ECE 421
Introduction to Signal Processing

Dror Baron
Assistant Professor
Dept. of Electrical and Computer Engr.
North Carolina State University, NC, USA

Sampling, Reconstruction, and More

Roadmap

- We have seen
  - Chapter 1 – from analog to digital and back
  - Chapter 2 – discrete time signals & systems; correlation
  - Chapter 3 – z-transforms; transfer functions; one-sided z
  - Chapter 4 – Fourier transforms
  - Chapter 5 – frequency domain analysis of LTI systems
- *About to discuss Chapter 6*
  - Sampling, aliasing, and reconstruction
  - End to end system design
  - Signal denoising (modern topic)
  - Sampling bandpass signals → compressed sensing (modern topic)

Sampling, Aliasing, & Reconstruction

[Reading material: Section 6.1]

Let's recap

- We surveyed

1) *Sampling*: $x(n)=x_a(nT)$
2) *Aliasing*: if we sample below Nyquist, higher frequencies get mixed in erroneously
3) *Reconstruction*: $x_a(t) = \sum_{n=-\infty}^{+\infty} x(n) \frac{sin\left(\frac{\pi}{T}(t-nT)\right)}{\frac{\pi}{T}(t-nT)}$

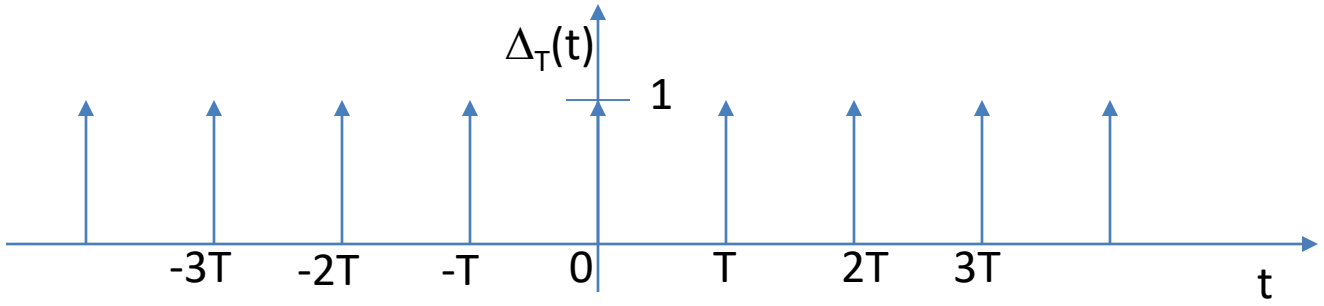
- Have covered enough math content to discuss in greater detail

Survey

[Personal perspective]

Impulse train perspective

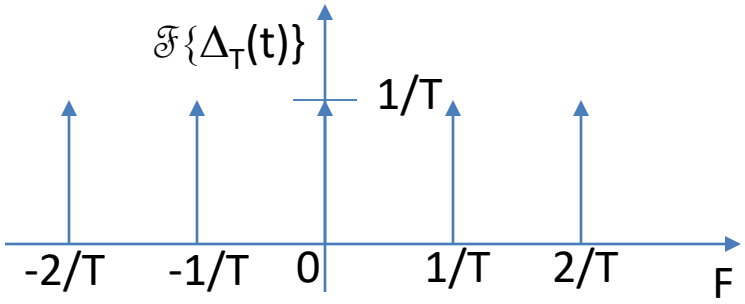
- Have discussed sampling from continuous to discrete time
- Consider impulse train $\Delta_T(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT)$
- Individual delta $\delta(t - nT)$ at time nT
- Summing over all n yields spacing T between them



Fourier perspective on impulse trains

- *Fourier transform of impulse train also impulse train*

$$\mathcal{F}\{\Delta_T(t)\} = \frac{1}{T}\sum_{k=-\infty}^{+\infty}\delta\left(F - \frac{k}{T}\right) = \frac{1}{T}\Delta_{\frac{1}{T}}(F)$$



- Will derive in detail later

Fourier perspective on impulse trains

- *Fourier transform of impulse train also impulse train*

$$\mathcal{F}\{\Delta_T(t)\} = \frac{1}{T}\sum_{k=-\infty}^{+\infty}\delta\left(F - \frac{k}{T}\right) = \frac{1}{T}\Delta_{\frac{1}{T}}(F)$$

- Multiplying $x_a(t)$ by $\Delta_T(t)$ samples input at multiples of T

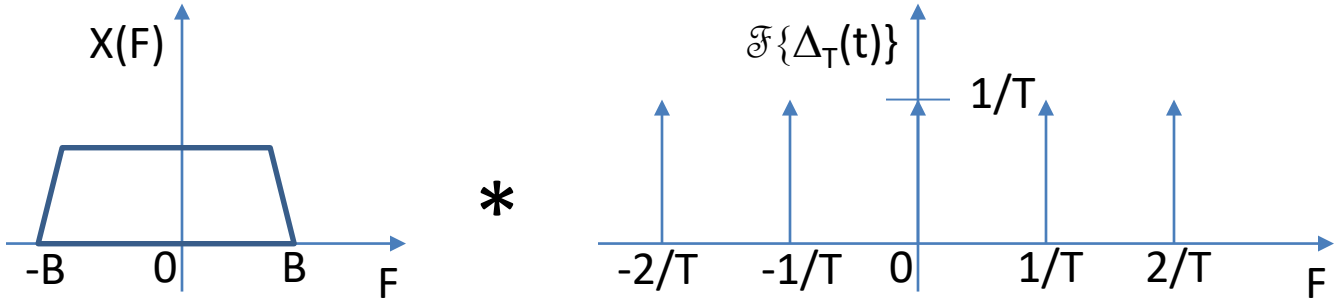
$$x_a(t)\Delta_T(t) = xa(t)\sum_{n=-\infty}^{+\infty}\delta(t-nT) = \sum_{n=-\infty}^{+\infty}x_a(nT)\delta(t-nT)$$

- "Information" in $x_a(t)\Delta_T(t)$ identical to that in $x(n)=x_a(t=nT)$

Fourier perspective on sampling

- What's Fourier transform of sampling (product)?

$$\mathcal{F}\{x_a(t)\Delta_T(t)\} = \mathcal{F}\{x_a(t)\} * \mathcal{F}\{\Delta_T(t)\} = X(F) * \frac{1}{T}\Delta_{1/T}(F)$$



