Lecture 17: LPC speech synthesis and autocorrelation-based pitch tracking

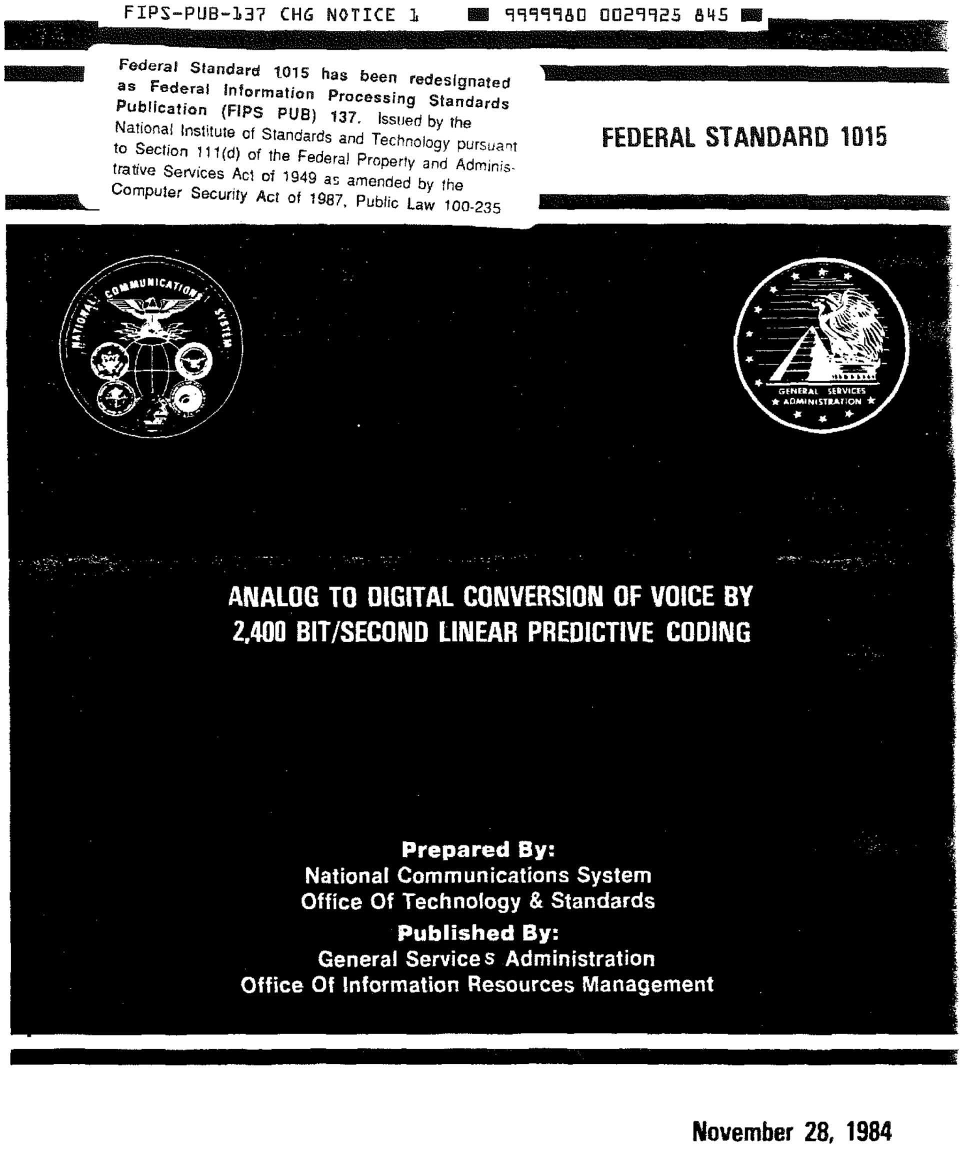
ECE 401, Signal and Image Analysis

November 5, 2020

Outline

- The LPC-10 speech synthesis model
- The LPC-10 excitation model: white noise, pulse train
- Linear predictive coding: how to find the coefficients
- Linear predictive coding: how to make sure the coefficients are stable
- Autocorrelation-based pitch tracking
- Inter-frame interpolation of pitch and energy contours

The LPC-10 speech synthesis model



FIPS-PUB-137 CHG NOTICE 1 ■ 9999980 0029925 845 ■

Federal Standard 1015 has been redesignated as Federal Information Processing Standards Publication (FIPS PUB) 137. Issued by the National Institute of Standards and Technology pursuant to Section 111(d) of the Federal Property and Administrative Services Act of 1949 as amended by the Computer Security Act of 1987, Public Law 100-235

FEDERAL STANDARD 1015

ANALOG TO DIGITAL CONVERSION OF VOICE BY
2,400 BIT/SECOND LINEAR PREDICTIVE CODING

Prepared By:
National Communications System
Office Of Technology & Standards
Published By:
General Services Administration
Office Of Information Resources Management

November 28, 1984

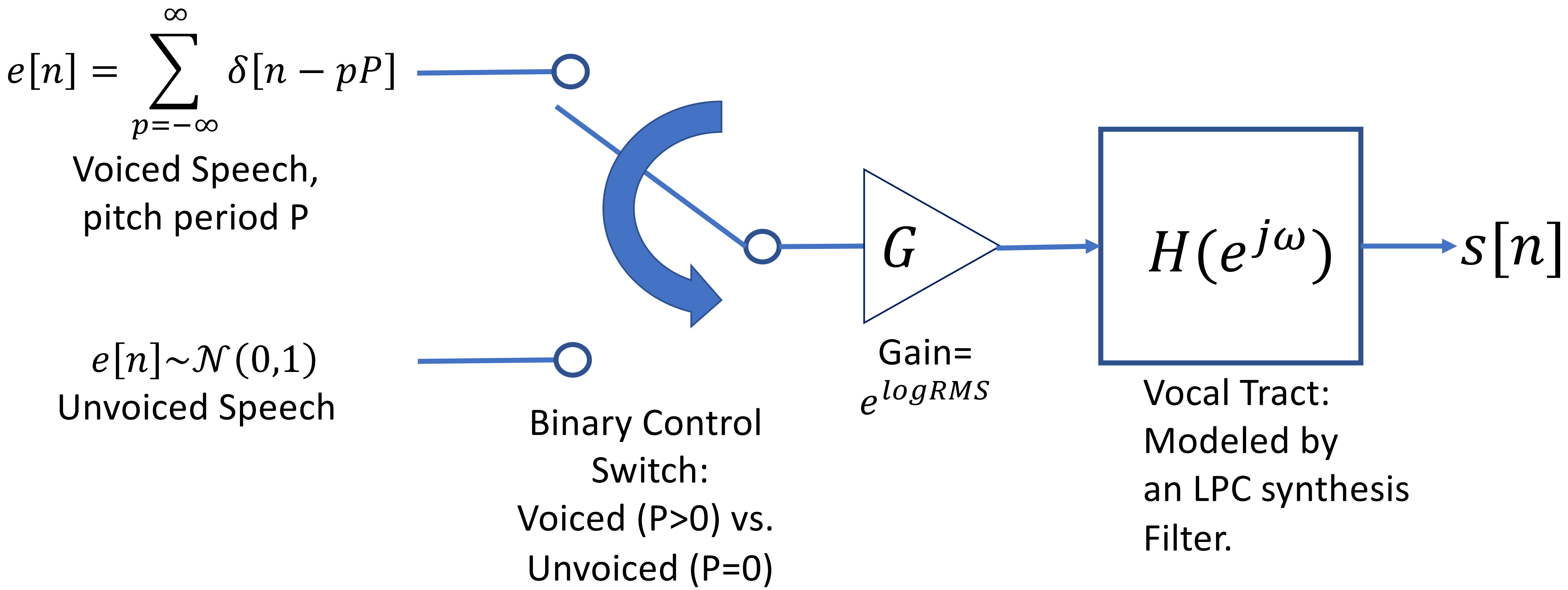
The LPC-10 Speech Coder: Transmitted Parameters

Each frame is 54 bits, and is used to synthesize 22.5ms of speech.

