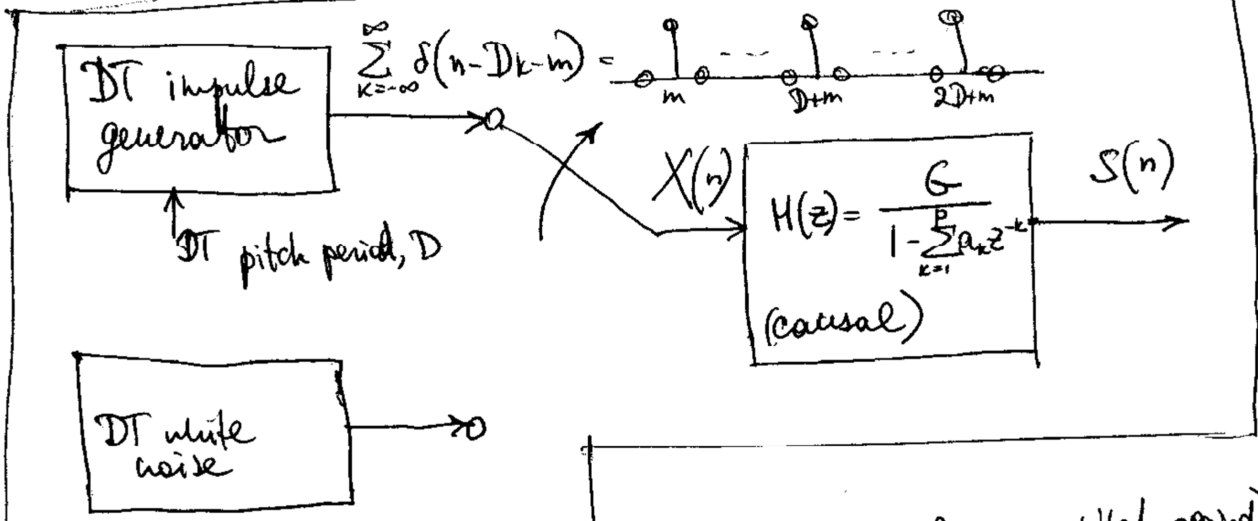
2.1 Autoregressive Model of Speech Production

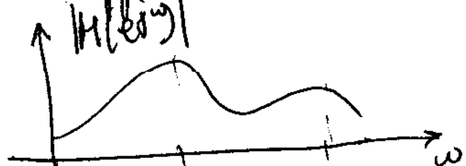
Lec 29, Fri 11/2/01

This is our speech production model from last time, slightly simplified and discretized:



Assume sampling rate $\gg$ pitch (i.e., many samples per pitch period)

Vocal tract: approx. LTI within one phoneme



formant frequencies: resonant frequencies of vocal tract

But $H(e^{j\omega})$ is different for different phonemes.

In particular, different voiced phonemes have different formants.

Model for unvoiced speech: white noise going through a constriction, to produce turbulence.

" " voiced speech: periodic pulses of air excite the vocal chords and are shaped by the vocal tract.

AR model:

$$\boxed{S(n) = GX(n) + \sum_{k=1}^{p} a_k S(n-k)}$$

Why would such a model be useful?

- **Synthesis:**
  model for each phoneme → synthesize speech
  (toy example in Lab 9, wk 1)

- **Coding:**
  1. Identify parameters over short time intervals.
  2. Transmit/store a stream of model parameters: one set per interval, + the error signal.
  3. Reconstruct.

- **Recognition**
  - Store a model for every sound (e.g., every phoneme).
  - Given a speech signal to be identified, estimate "the best" model parameters.
  - match them with the stored ones.

(Also, note that such (and similar) models are used in many other areas besides speech processing (e.g., finance, image processing, etc.))

Question: Given speech signal(s), how do we generate this model?

Need to estimate the following parameters:

1. Voiced/unvoiced (we talked a little about this last time)
2. DT pitch period (for now, assume we know this)
3. Order p (will also assume fixed and given. Can get reasonable models with p around 15.)
4. Gain G
5. Filter coefficients $a_1, \ldots, a_p$ — similar; we assume $G=1$ and try to estimate $a_1, \ldots, a_p$
6. Delay m. (Will model as random).

