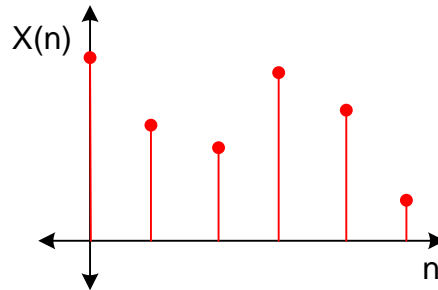


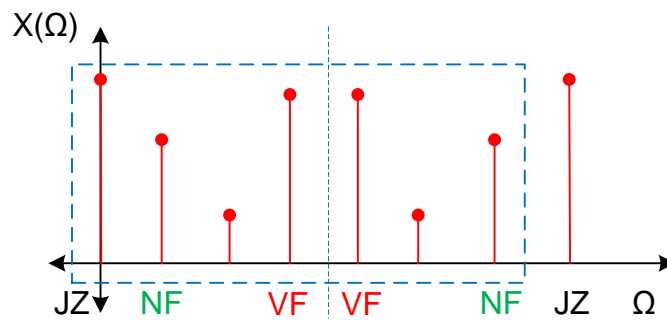
2D filtrácie

Úvod do problematiky:

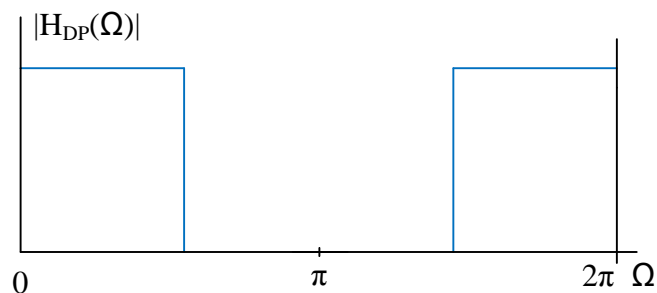
Podme si najprv zopakovať znalosti 1D DFT. Čiže máme nejaký navzorkovaný signál v čase:



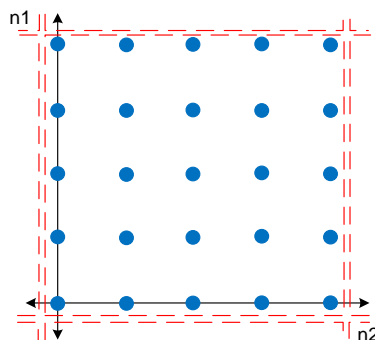
Z tohto signálu dostaneme po DFT nejaké spektrum. Ako si iste spomínate, tak spektrum je symetrické. Máme 7 vzoriek v našom signále, tak aj spektrum bude pozostávať zo 7 vzoriek, ktoré sa budú periodicky opakovať, pričom vzorky sa budú zrkadliť po polovici. Napr. takéto niečo:



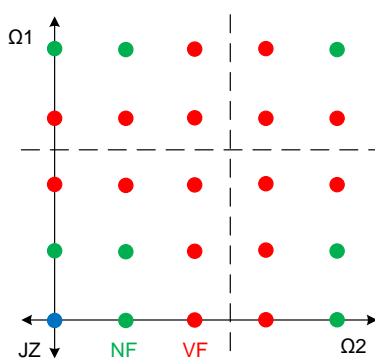
Ako vidíte a viete, vysoké frekvencie predstavujú prostredné zložky, preto ak chceme odfiltrovať napr. vysoké frekvencie, hodíme na to DP filter, ktorý ako si spomínate má takúto prenosovku, z čoho si môžete predstaviť aj vzťah medzi známymi digit. filtermi a skutočnými frekvenciami vypočítanými pomocou DFT.



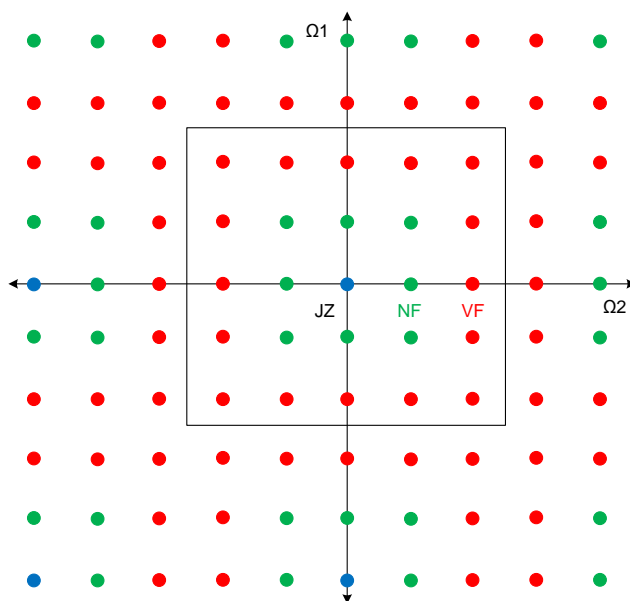
Teraz si zoberme nejaký klasický 2D signál, ktorý sa tiež periodicky šíri.



Spektrum takéhoto signálu má pochopiteľne tiež rovnaké rozmery, a zároveň si tiež ukážme ktoré vzorky predstavujú aké frekvencie, nebudeme rozkresľovať všetko, čitateľ si domyslí. Čierne osi predstavujú 2D symetrie, aké poznáme z DFT.