(54 bits/frame)/(0.0225 seconds/frame)=2400 bits/second

- **Pitch**: 7 bits/frame (127 distinguishable non-zero pitch periods)
- **Energy**: 5 bits/frame (32 levels, on a log-energy scale)
- **10 linear predictive coefficients** (LPC): 41 bits/frame
- Synchronization: 1 bit/frame

The LPC-10 speech synthesis model



$$e[n] = \sum_{p=-\infty}^{\infty} \delta[n - pP]$$

Voiced Speech,
pitch period P

$$e[n] \sim \mathcal{N}(0,1)$$

Unvoiced Speech

Binary Control
Switch:
Voiced (P>0) vs.
Unvoiced (P=0)

G

Gain=
e^{logRMS}

$H(e^{j\omega})$

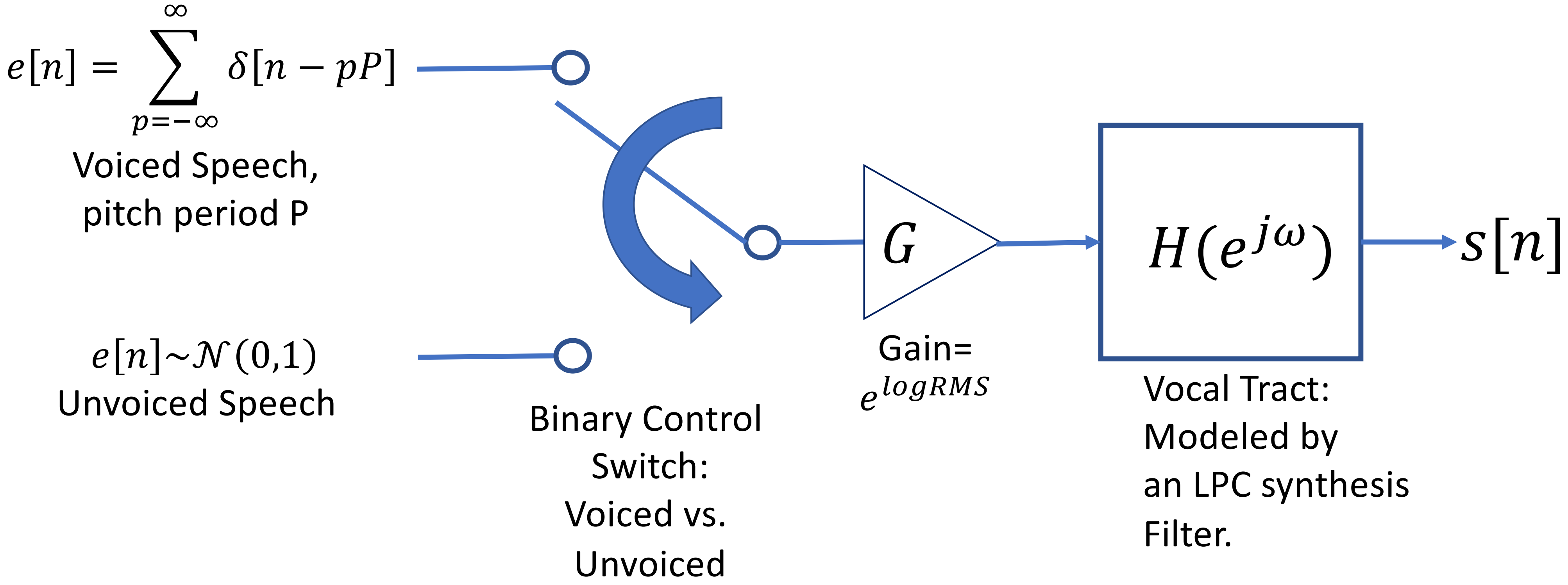
Vocal Tract:
Modeled by
an LPC synthesis
Filter.

$s[n]$

Outline

- The LPC-10 speech synthesis model
- The LPC-10 excitation model: white noise, pulse train
- Linear predictive coding: how to find the coefficients
- Linear predictive coding: how to make sure the coefficients are stable
- Autocorrelation-based pitch tracking
- Inter-frame interpolation of pitch and energy contours

The LPC-10 speech synthesis model



$$e[n] = \sum_{p=-\infty}^{\infty} \delta[n - pP]$$

Voiced Speech,
pitch period P

$e[n] \sim \mathcal{N}(0,1)$
Unvoiced Speech

Binary Control
Switch:
Voiced vs.
Unvoiced

G

Gain=
e^{logRMS}

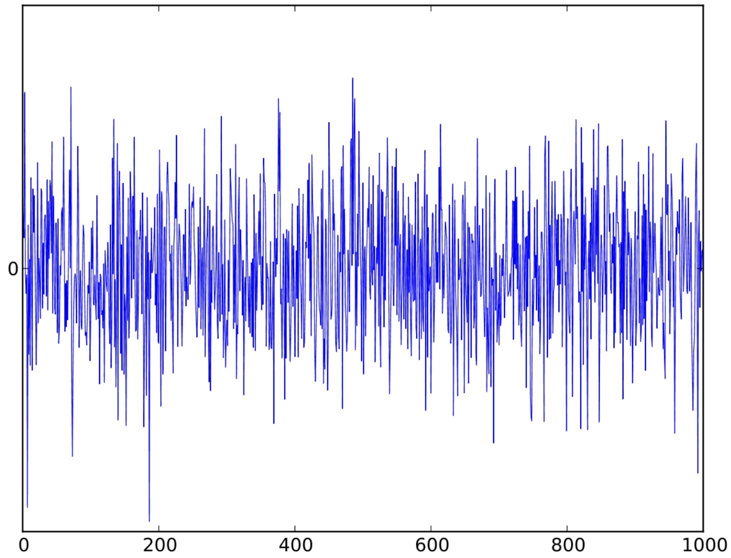
$H(e^{j\omega})$

Vocal Tract:
Modeled by
an LPC synthesis
Filter.

$s[n]$

Unvoiced speech: e[n]=white noise

- Use zero-mean, unit-variance Gaussian white noise
- The choice, to use “unvoiced speech,” is communicated by the special code word “P=0”



By Morn - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=24084756

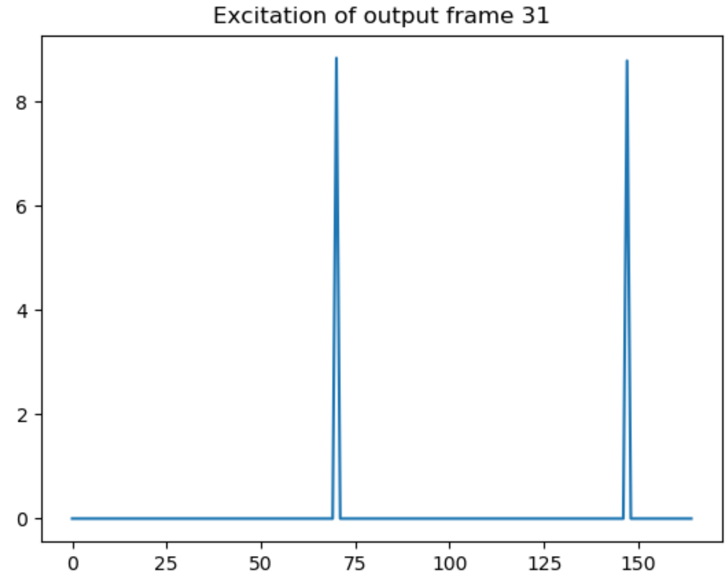
Voiced speech: e[n]=pulse train

- The basic idea:

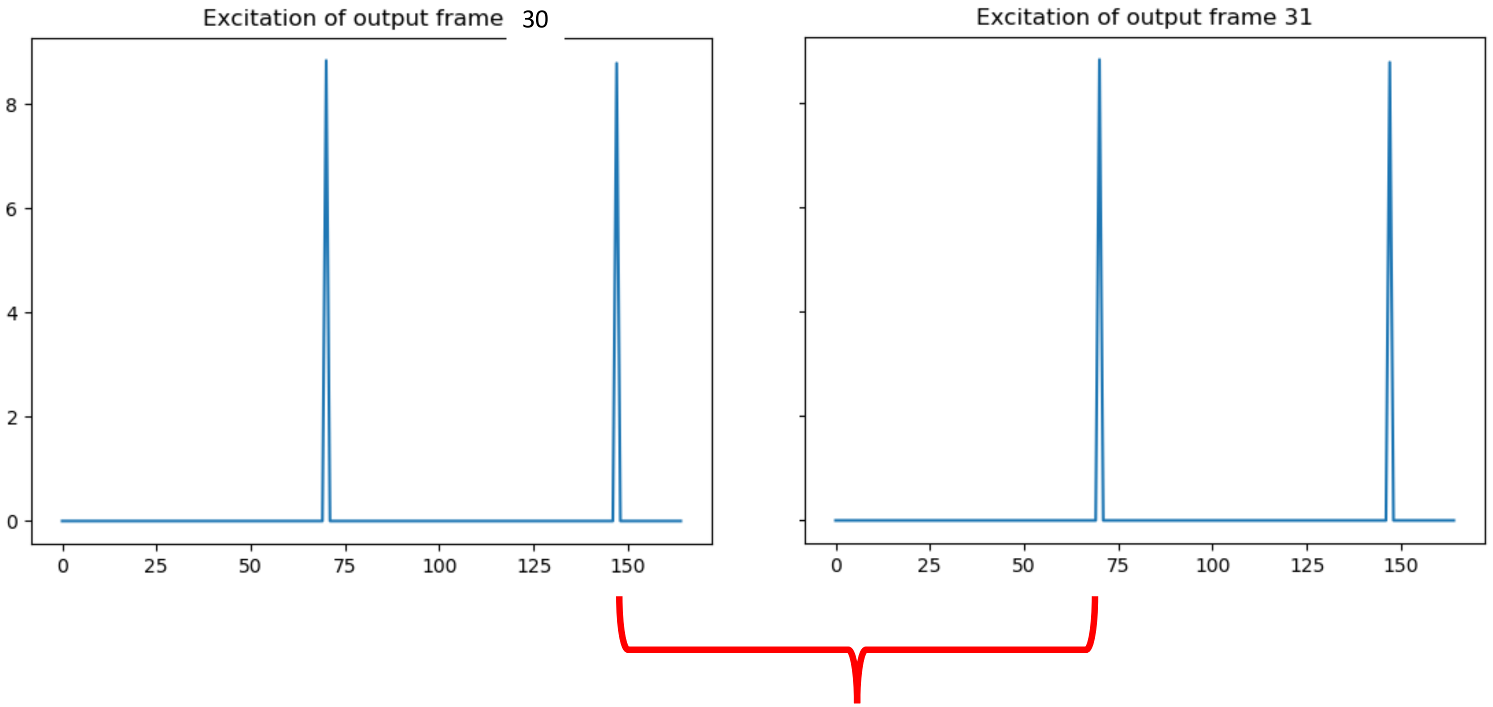
$$e[n] = \sum_{p=-\infty}^{\infty} \delta[n - pP]$$

- Modification #1: in order for the average energy to equal 1.0, we need to scale each pulse by $\sqrt{P}$:

$$e[n] = \sqrt{P} \sum_{p=-\infty}^{\infty} \delta[n - pP]$$



Modification #2: the first pulse is not at n=0



Pitch period = 80 samples ⇒ first pulse in frame 31 can't occur until the 70th sample of the frame

A mechanism for keeping track of pitch phase from one frame to the next

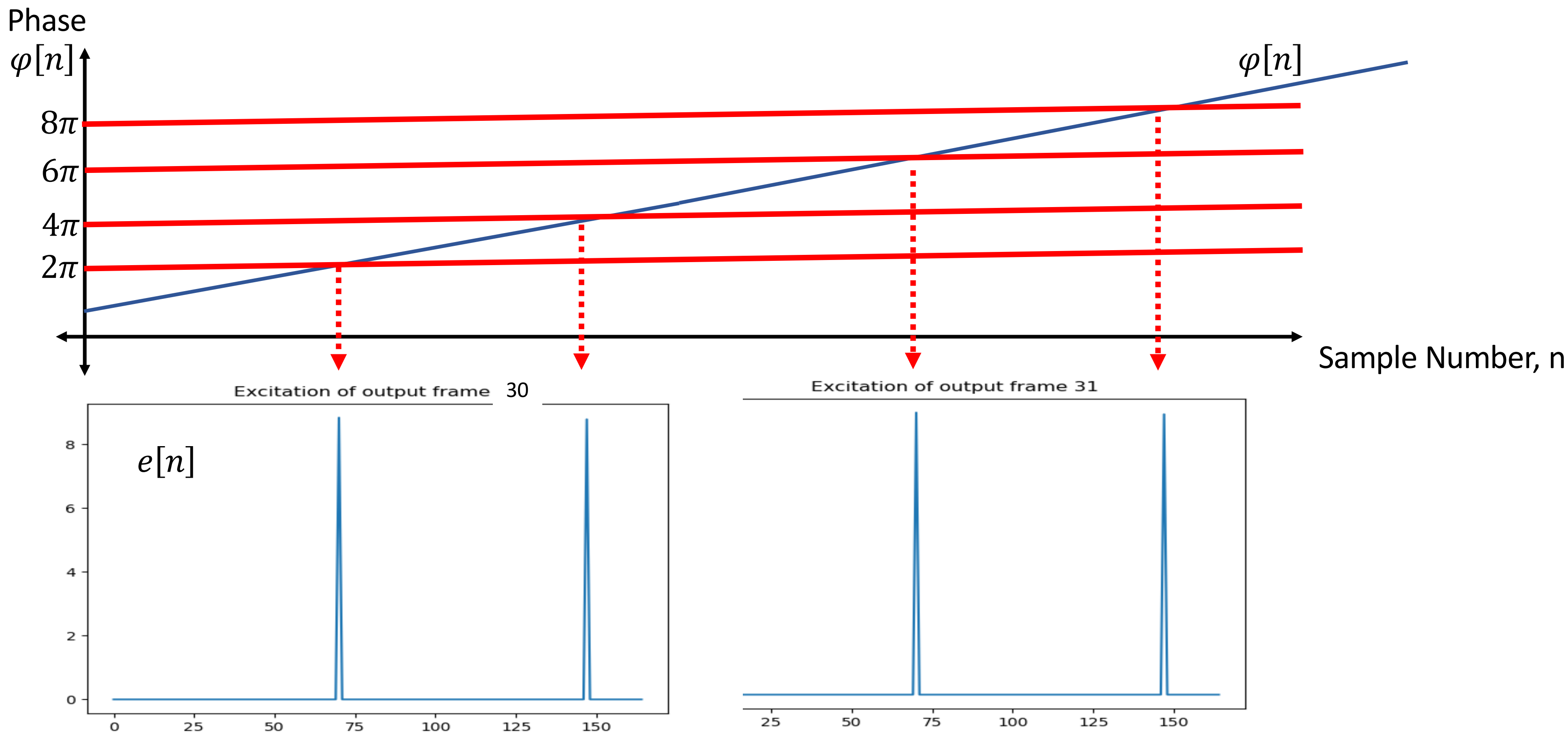
- Start out, at the beginning of the speech, with a pitch phase equal to zero, $\varphi[0] = 0$
- For every sample thereafter:
  - If the sample is unvoiced (P[n]=0), don't increment the pitch phase
  - If the sample is voiced (P[n]>0), then increment the pitch phase