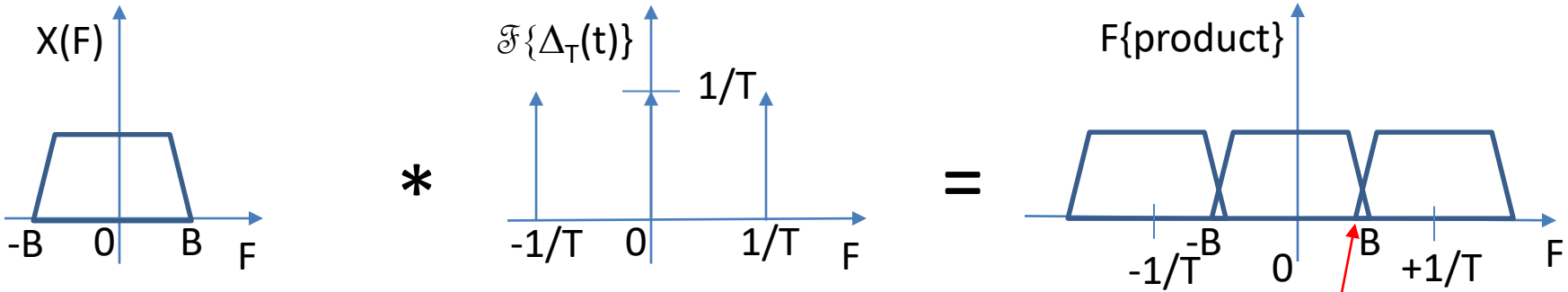
- Recall $x(t)*\delta(t)=x(t) \rightarrow x(t)*\delta(t-nT)=x(t-nT)$
  - Delta delayed by nT "picks off" signal at time nT

- Sampling: $X(F) * \left\{\frac{1}{T}\sum_{k=-\infty}^{+\infty}\delta\left(t-\frac{k}{T}\right)\right\} = \frac{1}{T}\sum_{k=-\infty}^{+\infty}X\left(F-\frac{k}{T}\right)$

Aliasing

Aliasing

- Sampling convolves X(F) with $\frac{1}{T}\Delta_{\frac{1}{T}}(F)$
- Graphical interpretation: copies of X(F) shifted to left/right by 1/T (and multiplied by 1/T)

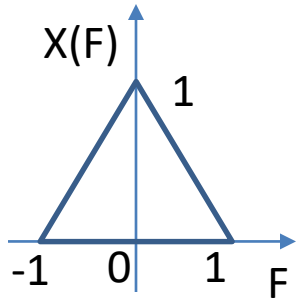


- Three cases:

1) 1/T>2B → copies don't overlap → no aliasing
2) 1/T<2B → copies overlap (our illustration) → aliasing
3) 1/T=2B → copies touch → sounds nice, don't try at home

Active learning

- Consider signal $x_a(t)$ with following Fourier
- Sketch $\mathcal{F}\{x_a(t)\Delta_T(t)\}$ for three cases

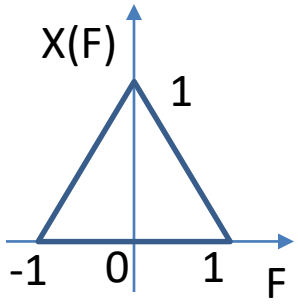


a) T=1/2 (sampling at Nyquist)

b) T=1/5 (sampling above Nyquist)

Active learning continued

c) T=1 (sampling below Nyquist → aliasing)



Comments

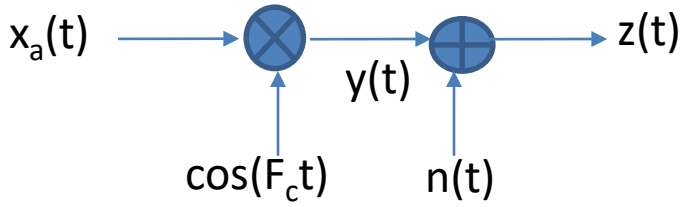
- In past, sampling was considered intuitively
- We understood that sampling sinusoids below 2/cycle was bad

- New mathematical understanding makes things well-defined
  - Shifts of Fourier copied left/right
  - Don’t want shifts to intersect/overlap

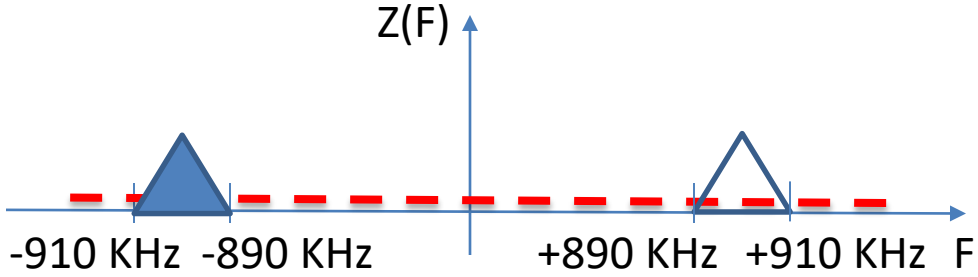
AM Demodulation Revisited

Previous setting

- Input $x_a(t)$ modulated by cosine with noise n(t), z(t)=y(t)+n(t)

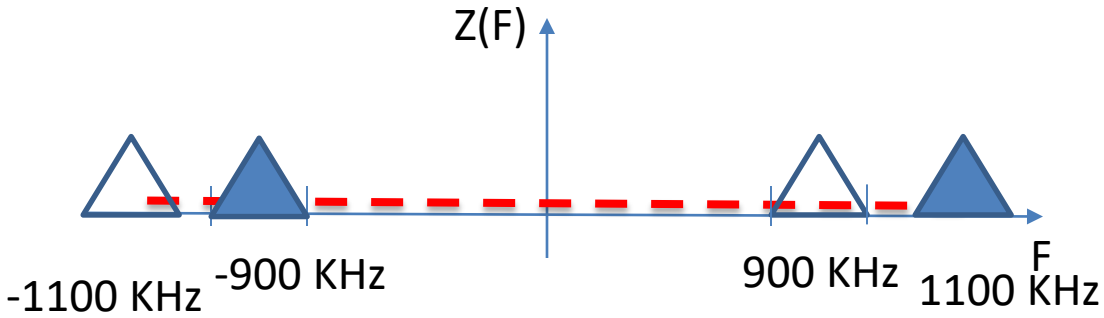


- Spectral domain contains two copies of X(F) with noise everywhere



Effect of sampling

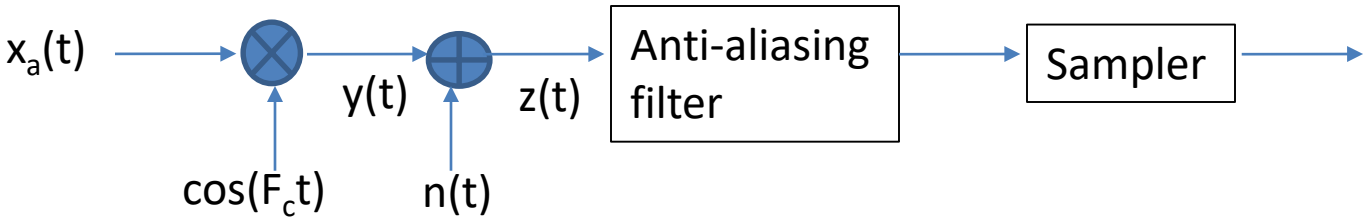
- Sampling at 2M samples/sec means spectrum gets copied left/right by 2M



- But noise also gets copied → more noise than before

How to reduce noise?

