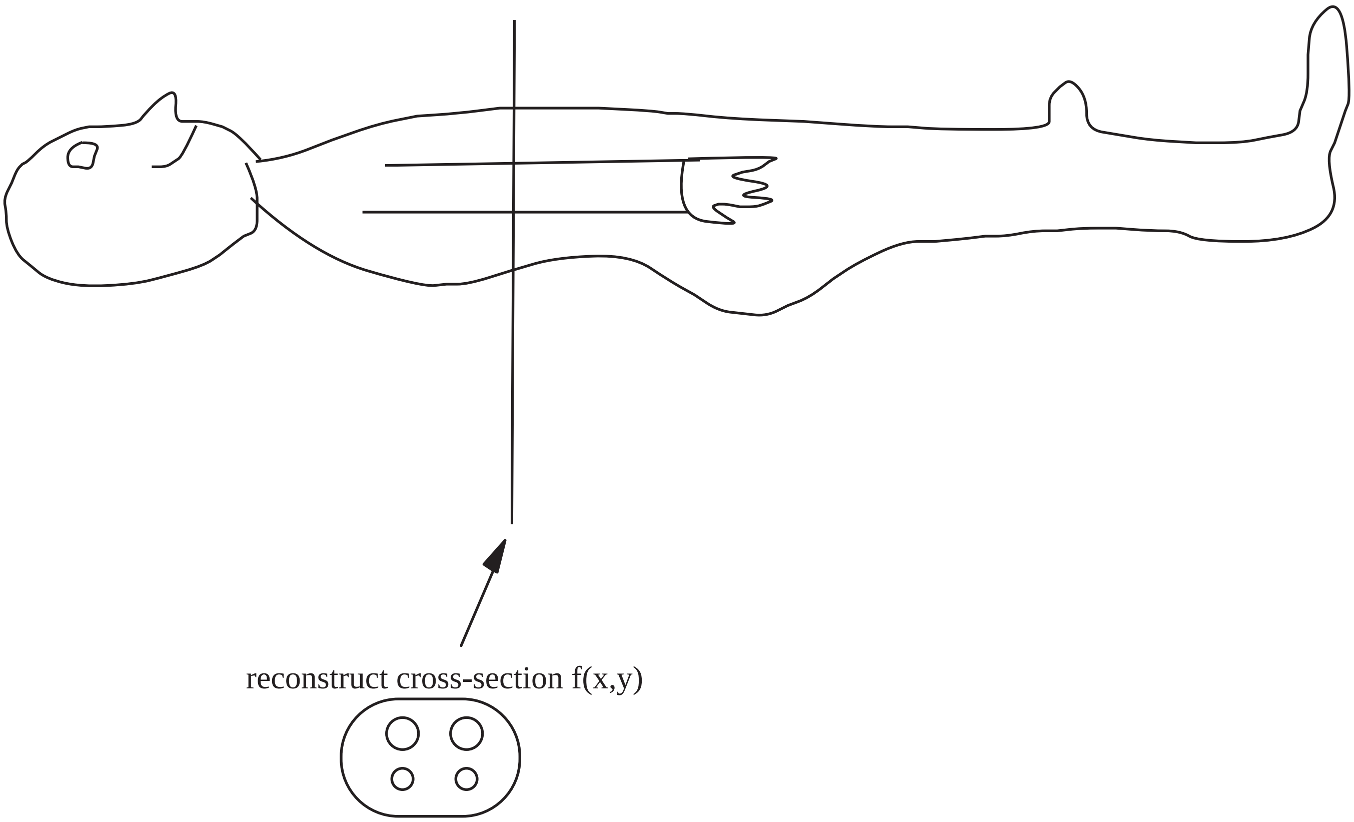
ECE 410 **DIGITAL SIGNAL PROCESSING** **D. Munson**
University of Illinois **Chapter 15**

Application 1: Computer Tomography (CT)

Used extensively for medical imaging, nondestructive testing.