$$\varphi[n] = \varphi[n-1] + \frac{2\pi}{P[n]}$$

- Every time the phase passes a multiple of 2π, output a pitch pulse

$$e[n] = \begin{cases} \sqrt{P} & \left\lfloor \frac{\varphi[n]}{2\pi} \right\rfloor - \left\lfloor \frac{\varphi[n-1]}{2\pi} \right\rfloor > 0 \\ 0 & else \end{cases}$$

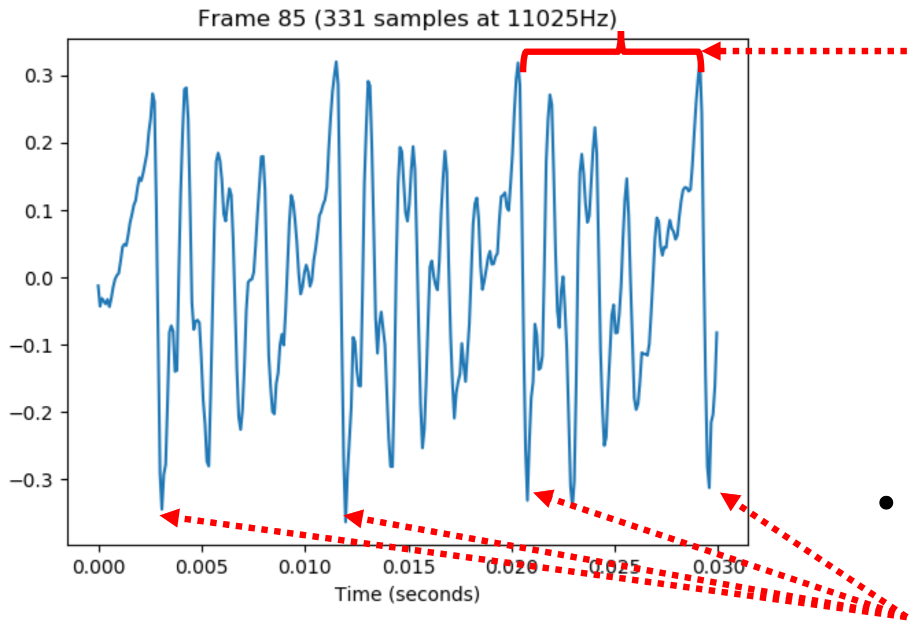
The pitch phase method: generate an excitation pulse whenever pitch phase crosses a 2π-level



Outline

- The LPC-10 speech synthesis model
- The LPC-10 excitation model: white noise, pulse train
- Linear predictive coding: how to find the coefficients
- Linear predictive coding: how to make sure the coefficients are stable
- Autocorrelation-based pitch tracking
- Inter-frame interpolation of pitch and energy contours

Speech is predictable

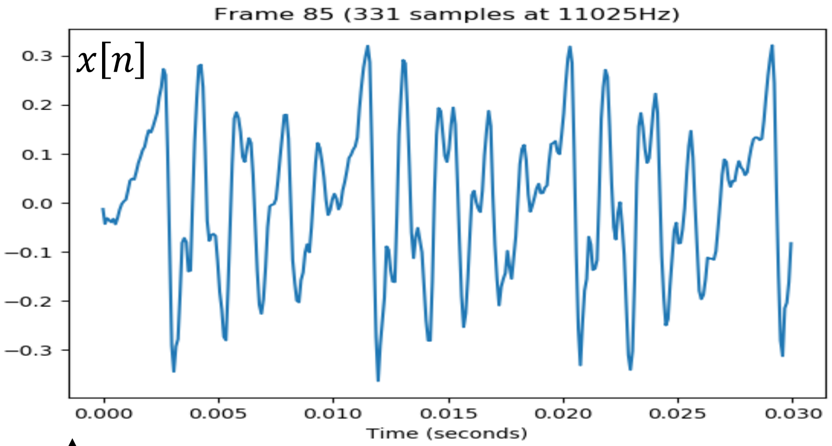


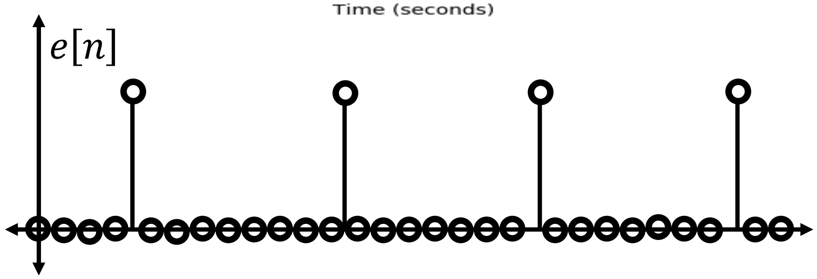
- Speech is not just white noise and pulse train. In fact, each sample is highly predictable from previous samples.

$$x[n] \approx \sum_{m=1}^{10} \alpha_m x[n-m]$$

- In fact, the pitch pulses are the only major exception to this predictability!

Linear predictive coding (LPC)





The LPC idea:

1. Model the excitation as error

$$e[n] = x[n] - \sum_{m=1}^{10} \alpha_m x[n-m]$$

2. Force the coefficients α_m to explain as much as they can, so that $e[n]$ is as close to zero as possible.

Linear predictive coding (LPC)

$$\varepsilon = E[e^2[n]] = E\left[\left(x[n] - \sum_{i=1}^{10} \alpha_i x[n-i]\right)^2\right]$$

$$\frac{\partial \varepsilon}{\partial \alpha_j} = -2E\left[x[n-j]\left(x[n] - \sum_{i=1}^{10} \alpha_i x[n-i]\right)\right]$$

Setting $\frac{\partial \varepsilon}{\partial \alpha_j} = 0$ gives

$$E[x[n-j]x[n]\,] = \sum_{i=1}^{10} \alpha_i E[x[n-j]x[n-i]\,]$$

$R_{xx}[j]$ $\qquad\qquad$ $R_{xx}[|i-j|]$

Linear predictive coding (LPC)

So we have a set of linked equations, for $1 \leq j \leq 10$:

$$R_{xx}[j] = \sum_{i=1}^{10} \alpha_i R_{xx}[|i-j|]$$

- We can write these 10 equations as a 10x10 matrix equation: $\vec{\gamma} = R\vec{\alpha}$
- …which immediately gives the solution: $\vec{\alpha} = R^{-1}\vec{\gamma}$
- …where