- Add *anti-aliasing filter* before sampling
- Sample at 2M samples/sec → filter passes [-1 MHz,+1 MHz]



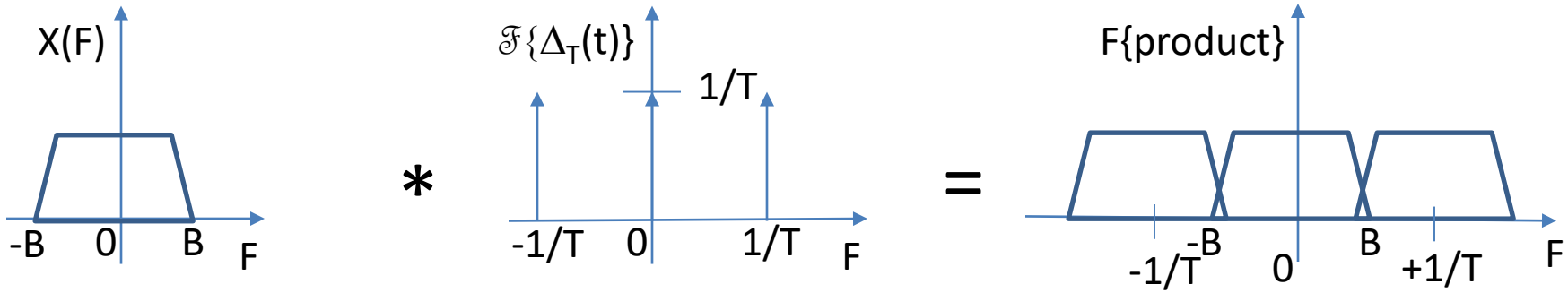
- Before anti-aliasing filter: broadband noise
  - Sampling would create (aliased) copies of noise

- After filter: noise limited to [-1 MHz,+1 MHz]
  - Sampling doesn't create copies

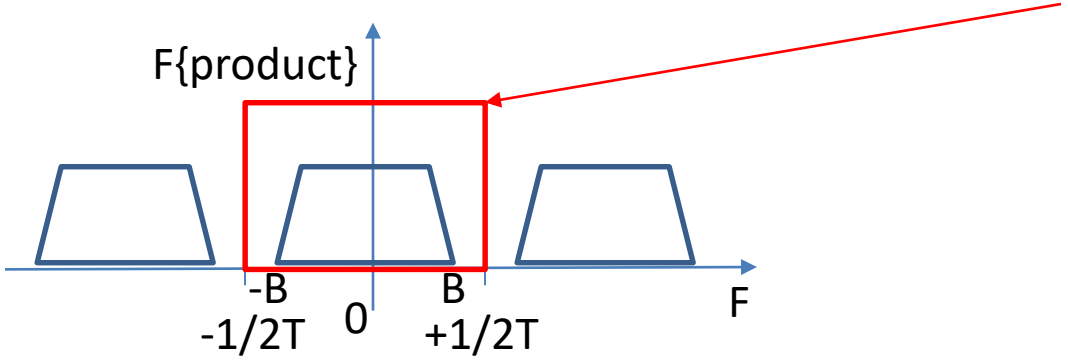
Reconstruction

Throwing away shifted copies

- Recall how sampling copies around shifted versions of X(F)



- Sampling above Nyquist means X(F) lies *inside* range (-1/2T,+1/2T)



- Obtain X(F) by applying low pass filter (LPF) to capture *red box*

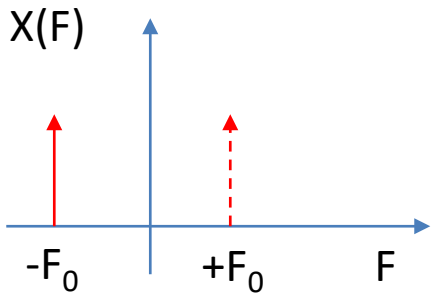
What's this LPF?

- Filtering sampled product by LPF = convolution by sinc
- $x_a(t) = \{x_a(t)\Delta_T(t)\} * sinc\left(\frac{\pi}{T}(t - nT)\right)$

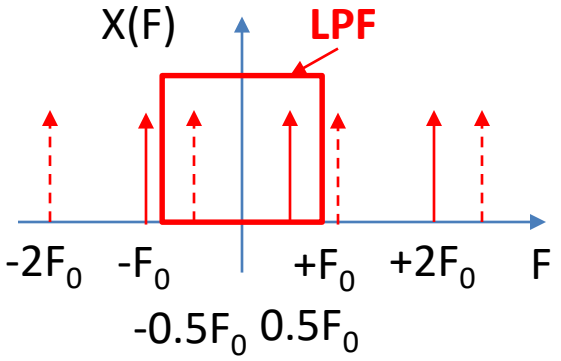
  $= \sum_{n=-\infty}^{+\infty} x_a(nT)\, sinc\left(\frac{\pi}{T}(t - nT)\right)$

Example 6.1.1

- $x_a(t)=\cos(2\pi F_0 t)=0.5\{e^{+j2\pi F0t}+e^{-j2\pi F0t}\}$



- Let's sample at $F_s<2F_0$ → aliasing
- Concrete value: $F_s=1.5F_0$
- Deltas get shifted right/left by $1.5F_0$



- Suppose we try to reconstruct
  - LPF for range $(-0.75F_0,+0.75F_0)$ where $0.5F_s=0.75F_0$
  - There are erroneous deltas (at $\pm 0.5F_0$) within this range
  - The "real" deltas ($\pm F_0$) aren't there

Recap

- Sampling = multiply $x_a(t)$ by impulse train $\Delta_T(t)$
  - Fourier of product is convolution between X(F) and $F\{\Delta_T(t)\}$
  - Because $F\{\Delta_T(t)\}$ is also impulse train, X(F) copied around

- Aliasing - overlapping spectral copies
  - Sample fast enough → no overlap
  - Sample too slow → impossible to identify X(F) within product signal

- Reconstruction - X(F) appears in lower frequencies
  - Pick off X(F) using LPF

Perfect reconstruction when sampling above Nyquist!!