Objective is to reconstruct a cross-sectional view of a 3-D object:

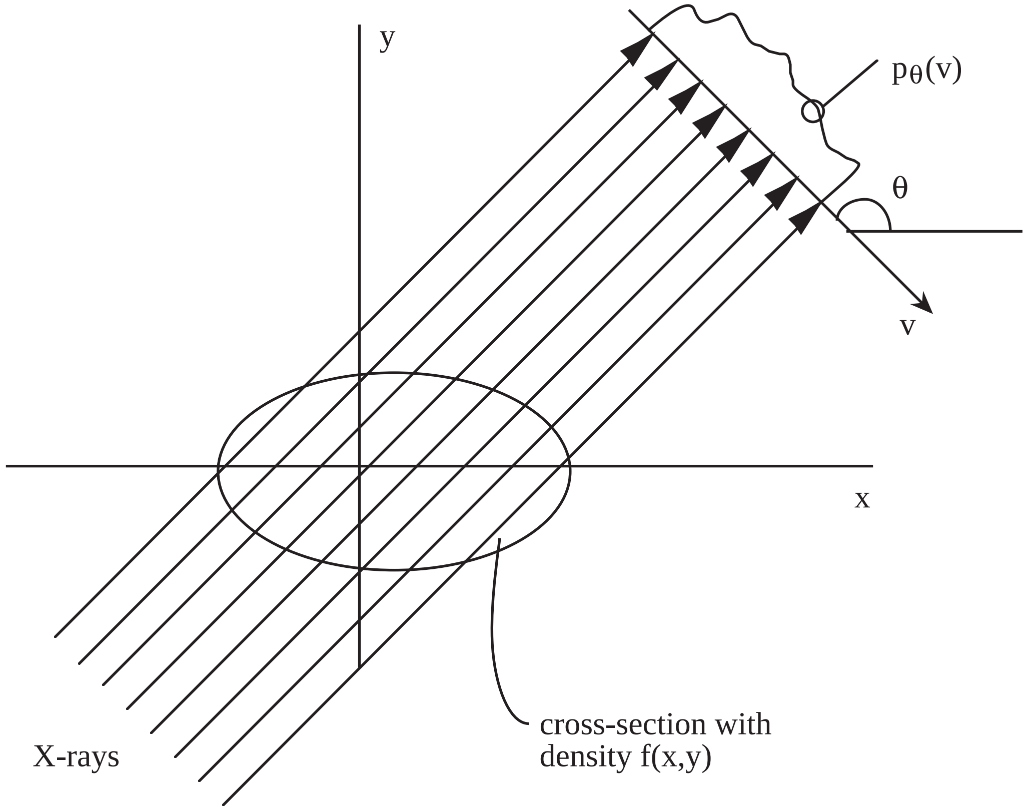


Accomplish this by shining x-rays sideways through the object and collecting “projections” at various angles. The projection data is then processed digitally to produce the image of f(x,y).

The oldest CT machines used narrow, parallel x-ray beams. Modern-day machines use a fan-beam geometry. Since the digital processing in both systems is similar, we will consider the parallel-beam case, which is a bit simpler mathematically.

15.2

Parallel beam geometry



$p_\theta(v)$ is a set of line integrals called a **projection.**

$p_\theta(v_o)$ is the integral of f(x,y) along the path of the x-ray at angle θ impinging at $v = v_o$.

Typically, projections $p_\theta(v)$ are collected through a full 360˚ by rotating the x-ray source(s) and detectors around the object being imaged.

How do we recover f(x,y) from the projections $p_\theta(v)$?

Define the 2-D Fourier transform of f(x,y) as

$$F(\Omega_1, \Omega_2) = \int_{-\infty}^{\infty}\!\!\int f(x,y)e^{-j(\Omega_1 x+\Omega_2 y)}\, dxdy$$

The inverse 2-D Fourier transform is

$$f(x,y) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty}\!\!\int F(\Omega_1,\Omega_2)\, e^{j(\Omega_1 x+\Omega_2 y)}\, d\Omega_1\, d\Omega_2$$

Notation:

Let $F(\Omega_1,\Omega_2)$ in <u>polar</u> coordinates be written as

$$F_{pol}(r, \phi) = F(r\cos\phi, r\sin\phi)$$

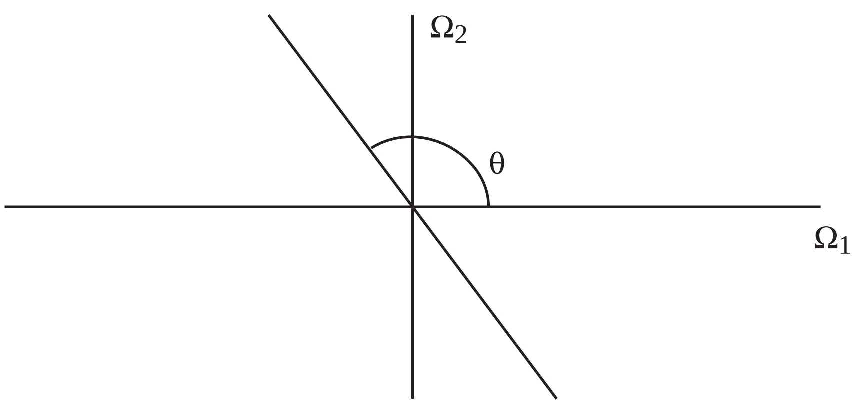
Then the following famous theorem forms the basis for reconstructing $f(x,y)$ from its projections.

Projection-Slice Theorem

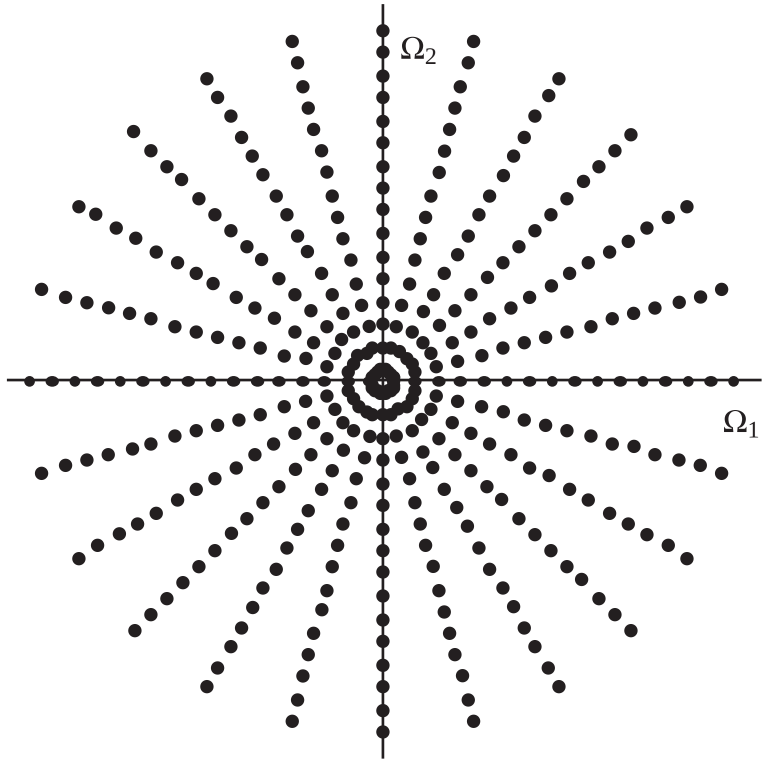
Let $P_\theta(\Omega)$ be the 1-D Fourier transform of $p_\theta(v)$. Then

$$\left[P_\theta(\Omega) = F_{pol}(\Omega,\theta)\right]$$

So, the Fourier transform of a projection is a radial slice of the 2-D Fourier transform of $f(x,y)$ at angle θ:



Collecting sampled projections at many (usually hundreds) of angles and taking the DFT (via FFT) of each projection gives samples of $F(\Omega_1, \Omega_2)$ on a polar grid:



To reconstruct samples of f(x,y) we might try discretization of the inverse 2-D Fourier transform.

We had:

$$f(x,y) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty}\!\!\int F(\Omega_1,\Omega_2)\, e^{j(\Omega_1 x+\Omega_2 y)}\, d\Omega_1\, d\Omega_2$$

Writing this integral in polar coordinates, and then discretizing, would give an approximate formula for f(nT,mT) in terms of the available polar samples of $F(\Omega_1, \Omega_2)$. Computing N^2 samples of f(x,y) from N^2 samples of F would require

$$N^2 \times N^2 = N^4 \otimes$$

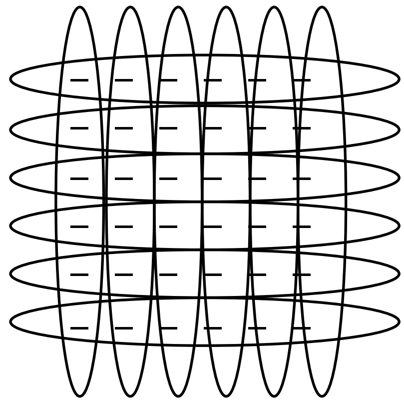
which is excessive. For example, if N = 512, this approach would require about 64×10^9 MAs.