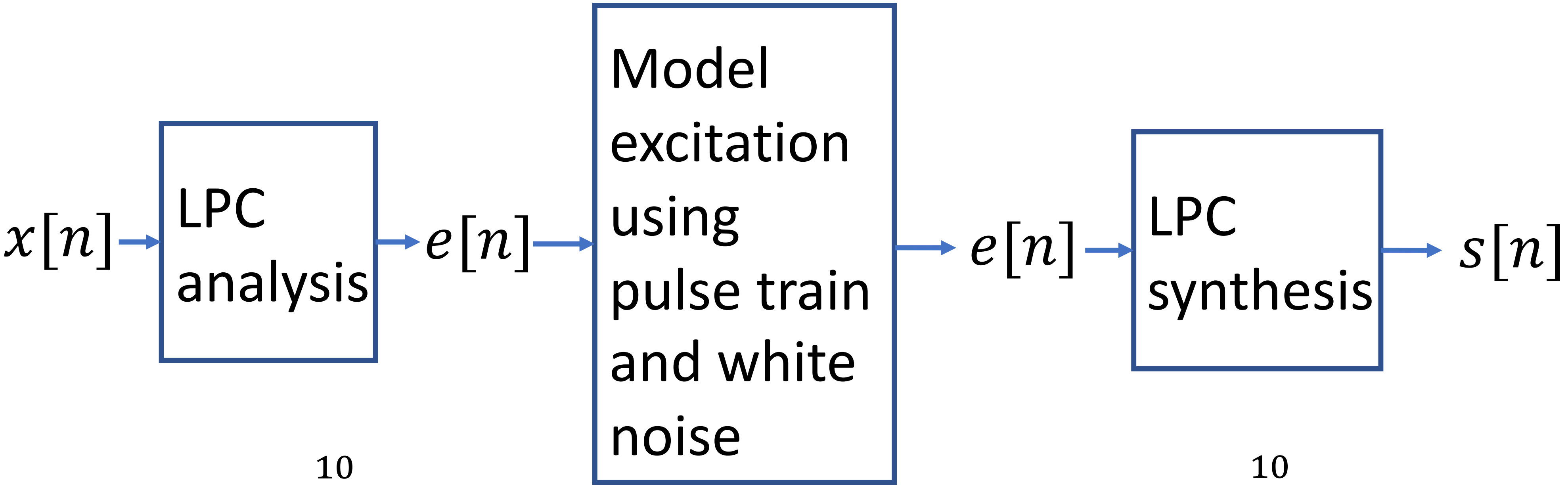
$$\vec{\gamma} = \begin{bmatrix} R_{xx}[1] \\ \vdots \\ R_{xx}[10] \end{bmatrix}, \qquad R = \begin{bmatrix} R_{xx}[0] & R_{xx}[1] & \cdots \\ R_{xx}[1] & R_{xx}[0] & \cdots \\ \vdots & \vdots & R_{xx}[0] \end{bmatrix}, \qquad \vec{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_{10} \end{bmatrix}$$

Outline

- The LPC-10 speech synthesis model
- The LPC-10 excitation model: white noise, pulse train
- Linear predictive coding: how to find the coefficients
- Linear predictive coding: how to make sure the coefficients are stable
- Autocorrelation-based pitch tracking
- Inter-frame interpolation of pitch and energy contours

Speech -> Excitation -> Speech

Now that we know how to find the LPC coefficients, we can imagine an end-to-end LPC analysis-by-synthesis:



$x[n]$

LPC analysis

$e[n]$

Model excitation using pulse train and white noise

$e[n]$

LPC synthesis

$s[n]$

$$e[n] = x[n] - \sum_{m=1}^{10} \alpha_m x[n-m]$$

$$s[n] = e[n] + \sum_{m=1}^{10} \alpha_m s[n-m]$$

The LPC Analysis Filter

The LPC Analysis Filter is an all-zeros filter (FIR = finite impulse response):

$$e[n] = x[n] - \sum_{m=1}^{10} \alpha_m x[n-m] \leftrightarrow E(z) = A(z)X(z)$$

…where…

$$A(z) = 1 - \sum_{m=1}^{10} \alpha_m z^{-m}$$

The LPC Synthesis Filter

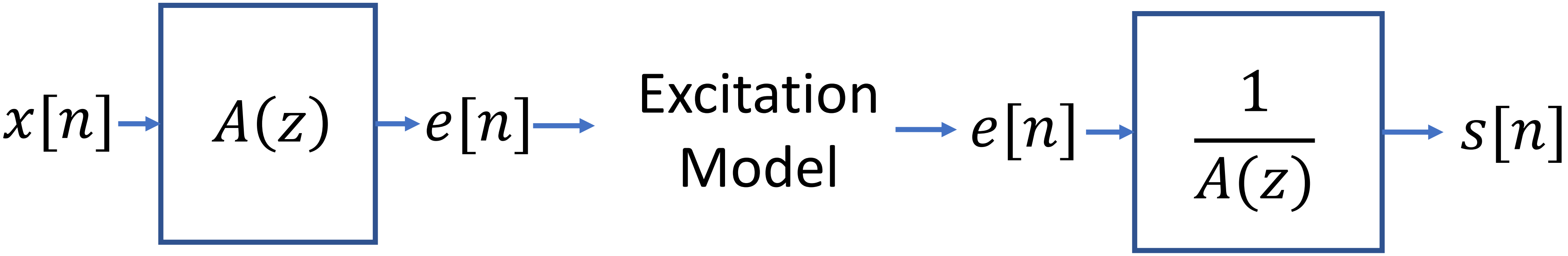
The LPC Synthesis Filter is an all-poles filter (IIR = infinite impulse response):

$$s[n] = e[n] + \sum_{m=1}^{10} \alpha_m s[n-m] \leftrightarrow S(z) = H(z)E(z)$$

…where…

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 - \sum_{m=1}^{10} \alpha_m z^{-m}}$$

Speech -> Excitation -> Speech



$x[n]$ $A(z)$ $e[n]$

Excitation Model

$e[n]$ $\frac{1}{A(z)}$ $s[n]$

The Stability Problem

- The analysis filter is guaranteed to be stable, as long as the coefficients are finite. Suppose you know that $|x[n]| \leq X_{MAX}$, and $|\alpha_m| \leq \alpha_{MAX}$. Then, even in the worst possible case, $|e[n]| \leq 11\alpha_{MAX}X_{MAX}$.
- The synthesis filter has no such guarantee. For example, suppose $e[n]$ is just a delta function ($e[n] = \delta[n]$), and suppose all of the $\alpha_m = 0$ except the first one, $\alpha_1 = 1.1$. Then

$$s[n] = \delta[n] + 1.1s[n-1] \ = (1.1)^n$$

Which overflows your 16-bit sample buffer after only 110 samples. Your output will be full of NaNs, and you'll be saying "What happened…?"