Back to the Book (Details)

Fourier transform of impulse train

- Impulse train $\Delta_T(t)$ is T-periodic → can express as Fourier series

  $\Delta_T(t) = \sum_{k=-\infty}^{+\infty} C_k e^{j2\pi k\frac{t}{T}}$
- Let's compute C_k
- $C_k = \frac{1}{T}\int_{t=-T/2}^{+T/2} \Delta_T(t) e^{-j2\pi k\frac{t}{T}} dt = \frac{1}{T}\int_{t=-T/2}^{+T/2} \delta(t) e^{-j2\pi k\frac{t}{T}} dt$

  $= \frac{1}{T} e^{-j2\pi k\frac{0}{T}} = \frac{1}{T}$
- Note: within [-T/2,+T/2] interval there's one delta, which picks off value of exponent at t=0

- Fourier *series* C_k can be represented as Fourier *transform* X(F) with deltas → $\mathcal{F}\{\Delta_T(t)\} = \frac{1}{T}\sum_{k=-\infty}^{+\infty} \delta\left(F - \frac{k}{T}\right)$

Example 6.1.2

- $x_a(t)=e^{-A|t|}$
- Saw in Question 4.2 that $X(F)=2A/[A^2+(2\pi F)^2]$
  - Resembles 2019 Midterm2 question 2
- Signal not band-limited
  - Undefined derivative at t=0 creates slowly decaying freq response
- Aliasing no matter how fast we sample
- However… X(F) decays quickly enough that aliasing is minor

End-to-End System Design

[Reading material: Sections 6.2-6.3]

Rationale

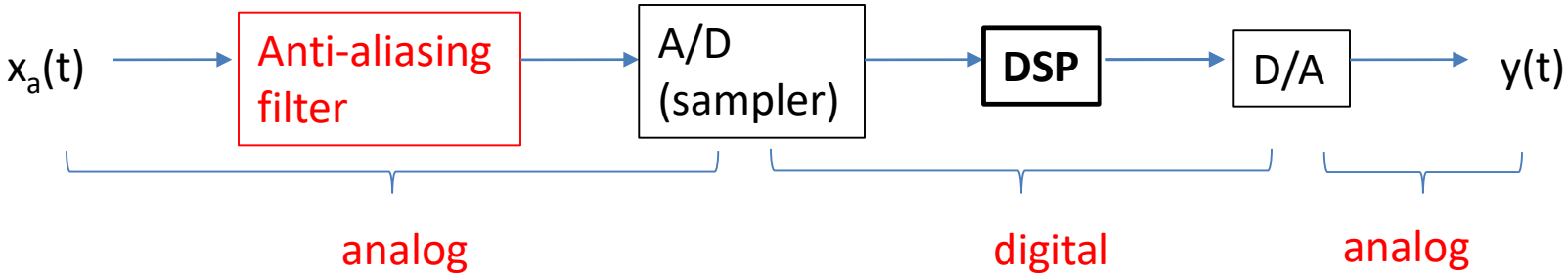
- Have seen how sampling / aliasing / reconstruction can be understood via $\mathcal{F}\{xa(t)\Delta_T(t)\}$

- Will now see how digital signal processing (DSP) can replace analog parts of system

- Recall that analog parts have various disadvantages
  - Imprecise, noise, non-linearities, costs, …
- Ideally want to use more digital components

Anti-aliasing filter

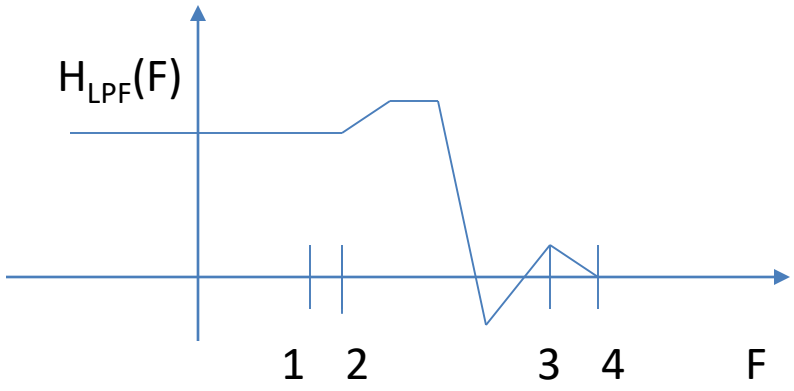
- Signal processing system using DSP:



- Anti-aliasing filter
  - Sampling introduces copies of X(F) at spacings F_s
  - No problem for truly bandlimited X(F)
  - No signal truly bandlimited → aliasing → reduce with filter

How to design anti-aliasing filter?

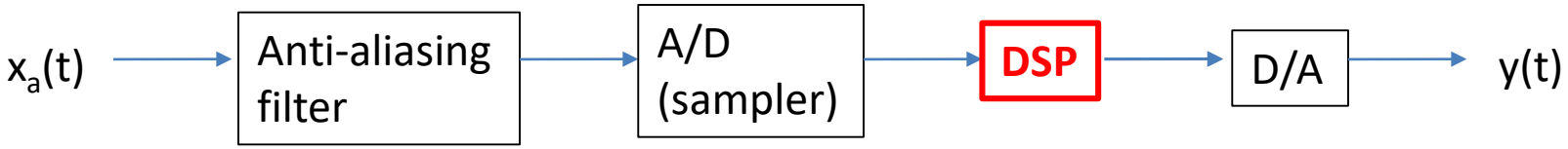
- Analog LPF has various imperfections



- Practical design introduces various guard bands:
  - #1: bandwidth of desired signal
  - #1 to #2: *guard band*
  - #2 to #3: transition from pass-band to stop band
  - #3 to #4: guard band
  - #4: beginning of stop band
- All these require running A/D well above Nyquist

Digital signal processing

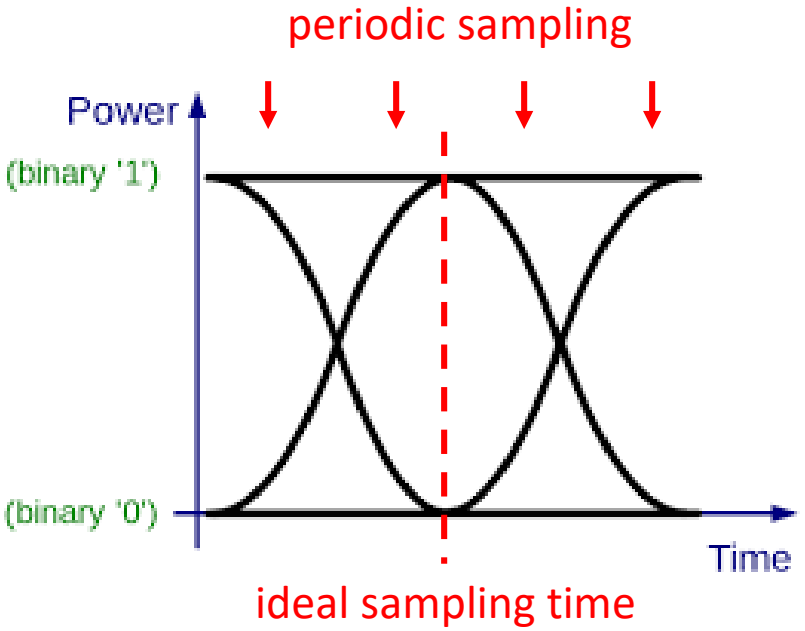
- Recall end-to-end system



- Suppose we want $\tilde{y}_a(t) = \int_{t=-\infty}^{+\infty} h_a(\tau) x_a(t-\tau) d\tau$
- Corresponds to $\tilde{Y}(F) = Ha(F) X_a(F)$
- For band-limited x_a will focus on $H(F) = \begin{cases} H_a(F), & |F| \leq B \\ 0, & else \end{cases}$
- Cascade of anti-aliasing, A/D, digital filtering with H(F), D/A yields desired output

Example 6.2.3 (& personal story)

- In some applications want to delay signal
- Communications application
  - Symbols being transmitted
  - Symbol = waveform
  - Noise → sample at right time
- A/D samples at "wrong" time



- Solution: estimate non-integer delay, then apply "delay filter"

Example 6.2.3 continued

- How does “delay filter” work?