A faster alternative would be to

1) Interpolate the polar Fourier data to a Cartesian grid.

2) Compute a 2-D FFT^{-1} (requires $\sim 2N^2 \log_2 N$ MAs)

A 2-D DFT is implemented by a series of row FFTs, followed by a series of column FFTs:



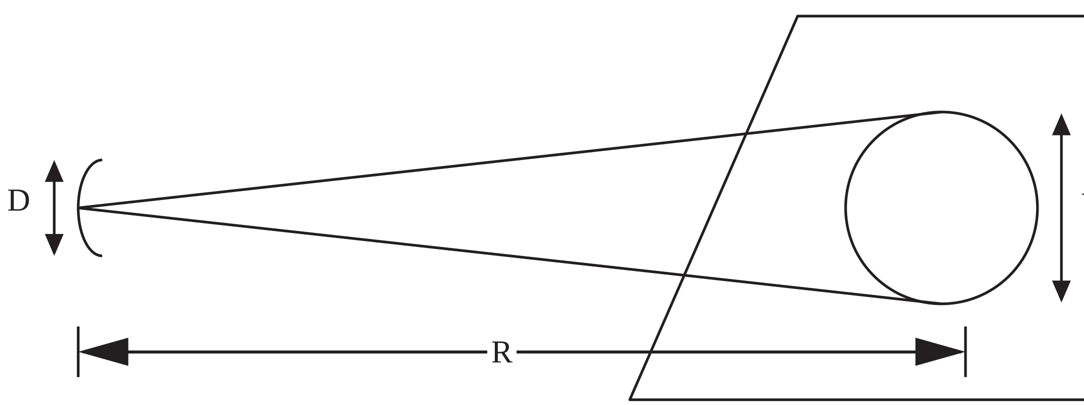
In practice, an accurate implementation of Step 1) requires more computation than the 2-D FFT^{-1}. The most popular image reconstruction algorithm for computer tomography is "convolution-back-projection" (also called filtered back-projection) which is essentially an accurate and efficient way to accomplish 1) and 2) in $0(N^3)$ MAs. Researchers are now working on convolution-back-projection algorithms that require only $0(N^2 \log_2 N)$ MAs.

Application 2: <u>Synthetic Aperture Radar (SAR)</u>

SAR is a high-resolution microwave imaging system. It is used widely in applications such as earth resources monitoring, military reconnaissance, planetary imaging, etc.

Same advantages of microwave imaging over optical: can penetrate fog, cloud cover, atmosphere of Venus, etc., and does not rely on illumination by the sun.

Disadvantage: It is hard to achieve optical resolution. Why? Because, although we can get high resolution in range via delay measurements, the cross-range resolution is seemingly limited by the antenna beamwidth, which can be very wide at microwave frequencies:



If D is the antenna diameter, R is the range to the scene, and λ is the wavelength of radiation, then the width of the antenna-footprint is