How to Guarantee Stability

Fortunately, the LPC synthesis filter is causal, so it’s easy to guarantee stability. We just need to make sure that all of the poles have magnitude less than 1:

$$|r_k| < 1$$

We find the poles like this:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 - \sum_{m=1}^{10} \alpha_m z^{-m}} = \frac{1}{\prod_{k=1}^{10}(1 - r_k z^{-1})}$$

in other words,

$$r_k = roots(A(z))$$

…which you can do using np.roots, if you define the polynomial correctly. Then you just truncate the magnitude,

$$r_k \leftarrow \min(|r_k|, 0.999) e^{j\measuredangle r_k}$$

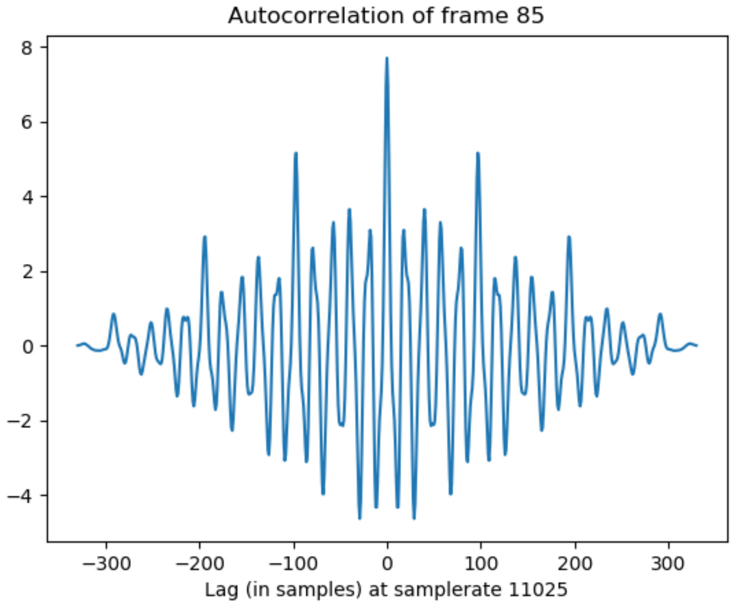
…and then use np.poly to convert back from roots to polynomial.

Outline

- The LPC-10 speech synthesis model
- The LPC-10 excitation model: white noise, pulse train
- Linear predictive coding: how to find the coefficients
- Linear predictive coding: how to make sure the coefficients are stable
- Autocorrelation-based pitch tracking
- Inter-frame interpolation of pitch and energy contours

Autocorrelation is maximum at n=0

$$r_{xx}[n] = \sum_{m=-\infty}^{\infty} x[m]x[m-n]$$

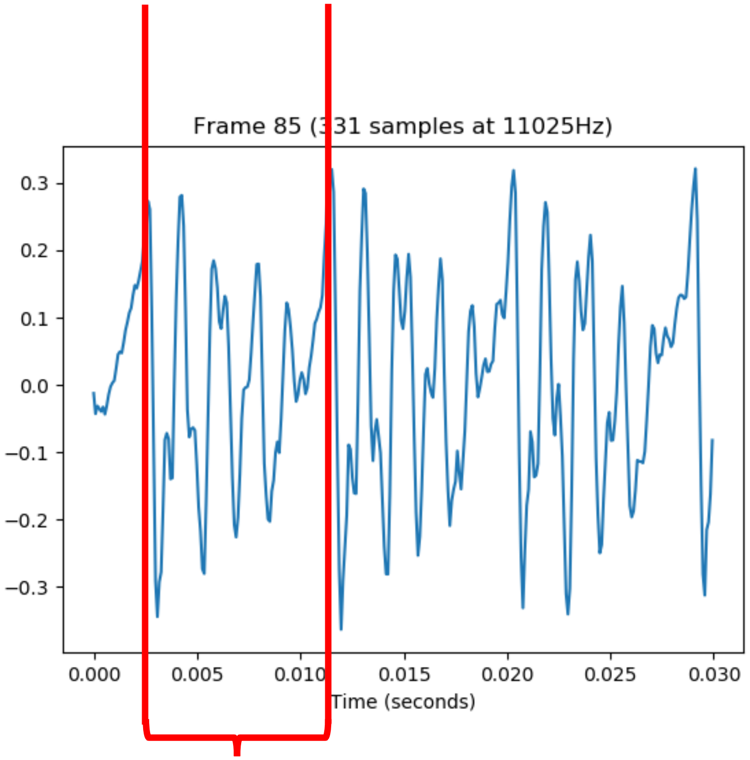


Autocorrelation of a periodic signal

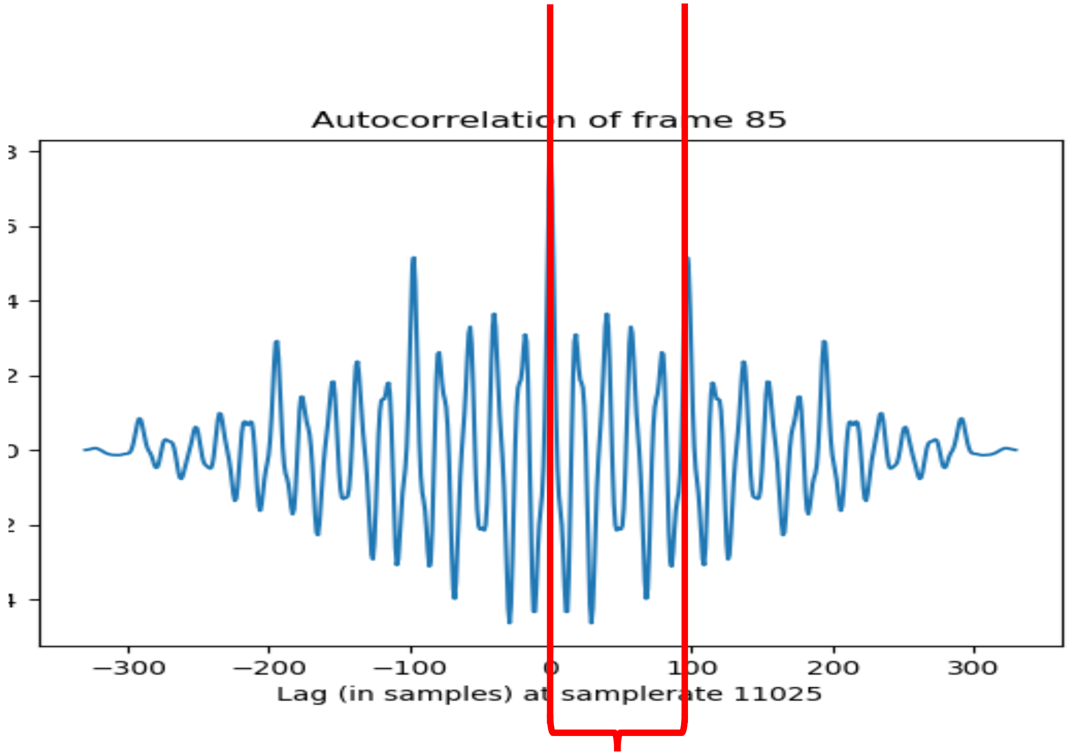
Suppose x[n] is periodic, $x[n] = x[n-P]$. Then the autocorrelation is also periodic:

$$r_{xx}[P] = \sum_{m=-\infty}^{\infty} x[m]x[m-P] = \sum_{m=-\infty}^{\infty} x^2[m] = r_{xx}[0]$$