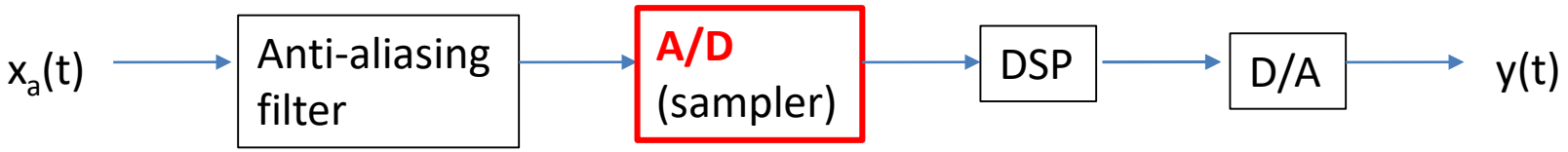
- Delay by integer # samples is easy (store in registers)

- Delay by non-integer Δ samples involves $H(\omega)=e^{-j\omega\Delta}$
  - Need filter structure $h(\Delta)$ - depends on Δ

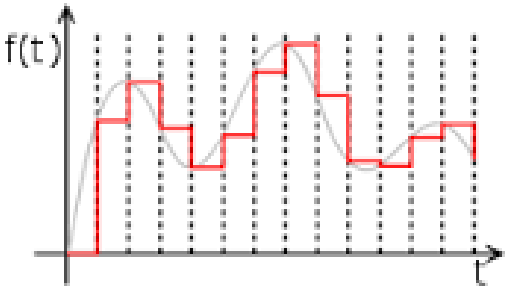
A/D and D/A Conversion

A/D conversion

- Have discussed entire signal processing system
- Will now add details for A/D, quantization, D/A

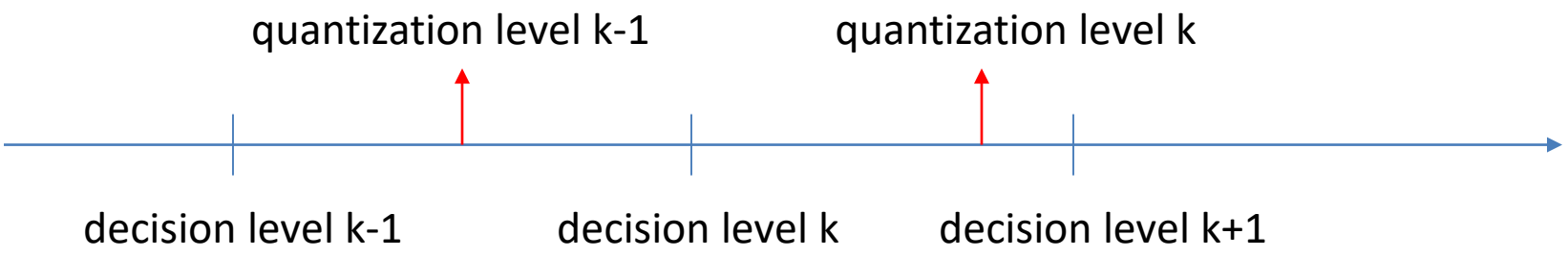


- A/D contains two parts
  - *Sample & hold* – analog circuit freezes signal
  - Actual A/D conversion once signal is frozen



Quantization

- Digital processing in *bits*
- Have finite # bits → finite # possible levels for signal
- Must map continuous valued signal (infinite possible levels) to finite # levels



- Partition signal levels into bins B_k={x(n)∈(level k, level k+1]}
- x(n)∈B_k → assign quantization level k to x(n)

Structure of bins

- *Quantizers* often have uniform step size
  - Level k+1 = Level k + Δ
  - First/last bins extend to $\pm\infty$

- Advantage – uniform quantizers simple to design

- Disadvantage – average error often sub-optimal (same # levels)

Quantization error

- Quantization level often in middle of bin
- Error obeys $e_q(n)=x(n)-x_q(n) \in (-\Delta/2,+\Delta/2]$

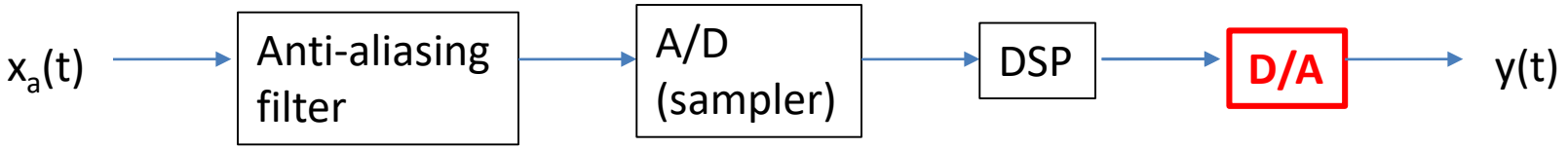
- Can show (see book) mean squared error $\approx \Delta^2/12$
  - Assumes noise distribution uniform in $(-\Delta/2,+\Delta/2]$

- More bins → smaller Δ → smaller error
- But more bins also requires more bits
  - You're also seeing this in image compression project

- *Trade-off between bits and error*

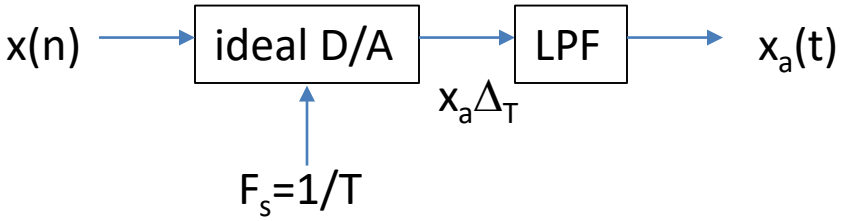
D/A conversion

- How do we convert back to analog?