$$W = \frac{R\lambda}{D}$$

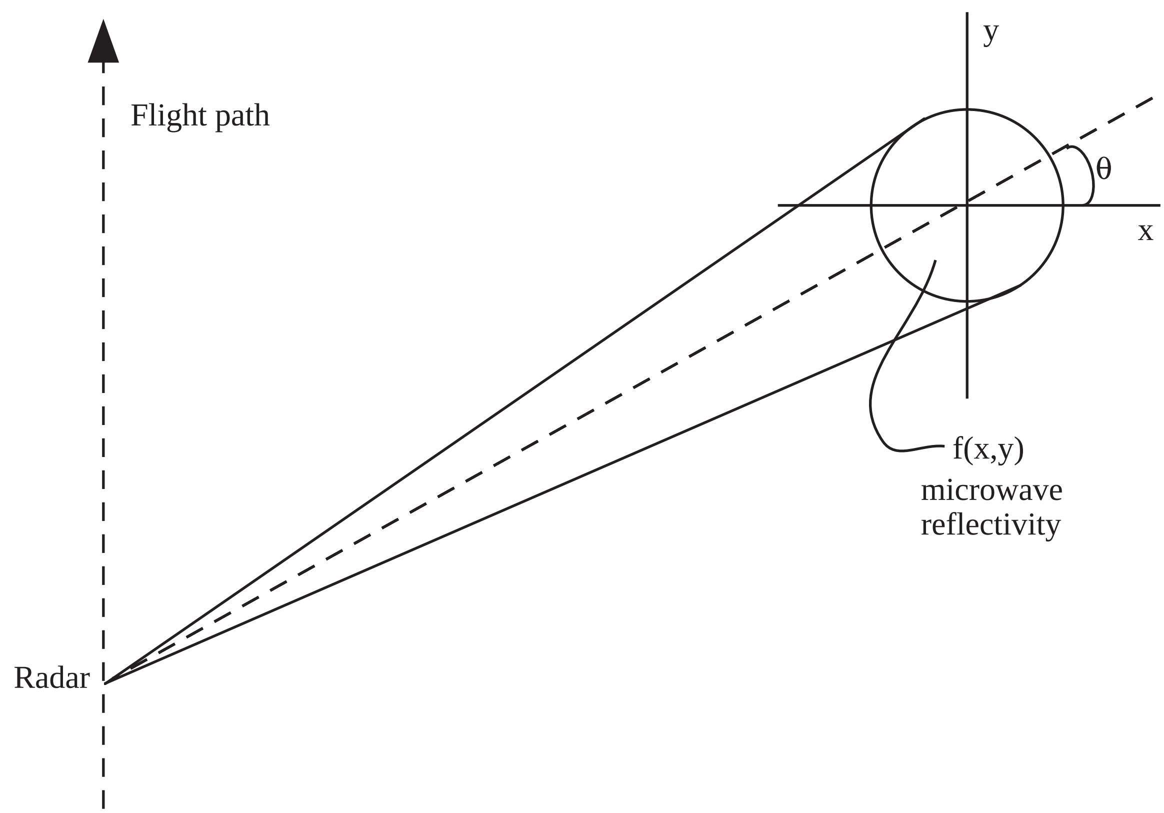
For high cross-range resolution, we want W to be small, but this is hard to get. R may be large and dictated by the imaging scenario, and you hope to use an antenna of practical size (D small).

For microwaves, λ is <u>much</u> larger than for visible light. So, with microwaves, D may need to be impractically-large to achieve a desired cross-range resolution.

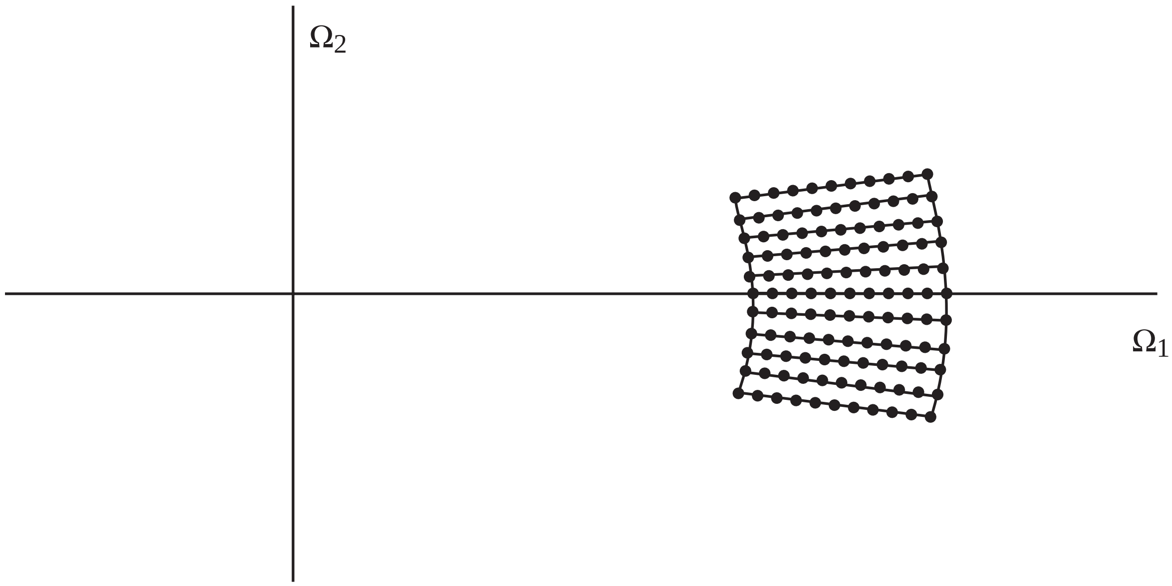
Solution:

Use small D with large W, but collect and process data from <u>many angles</u>. This is called <u>spotlight-mode SAR</u>.

