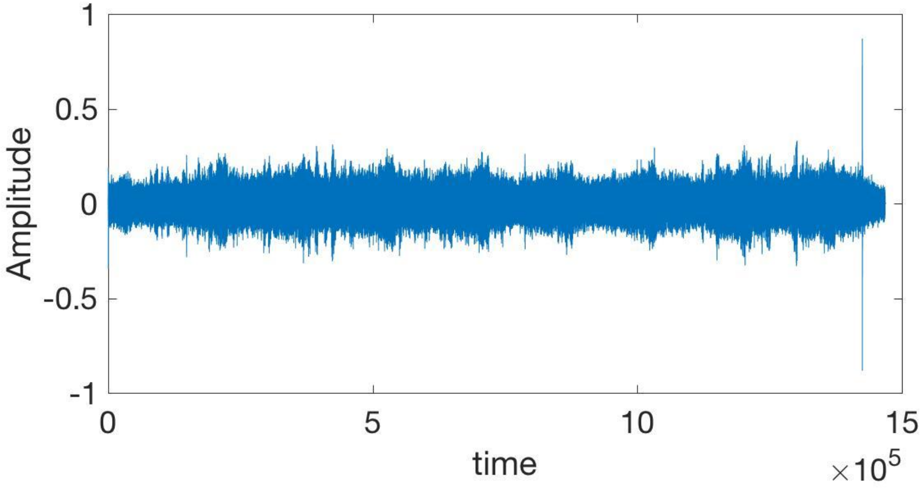
*Images are produced by student - by using MatLab

Realistic Sample Voice + Vacuum Cleaner Noise

< Only Voice >



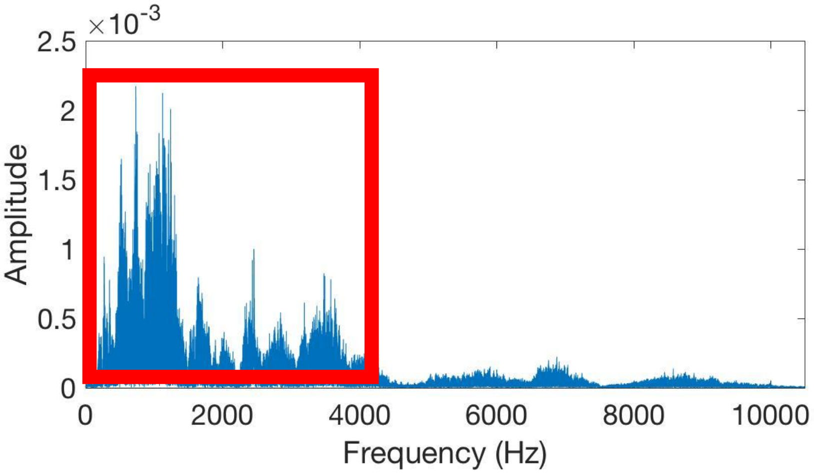
< Voice + Noise >



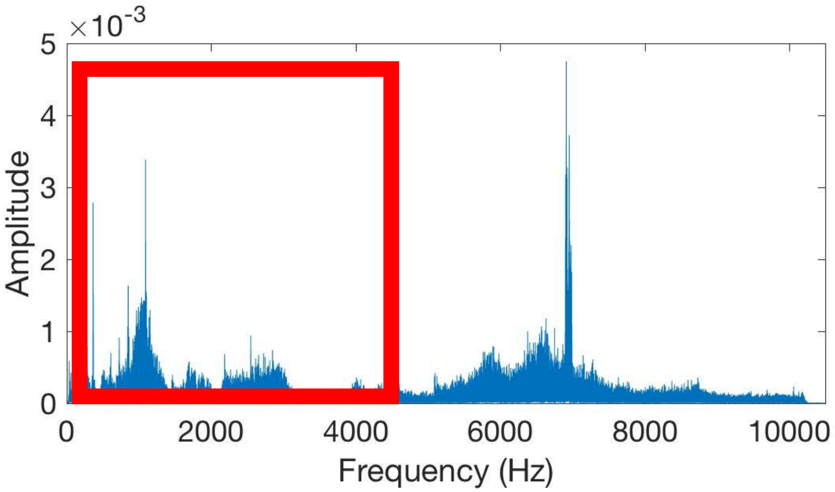
*Images are produced by student - by using MatLab