- Ideal solution:

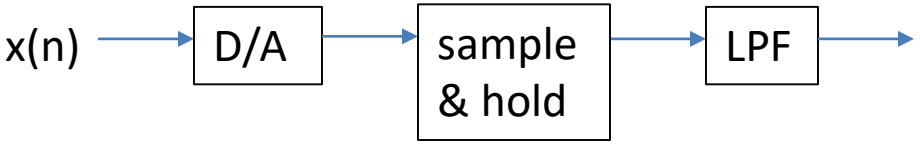
1) Modulate impulse train w/digital levels
2) Apply LPF



- Impossible/impractical to put x(n) on analog deltas

D/A using sample & hold

- Use non-ideal D/A
- Add sample & hold to stabilize D/A output

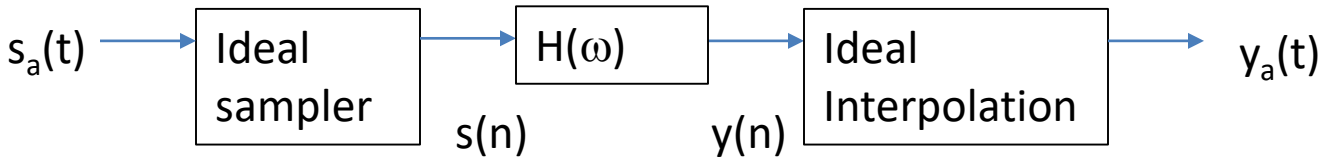


- However… sample & hold in time domain → convolve $x_a \Delta_T$ by square pulse → multiply by sinc in freq domain

- Can compensate in different ways
  - Ex: Interpolate between samples & run D/A at higher sampling rate
- No perfect solution, just changes trade-off

Problem 6.13

- Consider continuous time input $x_a(t)$ with bandwidth B
- Input and echo are received together, $s_a(t)=x_a(t)+\alpha x_a(t-\tau)$
- Applications:
  - Communication – signal traverses multiple paths → delays/echos
  - Audio – same idea (phones use echo cancelation)



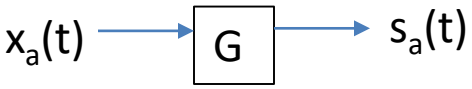
- <u>Challenge:</u> specify F_s and $H(\omega)$ such that $y_a(t)=x_a(t)$?

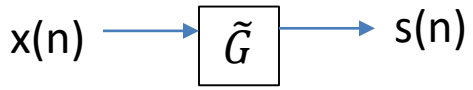
Problem 6.13 part 2

- What are the ideal components?
  - Ideal sampler = multiply by impulse train
  - Ideal interpolation = assign y(n) to deltas & apply LPF
- Observation: Fourier(echo)=X(F)·exponential
- Implication: bandwidth(echo)=bandwidth(input)=B
  - $F_s=2B \rightarrow$ s(n) contains all information in $s_a(t)$

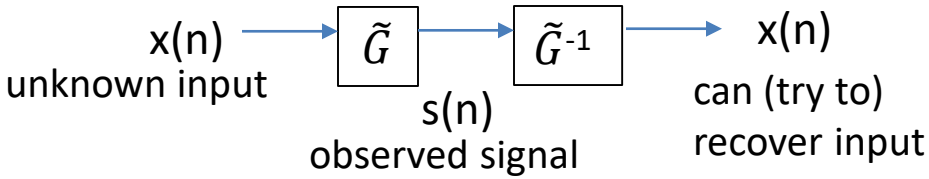
Problem 6.13 part 3

- Can think of $s_a(t)$ and $s_a(n)$ as follows:





- s(n) contains all information about $s_a(t)$ → apply inverse filter



This is equivalent to deconvolution

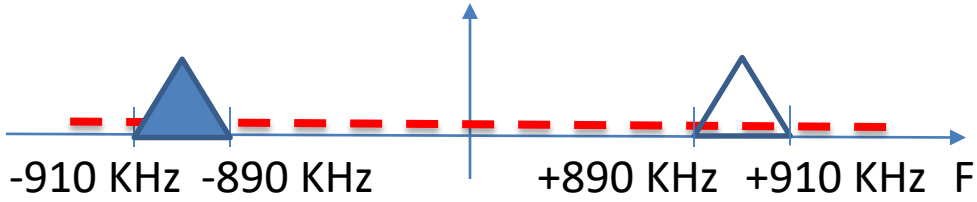
– We've discussed how it's quite complicated

Modern Signal Processing

Sampling Bandpass Signals

Bandpass signals

- Recall our old friend the AM modulated signal



- Spectral occupancy = 910 – 890 = 20 KHz << 910 KHz
  - Quite narrow-band
  - Sampling at Nyquist (2*910=1.82M samples/sec) inefficient

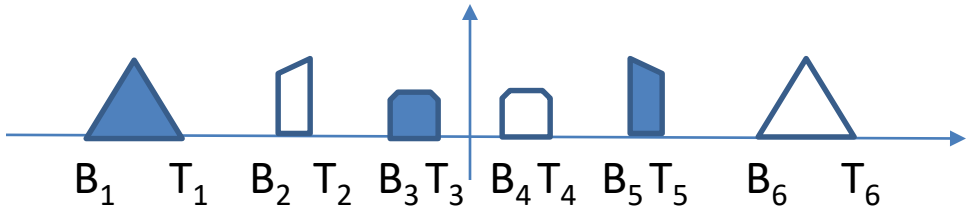
- Conventional approach in radio systems:
  - Multiply by sinusoidal signal at carrier freq (900 KHz)
  - New signal around DC and around 1.8 MHz
  - Apply LPF to remove copies at $\pm$1.8 MHz
- *New baseband signal can be sampled at 20K samples/sec*

Challenges

- What if our 900 KHz isn't precise?
- Copies of original signal moved from $\pm$900 KHz to baseband won't be centered around zero
- Can be done in principle…

Landau rate

- Let's go from one pass band to many



- Each spectral band begins at bottom B_i ends at top T_i

- Landau's sampling theorem provides perfect reconstruction for rates above total spectral occupancy $\Sigma_i(T_i-B_i)$
  - Significant expansion over traditional sampling theorem
  - Complicated to implement
  - Must know location of bands, B_i and T_i

- Research from 90's shows that locations need not be known

Transforms

Linear algebra 101

- Column vector $v = \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix}$
- Matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, more generally M rows, N columns
  - One row/column implies row/column vector

- Matrix vector product (examples):
  - $\begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \cdot 1 + 2 \cdot 2 \\ 2 \cdot 1 + 3 \cdot 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \end{bmatrix}$
  - $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} = 1 \cdot 4 + 2 \cdot 5 + 3 \cdot 6 = 32$

Matrix inverse

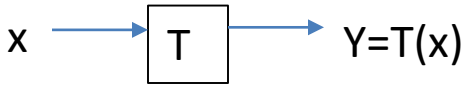
- For matrix A such that y=Ax, inverse A^{-1} satisfies $A^{-1}y=x$
- Can show $AA^{-1}=A^{-1}A=I$
  - I identity matrix (ones on diagonal, else zero)
- Matrix (usually) invertible if it's square