Imaging geometry



If a linear FM waveform $\cos(\Omega_o t + \alpha t^2)$ is transmitted, it can be shown that demodulated, sampled returns provide Fourier data on a polar grid:



The return collected at angle θ in the spatial domain gives Fourier data on the radial trace at the same angle θ in the Fourier domain. (Proof of this fact uses the projection-slice theorem.) The inner and outer radii of the Fourier data region are proportional to the lowest and highest frequencies, respectively, of the transmitted linear FM signal.

Reconstruction algorithm:

1) Polar-to-Cartesian interpolation

2) 2-D FFT^{-1}

3) Display magnitude of the result.

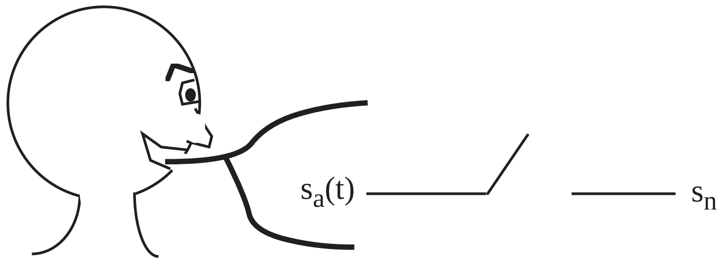
Typical resolution:

1 ft., or less, to 20 m, depending on the application

These resolutions are achievable at exceedingly long ranges (e.g., space-based monitoring of the earth).

Application 3: Speech Analysis/Synthesis

Consider an approach to speech coding called LPC – Linear Predictive Coding.
This scheme is used in many speech communication systems, automated answering systems and electronic games.



Speech samples are highly correlated, so that s_n can often be fairly-well predicted from its past values.

Suppose we wish to predict s_N from $s_{N-1}, s_{N-2}, \ldots, s_{N-K}$. Try linear prediction.

Estimate s_N by:

$$\hat{s}_N = \sum_{k=1}^{K} a_k \, s_{N-k}$$

The $\{a_k\}$ are called LPC coefficients.

Choose the $\{a_k\}$ to minimize $E\left\{(s_N - \hat{s}_N)^2\right\}$.

So, do this

$$\min_{\{a_i\}_{i=1}^{K}} E\left\{\left(s_N - \sum_{k=1}^{K} a_k \, s_{N-k}\right)^2\right\}$$

$$\Rightarrow \frac{\partial}{\partial a_i} E\left\{\left(s_N - \sum_{k=1}^{K} a_k \, s_{N-k}\right)^2\right\} = 0 \qquad i = 1, \ldots, K$$

$$\Rightarrow E\left\{2\left(s_N - \sum_{k=1}^{K} a_k \, s_{N-k}\right)(-s_{N-i})\right\} = 0 \quad i = 1, \ldots, K$$

$$\Rightarrow \sum_{k=1}^{K} a_k \, E\left\{s_{N-k}\, s_{N-i}\right\} = E\left\{s_N \, s_{N-i}\right\} \qquad i = 1, 2, \ldots K \tag{1}$$

Suppose $\{s_n\}$ is short-term "<u>wide-sense stationary</u>." Then $E\{s_m\, s_n\}$ depends only on the separation between m and n, i.e., on $|m\text{-}n|$, not on m and n individually.

In this case we can write $E\,\{s_n\, s_m\}$ as some function $R_s(n–m) = R_s(m–n)$ where R_s is called the <u>autocorrelation</u>.

