

Speech Synthesis

—

**The Art of Creating
Computer Speech**

History of Speech Synthesis

- **18th and 19th century:** mechanical devices for the production of artificial speech sounds
- **1936:**
 - *AT&T's Bell Labs produced the first electronic speech synthesiser.*
 - *This 'voice coder', or Voder, involved an operator manipulating a keyboard and foot pedals to shape the emitted sounds into recognisable speech.*
- **1970s:** Texas Instruments speech chip with the 'Speak and Spell' toy.
- **Current:** BT directory enquiries; railway announcements; speaking clock; aid to handicapped persons (astro-physicist Stephen Hawking) ...

Microsoft Speech Engines

- ***MS API*** – ***Microsoft Speech Application Interface***
- ***SAPI SDK Version 4***
- ***LISSET*** – ***Linguistic Information***
- ***WaveEdit***
- ***MS Agents***
- ***Demo of speech outputs***

Two kinds of Text-To-Speech Synthesis (TTS)

- Pre-recorded *words* or *phrases* are **concatenated** to produce the utterance (train announcements, telephone services, talking toys etc.)
- Automatic production of new sentences (much more difficult)
 - Stephen Hawking's voice

Approches to Speech Synthesis

- A process of producing acoustic signal by **controlling a model of speech production with a set of parameters**
- **Two major approaches:**
 - **Articulatory speech synthesis:**
 - To model the speech system in details, such as:
 - *the **motion** of the speech articulators,*
 - *the generation and propagation of sound inside the vocal tract.*
 - Still a research topics
 - **Terminal-analogue synthesis:**
 - To copy the **frequency characteristic** of the vocal tract.
 - This is based on the **source/filter model**
- **Only the second approach will be followed.**

How the machine converts text to sound?

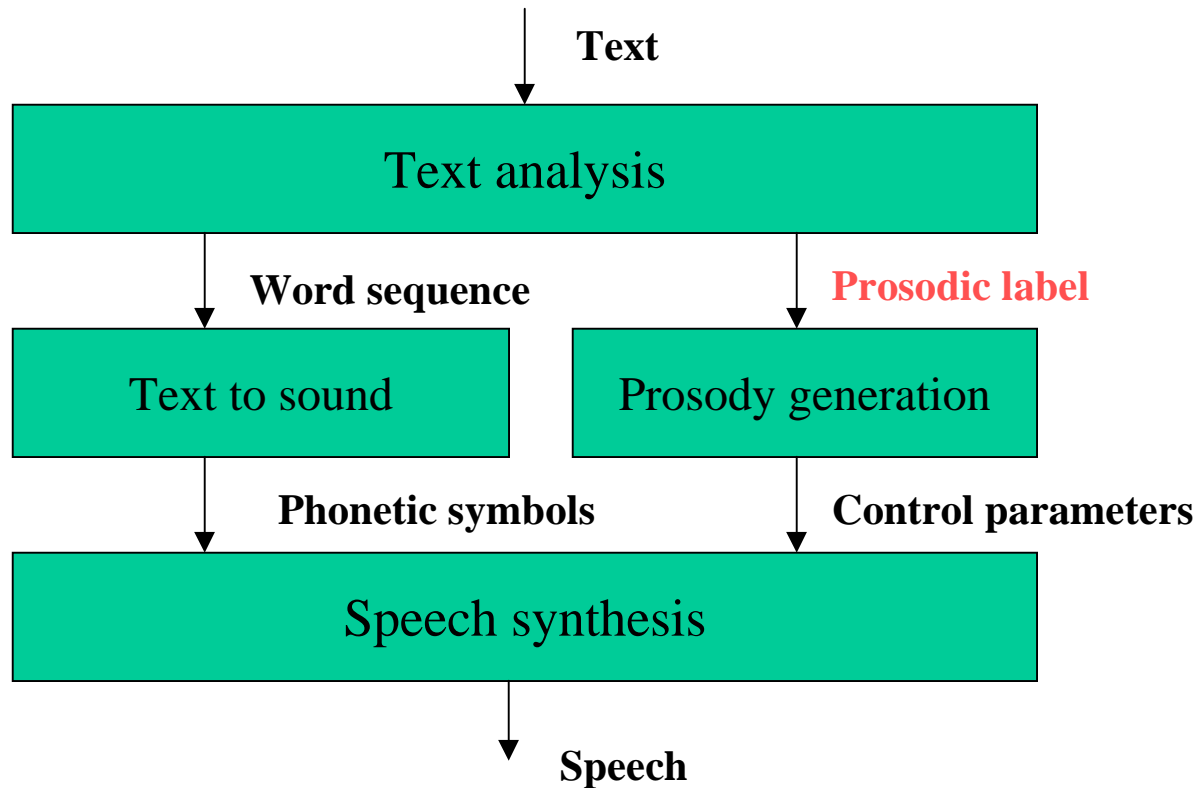
TEXT ⇒ Natural Language Processing ⇒ Digital Signal Processing ⇒ Sound

Written text is converted to acoustic output by:

- text **analysis**
- **letter to sound** conversion
- **parsing** (for grammatical structure, emphasis)
- assignment of **prosodic structure** (e.g. intonation, rhythm)

Segments are then linked to tables of parameters that describe acoustic output

Text to speech system model



Use Analysis for synthesis

- *Speech analysis*

- Analyze speech, identify unvoiced part and each pitch of voiced part, etc
- Store them as *synthesis units*.