Realistic Sample Voice + Vacuum Cleaner Noise

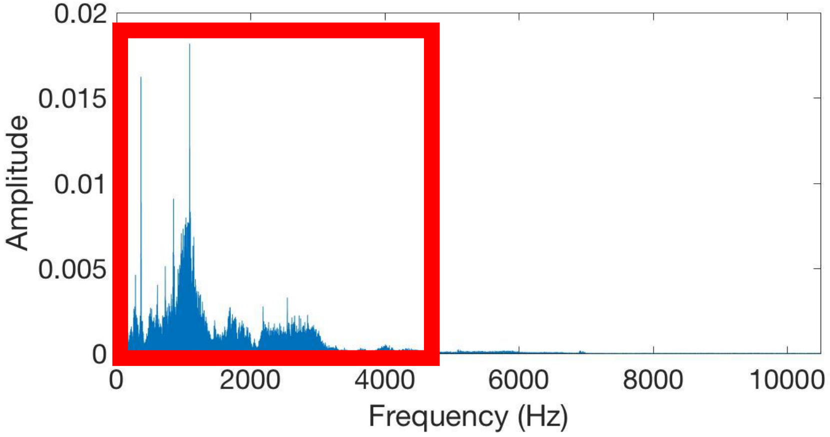
< Voice Only >



< Voice + Noise >



< Hamming Window + FIR Filter >



Voice + Noise Audio

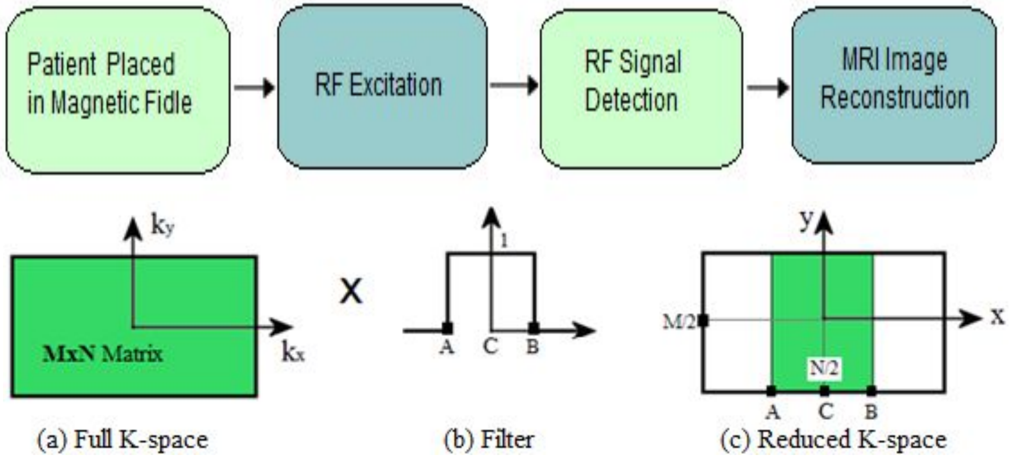
Just Window

Window + Filter

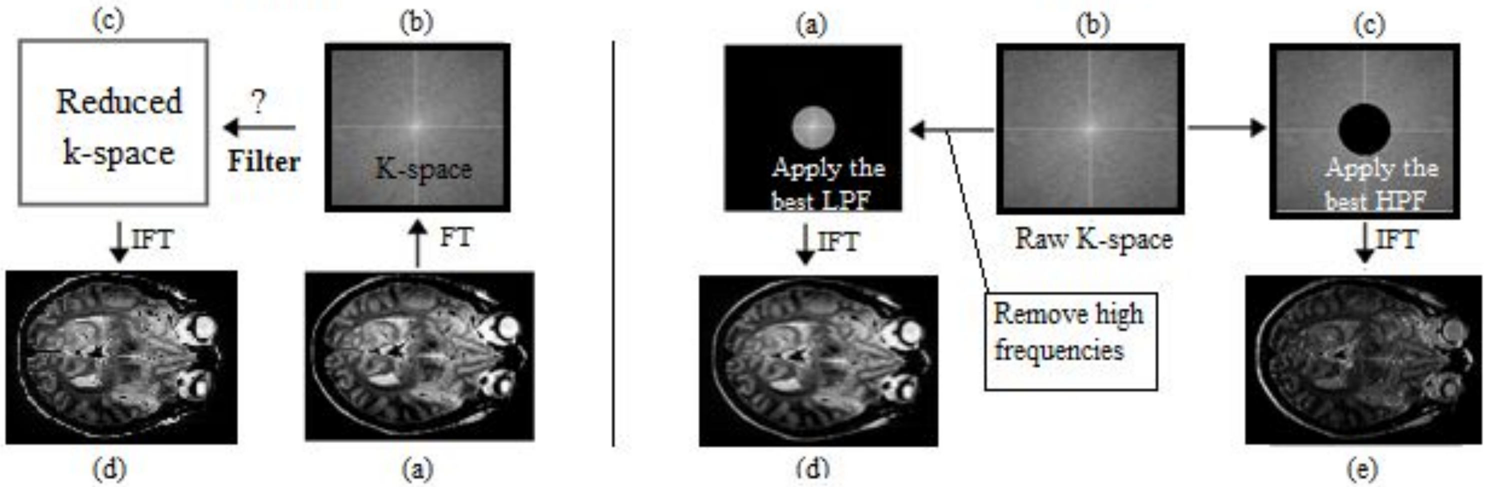
*Images are produced by student - by using MatLab