- Example:
  - $A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$
  - $A^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1/2 \end{bmatrix}$
  - $AA^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1/2 \end{bmatrix} = \begin{bmatrix} 1\cdot 1+0\cdot 0 & 1\cdot 0+0\cdot 1/2 \\ 0\cdot 1+2\cdot 0 & 0\cdot 0+2\cdot 1/2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

identity matrix

Transform

- Transforms are (typically) linear operators applied to input vectors



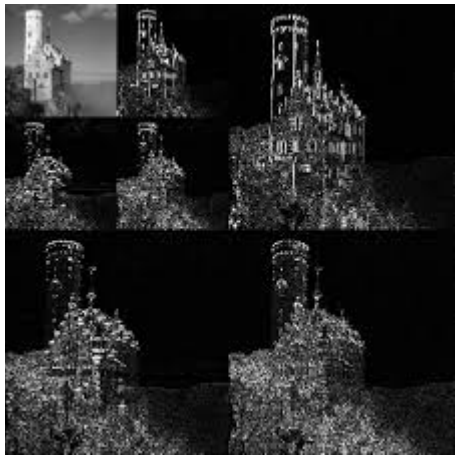
- Input $x \in \Re^N$ (N real valued numbers), output $y \in \Re^N$
- Y is linear combination of x, $y_i = \Sigma_j T_{ij} x_j$

- Example
  - Fourier transform of periodic signal in discrete time (finite length N)

- Weights T_{ij} chosen carefully
  - Want T to be invertible, $x = T^{-1} y$
  - Parseval holds (energy conservation)

Sparsity

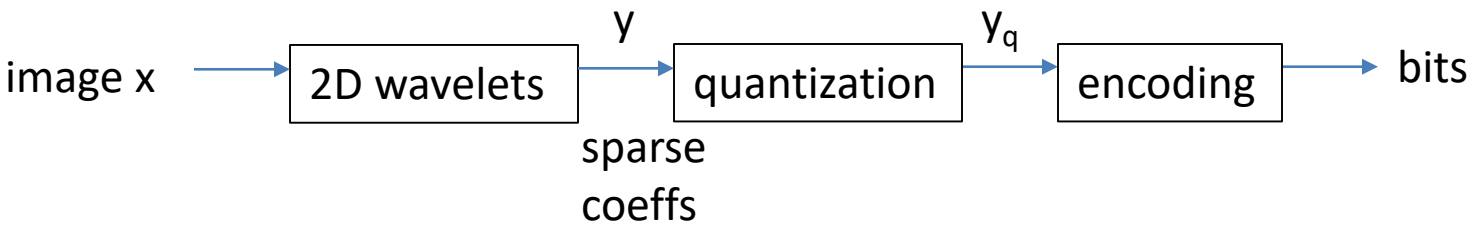
- Not all transforms are useful
- Transform useful if energy concentrated in few coefficients
  - This is called *sparsity*
  - Different classes of signals sparsified by different transforms



- Example
  - Fourier – sparsifies bandpass signals
  - Wavelets – for “smooth” signals w/few discontinuities
  - 2D wavelets (images) → most energy in 1-5% of coeffs

Image compression application

- Let's apply 2D wavelets to image compression



- Output y of transform is very sparse
- Quantize it with stepsize that takes most coeffs to zero
- Quantization error $e_i=y_i-y_{q,i}$
- Total energy of error $\varepsilon=\Sigma_i(y_i-y_{q,i})^2$
- Error in reconstructed version x', $\Sigma_i(x_i-x'_i)^2=\varepsilon$

- y_q mostly zero → can compress with few bits

Compressed Sensing

[R. Baraniuk, “A Lecture in Compressive Sensing.” Signal Proc. Mag. 2007]

Remaining challenges

- Transform coding greatly reduce coding rate required to describe signal → reduces bandwidth

- Still need to sample signal
  - Battery operated applications – lots of measurements will drain battery
  - High bandwidth applications (radar) – required sampling rate pushes limits of A/D's

- David Donoho: "Why go to so much effort to acquire all the data when most of what we get will be thrown away?"

What does compressed sensing do?

- New framework for sampling, processing, & reconstructing sparse signals

- Input $x \in \Re^N$
- Only K<<N nonzeros (sparsity)
  - Could be sparse with respect to sparsifying transform T
  - Could have K large coeffs, rest small (still nonzero)

- *Random measurements* $y_m = \sum_{n=1}^{N} \Phi_{mn} x_n$
  - Only M<<N measurements
  - Elements of Φ_{mn} generated randomly
  - M typically somewhat bigger than K

Under determined linear inverse problem

- Compressed problem can be written in linear algebra form

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \Phi_{13} \\ \Phi_{21} & \Phi_{22} & \Phi_{23} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

- Matrix contains fewer rows than columns → under determined

- General case:
  - No knowledge about x
  - Infinitely many solutions for x → impossible to determine x

- Compressed sensing:
  - x sparse → can reconstruct x from few measurements

Example

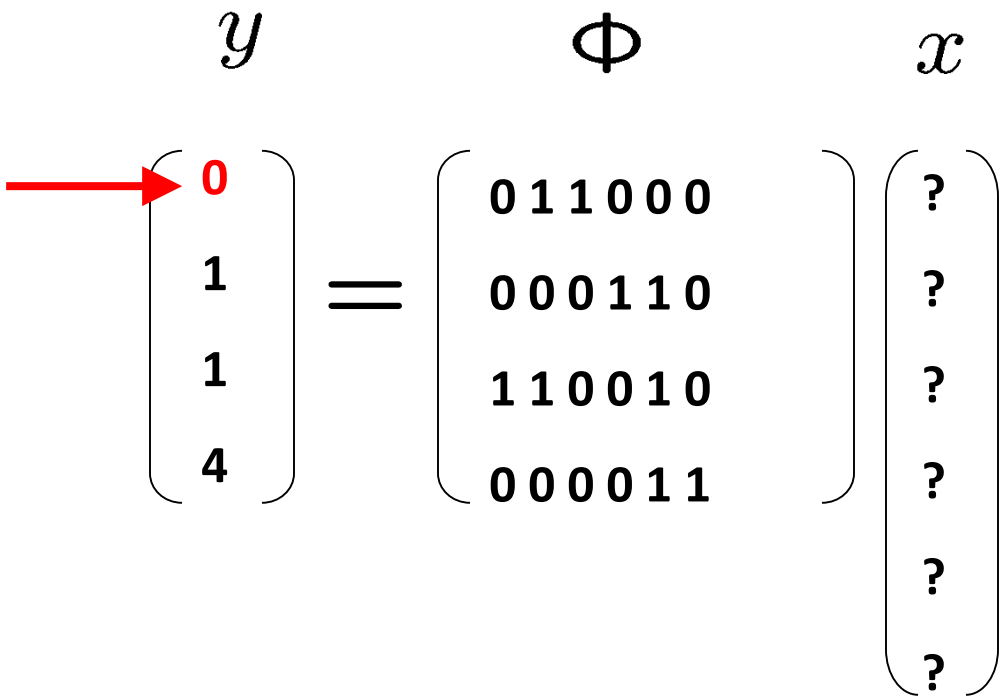
[Sarvotham, B, & Baraniuk, “Sudocodes - Fast Measurement and Reconstruction of Sparse Signals,” Int. Symp. Info. Theory, 2006]