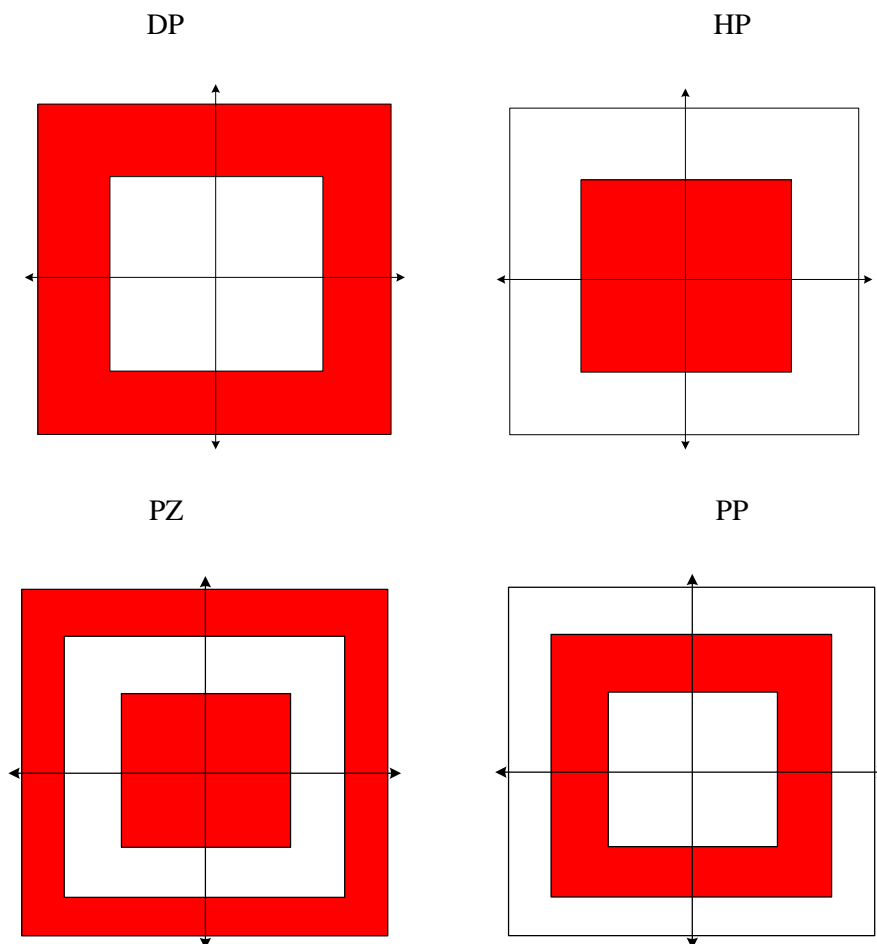


S takýmto niečím sa už dá trochu narábať čo sa týka filtrácie, ale skúsme si to ešte pred samotnou aplikáciou filtrov trochu prekresliť. Keď si spektrum zakreslíme periodicky, vidíme, že takýmto usporiadaním, sme jednoducho zoskupili nízke a vysoké frekvencie vzhľadom na stred. Z nich vyberieme našu veľkosť signálu, 5x5, a na to je teraz jednoduchšie hodiť filter.



A práve takéto posunutie „okna“ sa využíva pri 2D filtráciách. Keď pri spätnom procese po vyfiltrovaní znovu posunieme „okno“ na pôvodné miesto, vďaka periodicite останú vyfiltrované presne tie zložky čo treba.

Masky pre rôzne 2D pravouhlé filtre by teda vyzerali takto (červené sa odfiltruje):



Prax:

Takže teraz si to vyskúšajte v matlabe. Postup:

- 1) Načítať obrázok 256x256 (bmp, tiff, png, pri najhorsom jpg) – príkaz : `imread('názov.typ')`
(je dosť možné, že budete musieť upraviť na matlabovský grayscale – `rgb2gray(premenná)`)
- 2) Urobiť spektrum a posunúť do stredu, podľa teórie – príkaz : `fftshift(spektrum)`
- 3) Navrhnuť filter
- 4) Prefiltrovať obrázok
- 5) Späťne posunúť a transformovať – príkaz : `ifftshift(spektrum)`
- 6) Zobrazit' výsledok filtrácie

Praktické ukážky výsledkov filtrácii:

V 1D signáloch je jasné čo sú nízke a čo vysoké frekvencie. V 2D signáloch si predstavte napr. čierny kruh na bielom pozadí. Čierna a biela plocha sú nízke frekvencie a ten ostrý prechod medzi nimi sú frekvencie vysoké. **Nízke frekvencie predstavujú teda plochy, a vysoké frekvencie hrany.**

DP filter prepúšťa len nízkofrekvenčné zložky (vzhľadom na návrh). Keď teda odfiltrujeme vysoké frekvencie, spôsobíme tým ako keby **rozostrenie hrán** ako to môžete vidieť na obrázkoch. Pri efektívnom návrhu by ste sa mohli dopracovať až k niečomu ako je na obr. dole.



Naopak HP filter prepúšťa vysoké frekvencie, a teda odfiltrujú sa plochy, a tým dôjde k **zvýrazneniu hrán** ako vidíte na obr.



PZ a PP v sebe obsahujú (podľa návrhu) aj vysoké aj nízke frekvencie, preto interpretáciu necháme na čitateľovi. Príklady :