Denoising Images

- **Inverse Fourier Transformation** is computationally **costly**
- Ridding unnecessary information will **expedite** the process



*Images are produced from student - by using Paintbrush



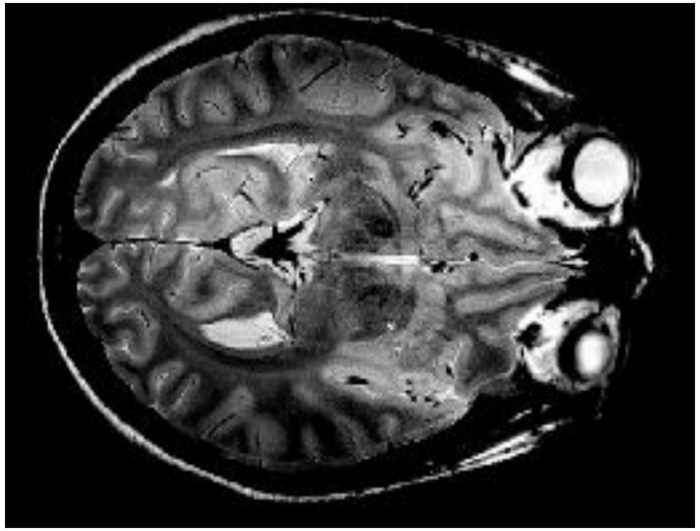
(A)

(B)

*Images (A)-a, (A)-d, (B)-d, and (B)-e are from MRI Research lab in UIUC) and student using MatLab. Others(images on top) are from Google.

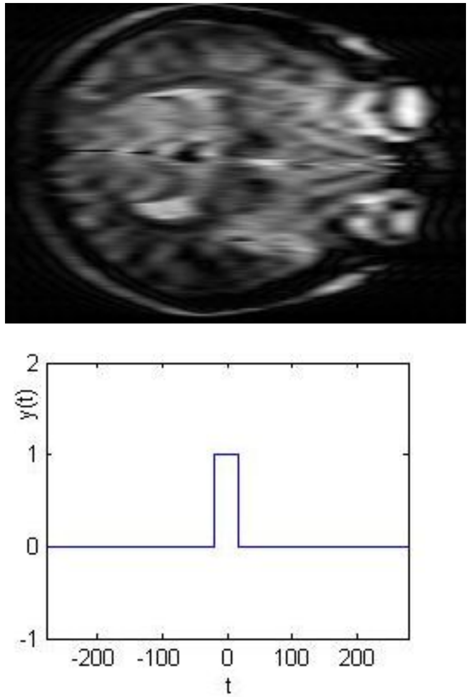
Gaussian = a*e-x^{2}k^2; k is an integer

Denoising Images - Existing Methods

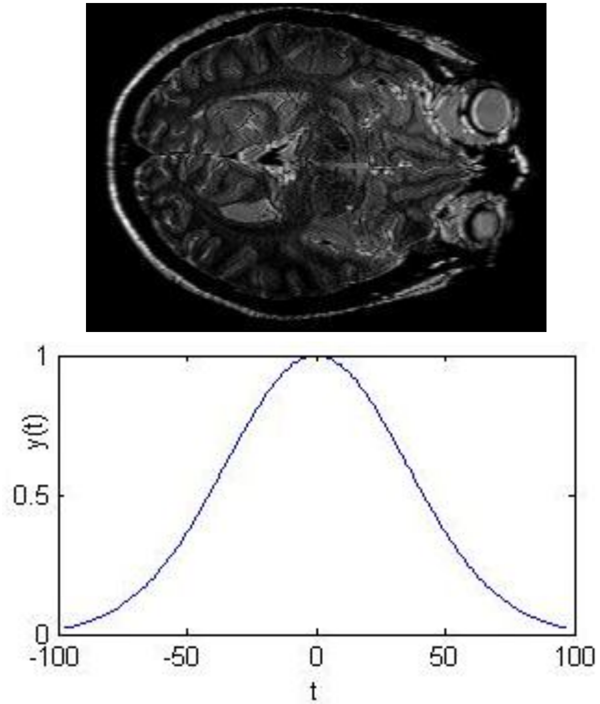


Initial MRI Scan
(MRI Research lab -in UIUC)