Example

$$
\begin{array}{ccc}
y & \Phi & x \\
\begin{pmatrix} 0 \\ 1 \\ 1 \\ 4 \end{pmatrix} & = & \begin{pmatrix} 0\,1\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0 \\ 1\,1\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1\,1 \end{pmatrix} \begin{pmatrix} ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{pmatrix}
\end{array}
$$

Example

- What does zero measurement imply?
- Hint: x strictly sparse



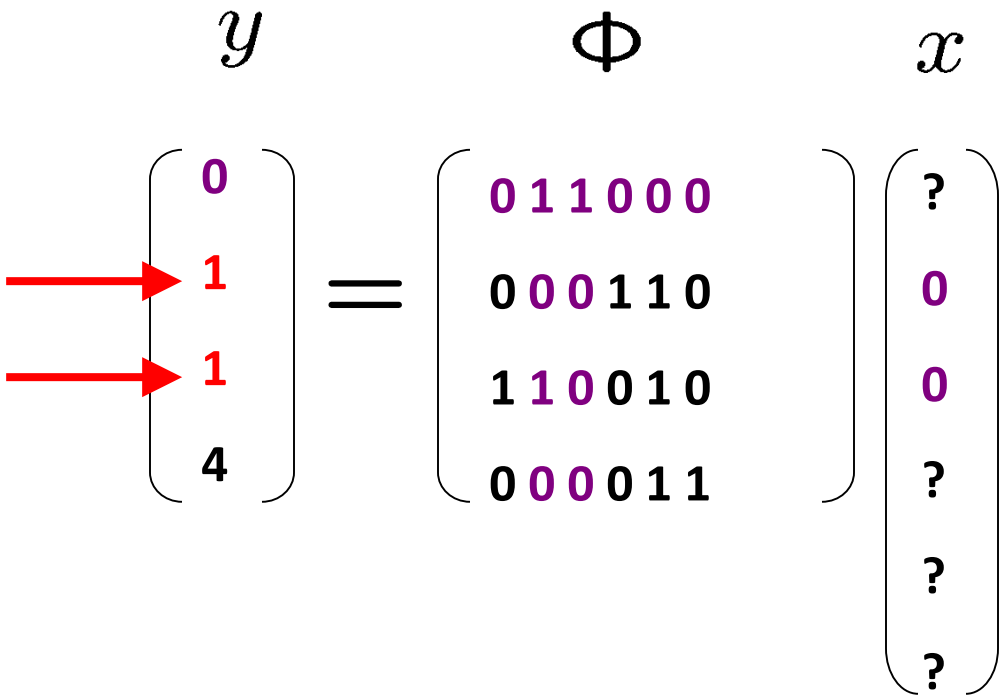
Example

- Graph reduction!

$$
\begin{array}{ccc}
y & \Phi & x \\
\begin{pmatrix} 0 \\ 1 \\ 1 \\ 4 \end{pmatrix} & = & \begin{pmatrix} 0\,1\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0 \\ 1\,1\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1\,1 \end{pmatrix} \begin{pmatrix} ? \\ 0 \\ 0 \\ ? \\ ? \\ ? \end{pmatrix}
\end{array}
$$

Example

- What do matching measurements imply?
- Hint: non-zeros in x are real numbers



Example

- What is the last entry of x?

$$y \qquad \Phi \qquad x$$

$$\begin{pmatrix} 0 \\ 1 \\ 1 \\ 4 \end{pmatrix} = \begin{pmatrix} 0\,1\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0 \\ 1\,1\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1\,1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ ? \end{pmatrix}$$

Discussion

- Compressed sensing can solve under determined linear inverse problem $y=\Phi x$

- Compressive signal processing
  - Acquire random linear measurements (in hardware)
  - Use optimization algorithms to search for sparse x that satisfies measurements

- Described toy problem; can be greatly extended
  - Measurement noise, $y=\Phi x+z$
  - Input x not sparse but *structured*
  - Fast reconstruction algorithms
  - Acquisition and processing multiple correlated signals

Denoising

[Retired project]

Where does denoising appear?

- Setting: $y(t)=x_a(t)+n(t)$
  - Noise n(t) could be Gaussian "bell curve" noise (thermal noise)
  - Speckle effects → saturation → sensor returns min/max values
    - Called "salt & pepper" noise

- Audio processing, e.g., old recording is scratched
  - Large amplitude noise resembling salt & pepper

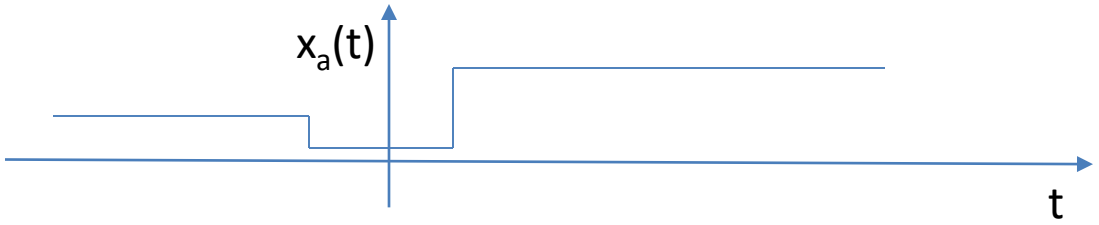
- Image processing (project)
  - From 1D signal to 2D image

- Modern example:
  - In iterative estimation algorithms, n(t) is estimation error
  - Error n(t) becomes smaller over iterations

How can we denoise?

- Need to start somewhere – exploit *structure* in input $x_a(t)$

- Types of structure?
  - *Band limited* signals – signal occupies few freqs w/large coeffs; broadband noise w/small coeffs
  - Knowledge about how much energy (on average!) signal/noise have in different spectral bands
    - More signal → barely attenuate
    - More noise → attenuate a lot
  - Beyond Fourier coefficients and frequency…
    - *Piecewise constant* signal $x_a(t)$ jumps at times $< T_{-1} < T_0 < T_1 < \ldots$



Median filters

- Moving average filters are low pass; can be used to denoise Gaussian noise

- Noise with outliers (e.g., salt & pepper noise) does not work well with standard low pass

- Non-linear median filters better for outliers

n	0	1	2	3	4	5	6
x(n)	12	13	11	105	10	14	12
Mean	…	12	43	42	43	12	…
Median	…	12	13	11	14	12	…

What about noise?

- Noise (e.g., static) typically added to signal
- Time domain – signal/noise might be comparable in magnitude
- Frequency domain perspective
  - Energy of signal concentrated into prominent spikes
  - Noise energy modest as before
  - Signal (at spikes) much larger than noise