Substituting R_s into (1) gives

$$\sum_{k=1}^{K} a_k \, R_s(i-k) = R_s(i) \qquad i = 1, 2, \ldots, K$$

This set of K equations can be expressed in matrix form as

$$\begin{bmatrix} R_s(0) & R_s(1) & \cdots & R_s(K-1) \\ R_s(1) & R_s(0) & \ddots & \\ \vdots & \ddots & \ddots & R_s(1) \\ R_s(K-1) & \cdots & R_s(1) & R_s(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{bmatrix} = \begin{bmatrix} R_s(1) \\ R_s(2) \\ \vdots \\ R_s(K) \end{bmatrix} \tag{2}$$

Given the $R_s(i)$, this set of equations can be solved for the optimal $\{a_k\}_{k=1}^{K}$

We might approximate $R_s(i)$ as:

$$R_s(i) = \frac{1}{L_i} \underbrace{\sum_n}_{\uparrow} s_n \, s_{n+i}$$

terms in sum = L_i

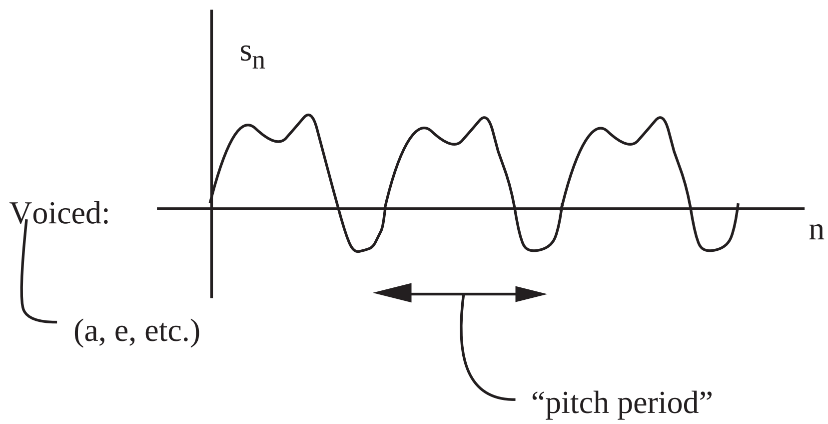
The solution of (2) would ordinarily require $0(K^3)$ operations.

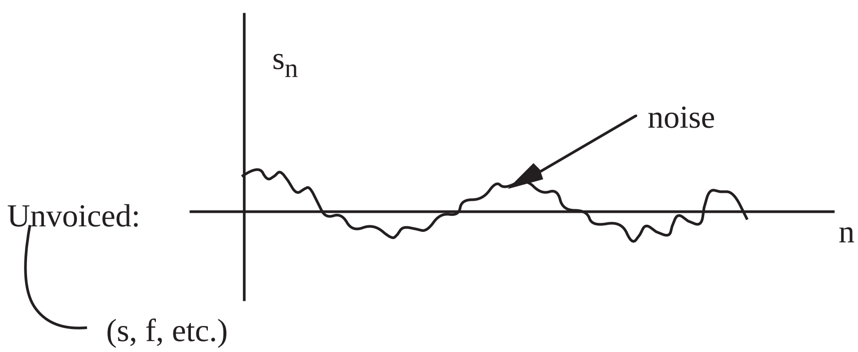
But, the matrix has a special Toeplitz structure $\Rightarrow$ faster algorithms exist.

The Levinson - Durbin algorithms require $0(K^2)$ operations.

Now, look at speech!

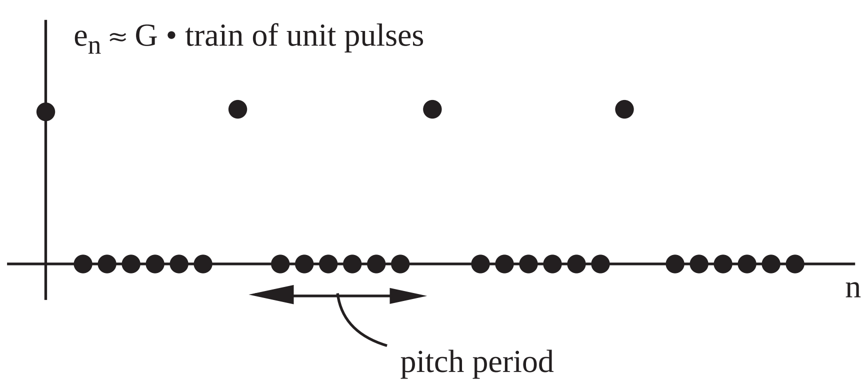
Classification of speech segments:





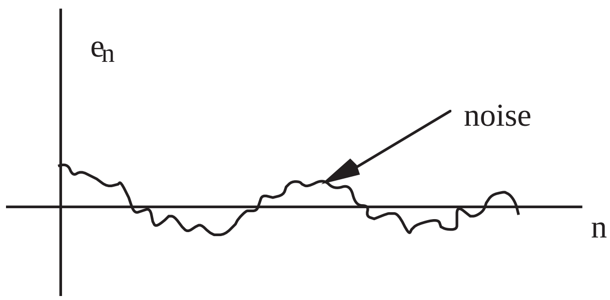
Now, suppose we have the optimal $\{a_k\}$ and we look at the prediction error $e_n = s_n - \hat{s}_n$.

It turns out that for voiced sounds e_n is well approximated by a pulse train:



where G is a slowly varying gain.

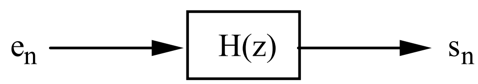
For unvoiced sounds, e_n looks like noise:



Using our definition of e_n and the LPC model, we have

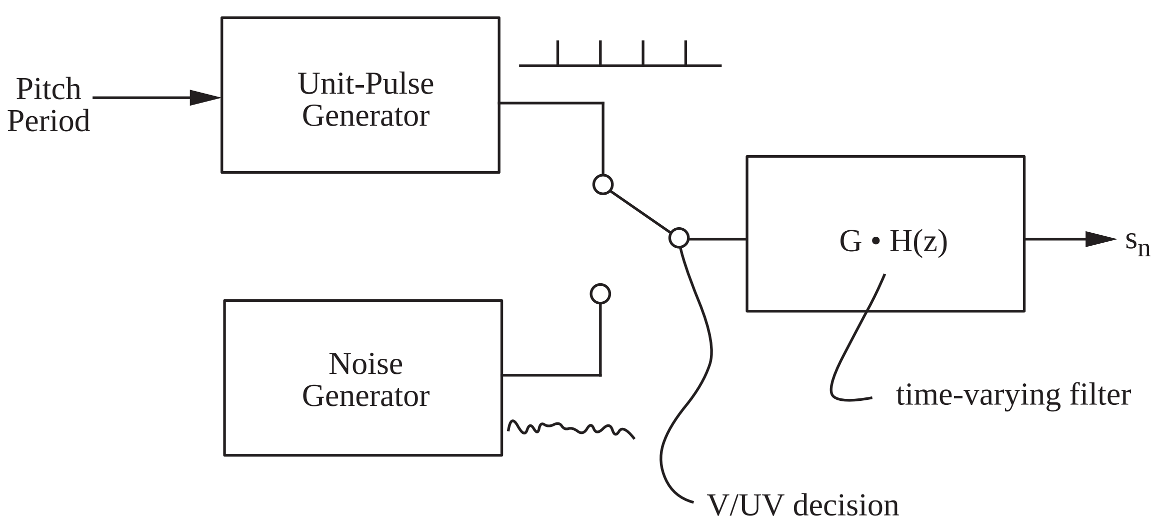
$$s_n = \hat{s}_n + e_n = \sum_{k=1}^{K} a_k s_{n-k} + e_n$$

Thus, we can obtain s_n from e_n via



where $H(z) = \dfrac{1}{1 - \sum_{k=1}^{K} a_k z^{-k}}$

Standard Speech Model for Analysis/Synthesis:



This model provides the basis for a speech analysis/synthesis scheme:

1) Analyze each 20 msec segment of the speech waveform to get:

   a) V/UV decision

   b) Pitch period (if voiced)

   c) $\{a_k\}_{k=1}^{K}$ ~ by solving equations in (2)

   d) Gain ~ G

Transmit a) – d) every 20 msec. At the receiver, reconstruct an approximation to the original speech waveform by using the above model.

<u>Comparison with PCM</u>

PCM: Sample speech at ~ 8 kHz; use 7 bits/sample

$\Rightarrow$ 56 K bits/sec.

Analysis/Synthesis:

(assuming a fancier version of LPC than we just covered)

8 K bits/sec: very close to regular telephone quality

2 K bits/sec: very understandable, somewhat machine-like

600 bits/sec: understandable, quite machine-like

Thus, we see that the LPC scheme can greatly reduce the bit rate for both transmission and storage of speech.