Autocorrelation of a periodic signal is periodic



Pitch period = 9ms = 99 samples



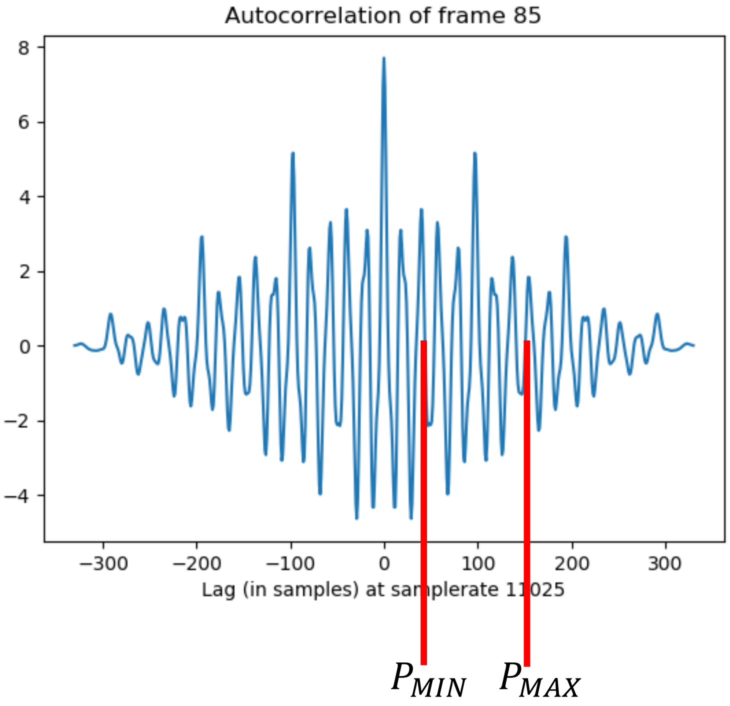
Pitch period = 9ms = 99 samples

Autocorrelation pitch tracking

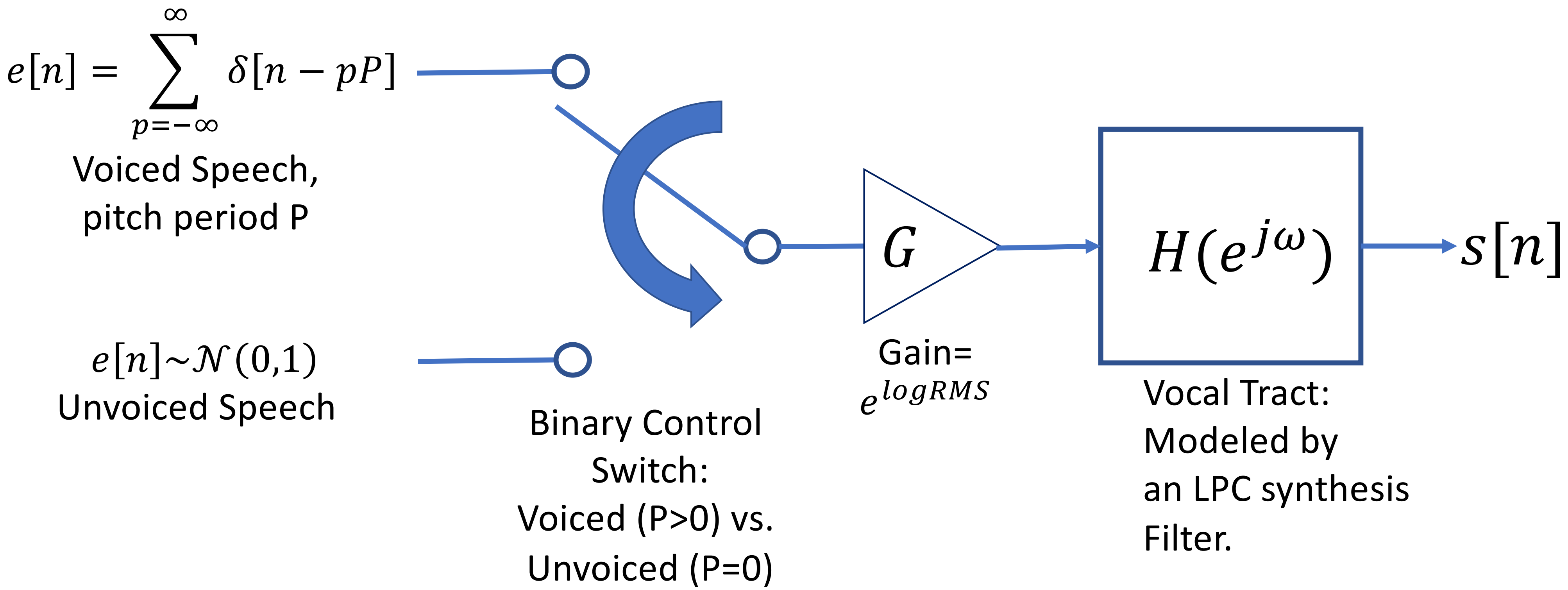
- Compute the autocorrelation
- Find the pitch period:

$$P = \underset{P_{MIN} \le m \le P_{MAX}}{\operatorname{argmax}} \; r_{xx}[m]$$

- The search limits, P_{MIN} and P_{MAX}, are important for good performance:
  - P_{MIN} corresponds to a high woman's pitch, about $F_S/P_{MIN} \approx 250$ Hz
  - P_{MAX} corresponds to a low man's pitch, about $F_S/P_{MAX} \approx 80$ Hz



The LPC-10 speech synthesis model



$$e[n] = \sum_{p=-\infty}^{\infty} \delta[n - pP]$$

Voiced Speech,
pitch period P

$e[n] \sim \mathcal{N}(0,1)$
Unvoiced Speech

Binary Control
Switch:
Voiced (P>0) vs.
Unvoiced (P=0)

G

Gain=
e^{logRMS}

$H(e^{j\omega})$

Vocal Tract:
Modeled by
an LPC synthesis
Filter.

$s[n]$

The voiced/unvoiced decision

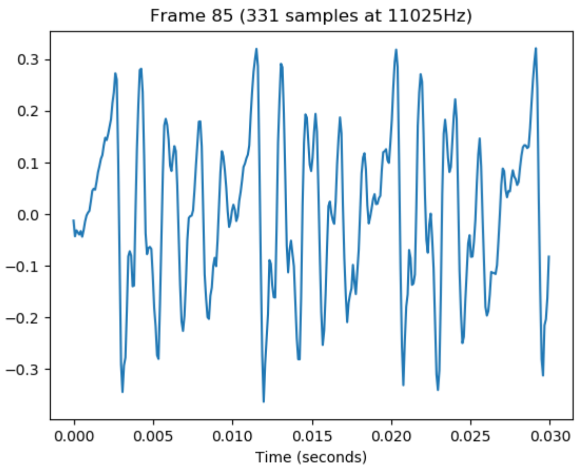
- $x[n]$ voiced: $r_{xx}[P] \approx r_{xx}[0]$
- $x[n]$ unvoiced (white noise):
  $r_{xx}[n] \approx \delta[n]$
  which means that $r_{xx}[P] \ll r_{xx}[0]$

So a reasonable V/UV decision is:

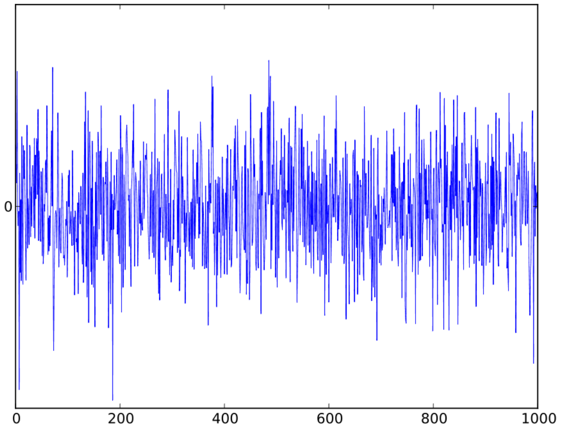
- $\frac{r_{xx}[P]}{r_{xx}[0]} \geq threshold$: say the frame is voiced.
- $\frac{r_{xx}[P]}{r_{xx}[0]} < threshold$: say the frame is unvoiced.