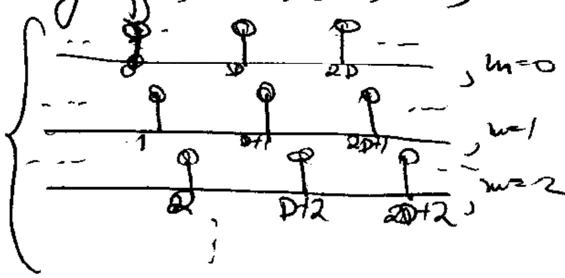
2.1.1 Parameter Estimation for the AR Model.

Observe a voiced sound $S(n)$; want to estimate the corresponding $a_1, \ldots, a_p$.

Problem: we don't know when $X(n)$ started – i.e., don't know m.
One possible solution: model m as random between 0 and $D-1$, i.e., the input to the system is a random sequence:

$$X(n) = x^{(m)}(n) \quad \text{with probability } \frac{1}{D}, \; m = 0, 1, \ldots, D-1,$$

where $x^{(m)}(n) = \sum_{k=-\infty}^{\infty} \delta(n - Dk - m) =$



$$E\left[(X(n))^2\right] = \sum_{m=0}^{D-1} \left(x^{(m)}(n)\right)^2 \frac{1}{D} = \frac{1}{D}$$

$$E\left[X(n)X(n+\ell)\right] = \begin{cases} 0, & \ell \text{ is not a multiple of } D \\ \frac{1}{D}, & \text{otherwise} \end{cases}$$

I.e., $X(n)$ is white over $[0, D-1]$.
We will make a further approximation and <u>model</u> $X(n)$ as a white noise process with variance $\frac{1}{D} = \sigma_X^2$:

$$\xrightarrow[\text{white noise, } \sigma_X^2 = \frac{1}{D}]{X(n),} \boxed{S(n) = X(n) + \sum_{k=1}^{p} a_k S(n-k)} \xrightarrow[\text{WSS random sequence}]{S(n),}$$

What are a_k's?
Multiply both sides by $S(n-\ell)$, and take expected values:

$$E[S(n)S(n-\ell)] = E[X(n)S(n-\ell)] + \sum_{k=1}^{p} a_k E[S(n-k)S(n-\ell)]$$

$$\underbrace{r_{SS}(\ell)}_{\substack{\text{autocorrelation} \\ \text{fcn of } S}} = \underbrace{C_{XS}(-\ell)}_{\substack{\text{cross-correlation} \\ \text{function of } X \text{ and } S}} + \sum_{k=1}^{p} a_k r_{SS}(\ell-k) \qquad (1)$$

Recall that
$$C_{XS}(-\ell) = E[X(n)S(n-\ell)]$$
$$= E\Big[\underbrace{\sum_{k=-\infty}^{\infty} h(k) X(n-\ell-k)}_{S(n-\ell)} X(n)\Big]$$
$$= \sum_{k=-\infty}^{\infty} h(k) r_{XX}(k+\ell)$$

since X is white
$$= \sum_{k=-\infty}^{\infty} h(k) \sigma_X^2 \delta(k+\ell) = \sigma_X^2 h(-\ell)$$

since the system is causal, and since $h(0)=1$
$$= \begin{cases} 0 & \ell>0 \leftarrow S(n) \text{ is uncorrelated with future values of } X(n) \quad (2) \\ \sigma_X^2, & \ell=0 \\ \sigma_X^2 h(-\ell), & \ell<0 \end{cases}$$

Putting (1) and (2) together,

$$r_{SS}(\ell) = \sum_{k=1}^{p} a_k r_{SS}(\ell-k), \quad \ell>0$$

"Yule-Walker equations"

(Also note that
$$r_{SS}(0) = \sum_{k=1}^{p} a_k r_{SS}(0-k) + \sigma_X^2$$
$$= \sum_{k=1}^{p} a_k r_{SS}(k) + \sigma_X^2;$$
and $r_{SS}(\ell) = r_{SS}(-\ell)$; $\ell<0$.)