*Images and graphs on the right are produced from student -by using MatLab

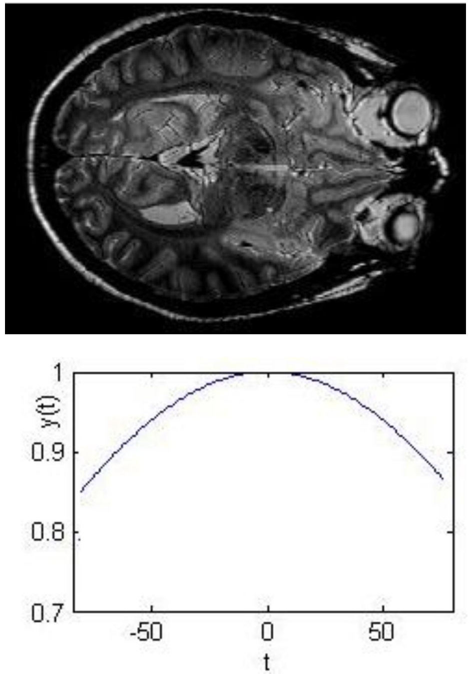


Implementations of **Boxcar**
Observation: Blurry



Implementation of **Gaussian** with A=1
and **k=50**
Observation: Dark

$y=A \exp(-x^2/k^2)$



Implementation of **Gaussian** ith A=1
and **k=60**
: Lighter

$y=A \exp(-x^2/k^2)$

Denoising Images - Proposed Method

$x^n + y^n = 1$

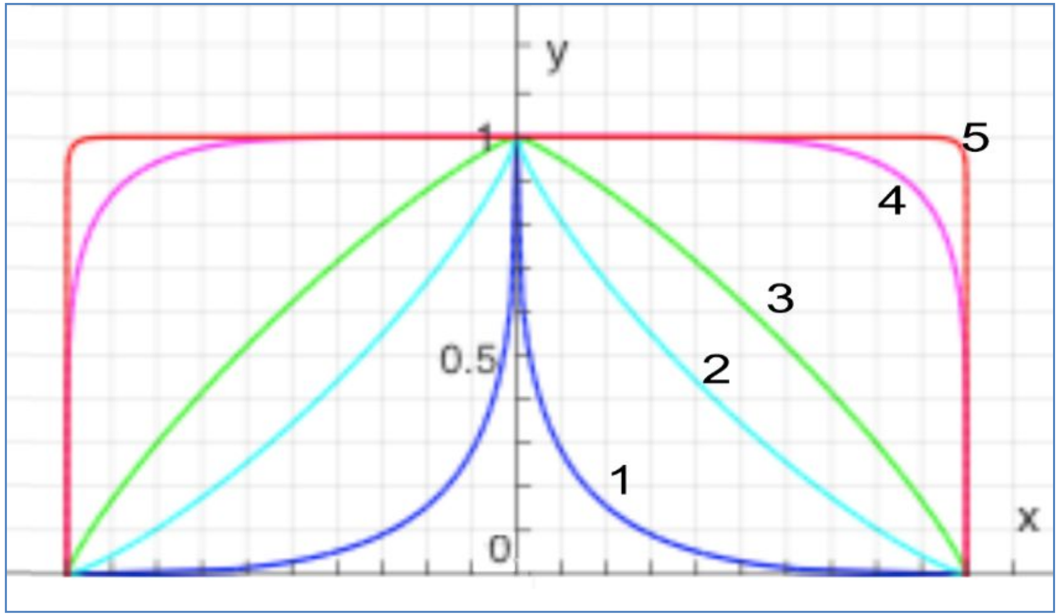


*Image is produced by student -by using MatLab

Different types of graphs that can be potentially used as LPF were considered. Powers with the fractions with an even numerator and an odd denominator are continuous throughout all four quadrants. The equation is:

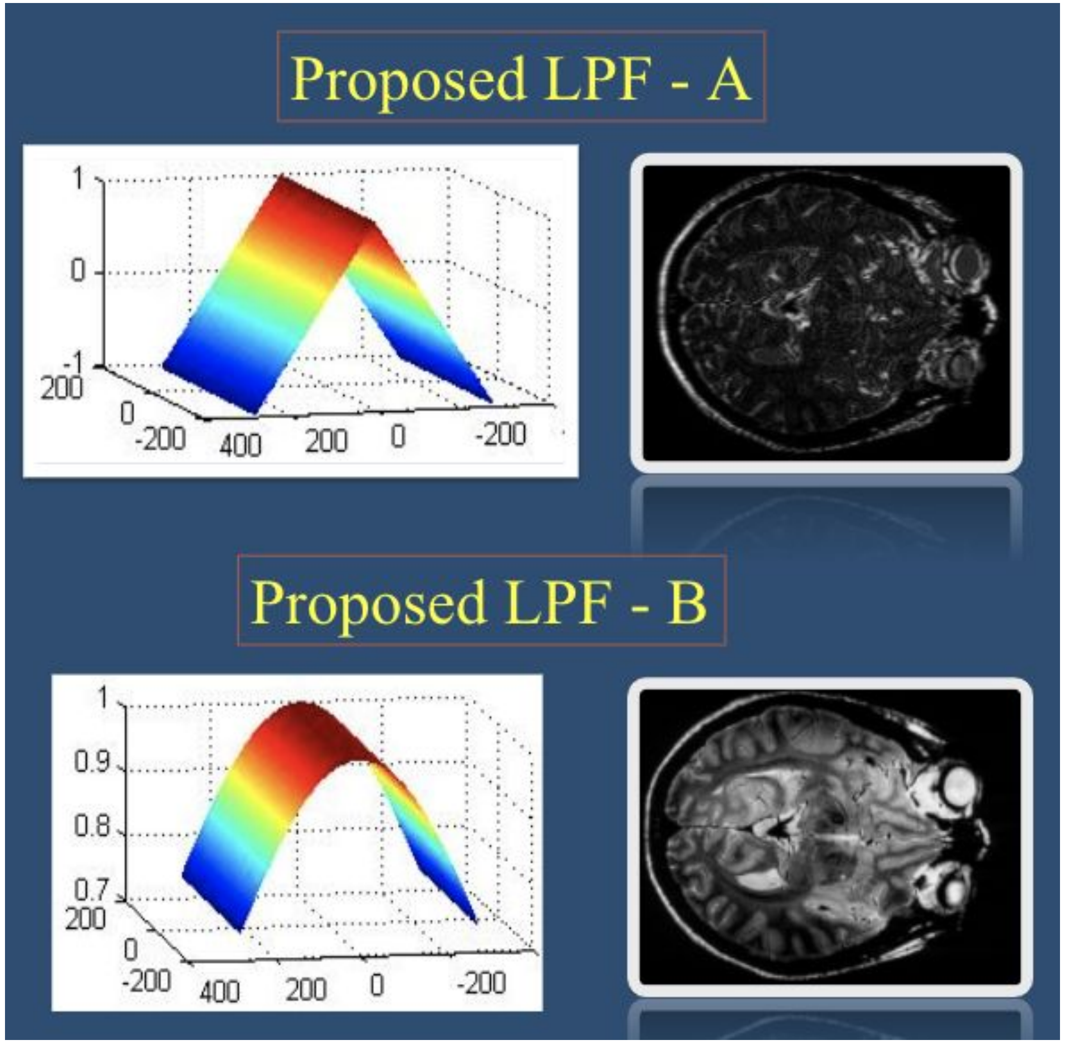
$$x^n + y^n = 1$$

When the power n is less than 1, the graph caves inwards. When the power n is greater than 1, the graph curves outward, and finally, it becomes a rectangular-shaped graph as n becomes infinite.

Tested Mode:

- **$y = 1 - |x/(\text{width}/2)|^n$**
- n=1~10, width=516~1024

Denoising Images - Proposed Method



- **LPF A**
  - $y = 1 - |x/(width/2)|^n$
  - n=1, width=516
  - Dark but clear edges
  - Hard to make out

- **LPF B:**
  - $y = 1 - |x/(width/2)|^n$
  - n=6, width=516
  - Light
  - Easy to see

*Images are produced by student -by MatLab

Conclusion

- **Sound**
  - The effectiveness of the filter and the windows through comparison of pure sample, pure voice, and noise removal outputs.
  - Application of **proposed LPF** reduced the amplitude of pure noise, while application of windows augmented amplitude of pure voice.
  - Through applying both proposed LPF and existing windows, we can execute effective noise removal.

- **Image**
  - This research tested variations of three types of filter functions:
  - (1) **Rectangular** Function: produced image with ringing artifact
  - (2) **Gaussian** Function: produced image with high resolution – clear tissue region or borders depending on width of function
  - (3)**Proposed** Function: produced image with high resolution – clear borders

Thank you