Setting threshold~0.25 works reasonably well.

voiced: $x[n+P] \approx x[n]$



unvoiced:
$\mathrm{E}[x[m]x[m-n]] \approx \delta[n]$

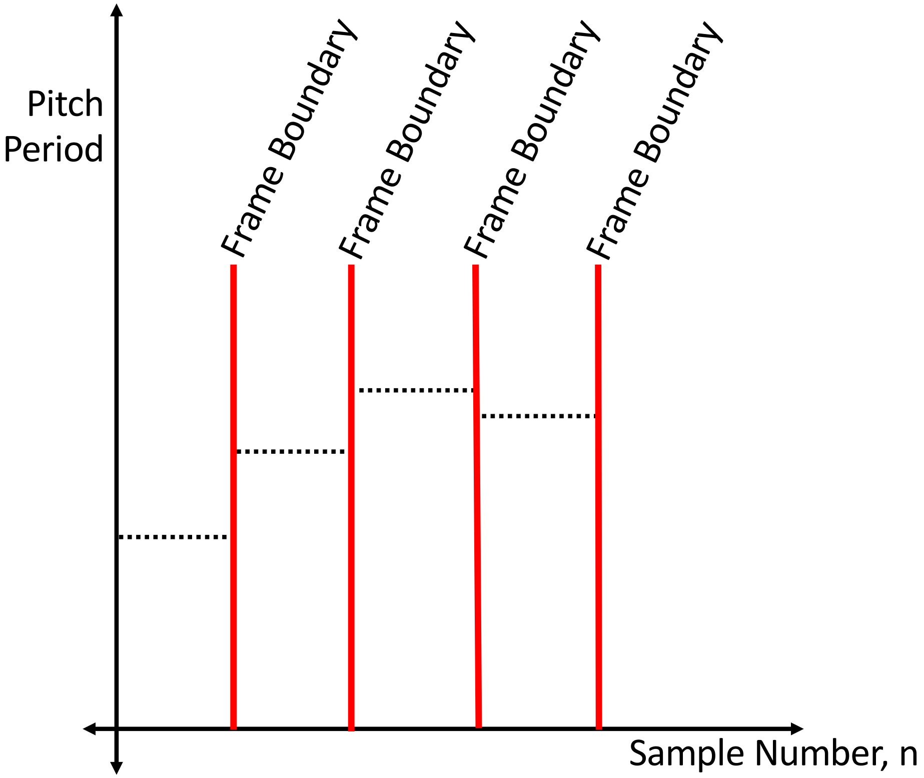


Outline

- The LPC-10 speech synthesis model
- The LPC-10 excitation model: white noise, pulse train
- Linear predictive coding: how to find the coefficients
- Linear predictive coding: how to make sure the coefficients are stable
- Autocorrelation-based pitch tracking
- Inter-frame interpolation of pitch and energy contours

Inter-frame interpolation of pitch contours

We don’t want the pitch period to change suddenly at frame boundaries; it sounds weird.



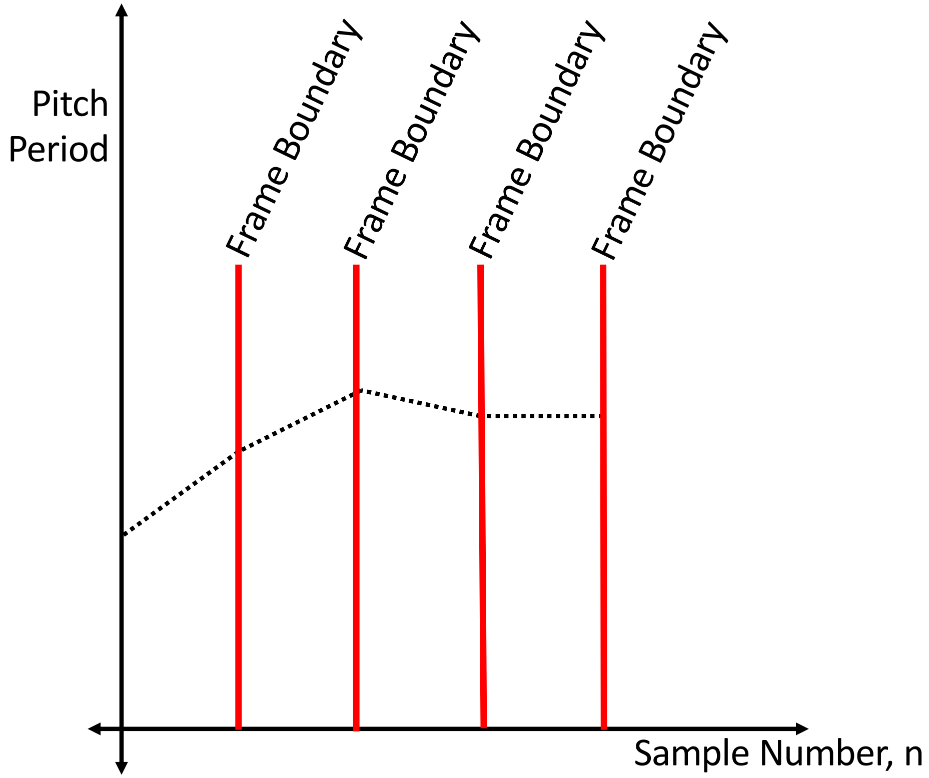
Inter-frame interpolation of pitch contours

Linear interpolation sounds much better. We can accomplish linear interpolation using a formula like

$$P[n] = (1 - f)P_t + fP_{t+1}$$

Where

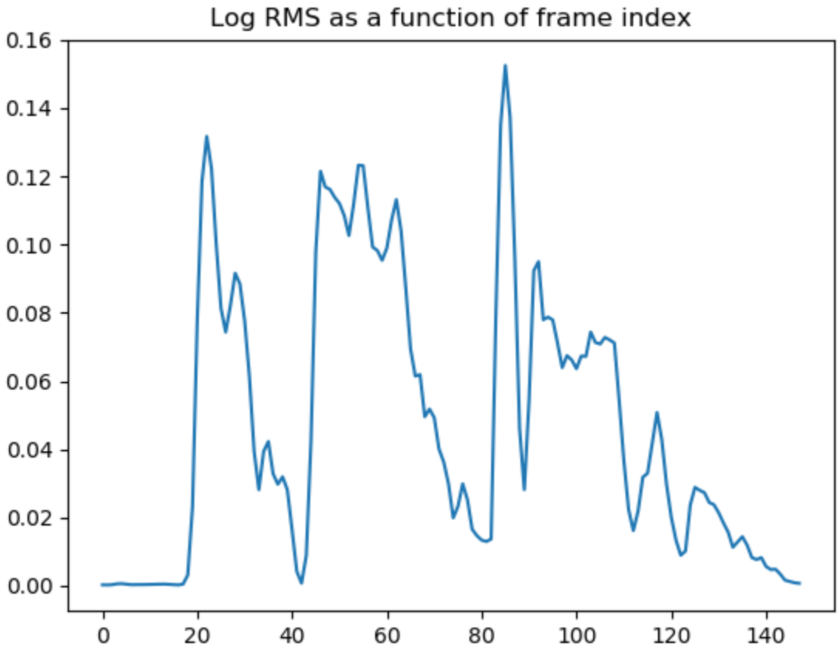
- P_t is the pitch period in frame t
- $f = \frac{n-tS}{S}$ is how far sample n is from the beginning of frame t
- S is the frame-skip.



Inter-frame interpolation of energy

Linear interpolation is also useful for energy, EXCEPT: it sounds better if we interpolate log energy, not energy.

$$\log RMS_t = \log \sqrt{\frac{1}{L}\sum_{n=tS}^{tS+L-1} x^2[n]}$$



Outline

- The LPC-10 speech synthesis model
- The LPC-10 excitation model: white noise, pulse train
- Linear predictive coding: how to find the coefficients
- Linear predictive coding: how to make sure the coefficients are stable
- Autocorrelation-based pitch tracking
- Inter-frame interpolation of pitch and energy contours