In matrix form,

$$\begin{pmatrix} r_{ss}(1) \\ r_{ss}(2) \\ \vdots \\ r_{ss}(p) \end{pmatrix} = \begin{pmatrix} r_{ss}(0) & r_{ss}(-1) & \cdots & r_{ss}(-p+1) \\ r_{ss}(1) & r_{ss}(0) & \cdots & r_{ss}(-p+2) \\ \vdots & & & \\ r_{ss}(p-1) & r_{ss}(p-2) & \cdots & r_{ss}(0) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix}$$

- Estimate the autocorrelation of $S(n)$ (recall Lab 7)
- Estimate $a_1, \ldots, a_p$ by solving Yule-Walker equations - i.e., by inverting the autocorrelation matrix
- To synthesise speech using the estimated a_k's, we can excite the system with $\sum_{k=-\infty}^{\infty} \delta(n-Dk)$ - see Lab 9 wk 1, Sect 2.4

(I.e., for system identification we assumed $X(n)$ to be white noise since we didn't know $X(n)$ exactly. But for synthesis can use a more correct $\sum_{k=-\infty}^{\infty} \delta(n-Dk)$.)

2.2 Linear Prediction.

Suppose we observe a WSS random sequence $S(n)$ and would like to predict the next sample as a linear combination of q previous samples:

$$\boxed{\hat{S}(n) = \sum_{k=1}^{q} \alpha_k S(n-k)} \qquad (*)$$

Prediction error: $\mathcal{E}(n) = S(n) - \hat{S}(n)$.

Find the predictor coefficients so as to minimize the mean square error, $e = E[(\mathcal{E}(n))^2]$.

$$\frac{\partial e}{\partial \alpha_\ell} = \frac{\partial}{\partial \alpha_\ell} E\left\{\left(S(n) - \sum_{k=1}^{q} \alpha_k S(n-k)\right)^2\right\}$$

$$= E\left\{2\left(S(n) - \sum_{k=1}^{q} \alpha_k S(n-k)\right)\left(-S(n-\ell)\right)\right\}$$

$$= -2 r_{SS}(\ell) + 2\sum_{k=1}^{p} \alpha_k r_{SS}(\ell-k)$$

Set this to zero:

$$\boxed{r_{SS}(\ell) = \sum_{k=1}^{q} \alpha_k r_{SS}(\ell-k), \quad \ell = 1, \ldots, p}$$

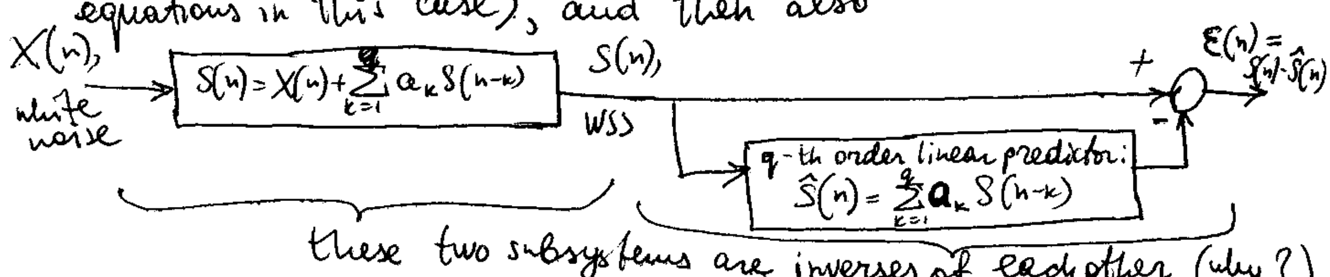
- normal equations

Remarks:

1. $\frac{\partial^2 e}{\partial \alpha_\ell^2} = E[2(S(n-\ell))^2] > 0$ unless $S(n) \equiv 0$

   $\Rightarrow$ what we found is a minimum.

2. If $S(n)$ is a q-th order AR process, then $\alpha_k = a_k$ (the normal equations are identical to the Yule-Walker equations in this case), and then also



these two subsystems are inverses of each other (why?)

Then $\mathcal{E}(n) = X(n) =$ white noise.
The linear predictor acts as as "whitening filter", i.e., removes correlation from $S(n)$, producing a white error process.

3. Let's step back and think what exactly we are doing here. ~~Because~~ This prediction problem is a linear least-squares estimation problem:
we are trying to estimate random variable $S(n)$ as a linear combination of the random variables $S(n-1), \ldots, S(n-p)$, by minimizing the mean-square error.