- *Speech synthesis*

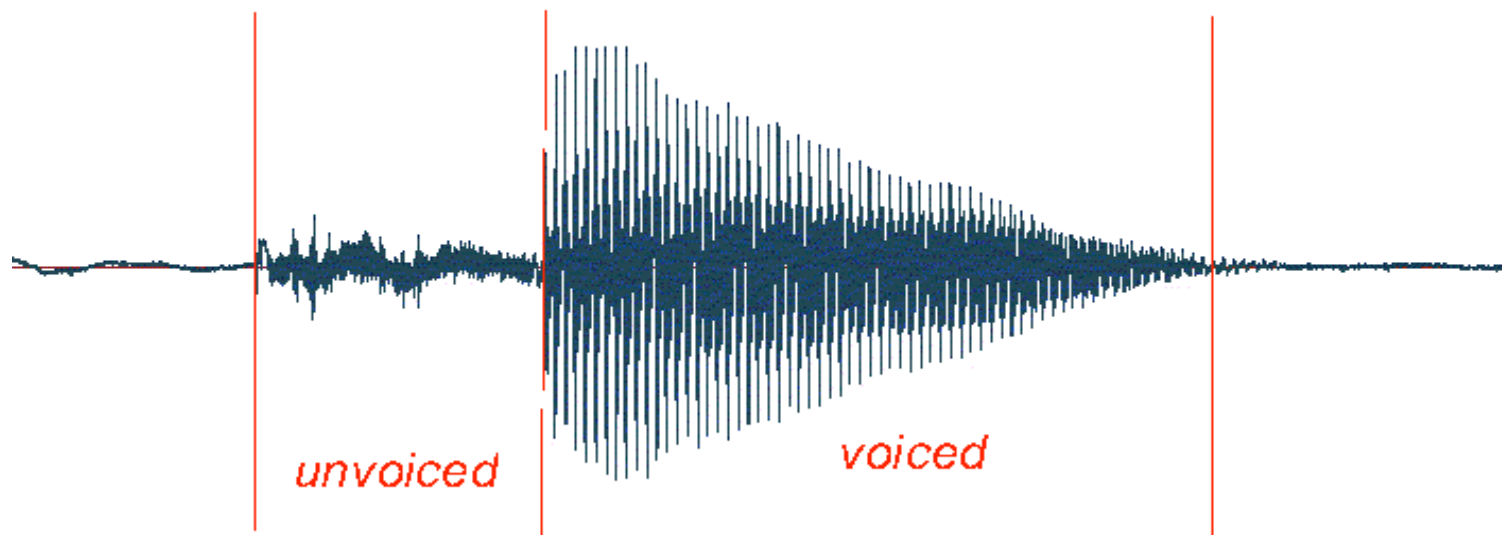
- Input: Prosody control parameters, phonetic transcripts.
- Generate speech using both:
 - synthesis unit **from analysis**
 - prosodic control parameters.

Analysis: Problems

- **Problems:**
 - **Speech corpus:**
 - sentence, word, syllable
 - **Determine Synthesis Unit :**
 - syllable, diphone, etc
- **Process:**
 - **voiced/unvoiced** determination,
 - **Pitch** marking,
 - Store all the speech pieces to create **unit database**.

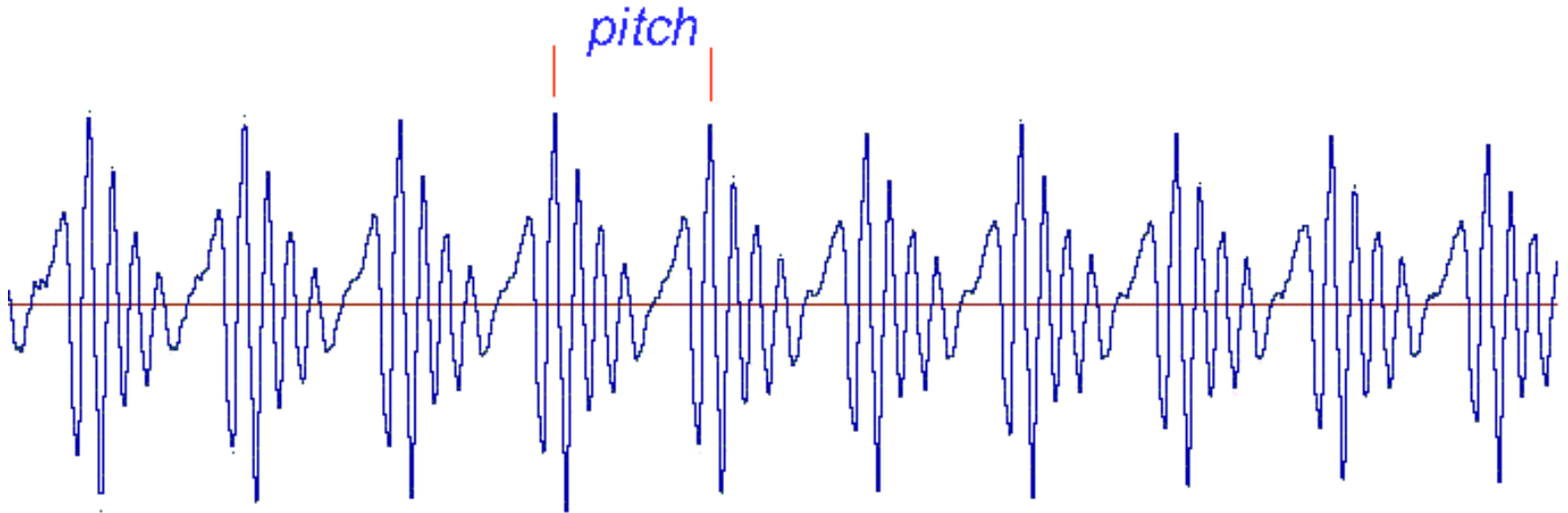
Unvoiced/voiced speech.

- *Voiced: Periodic, vowels and some consonants*
- *Unvoiced: Random, some consonants*

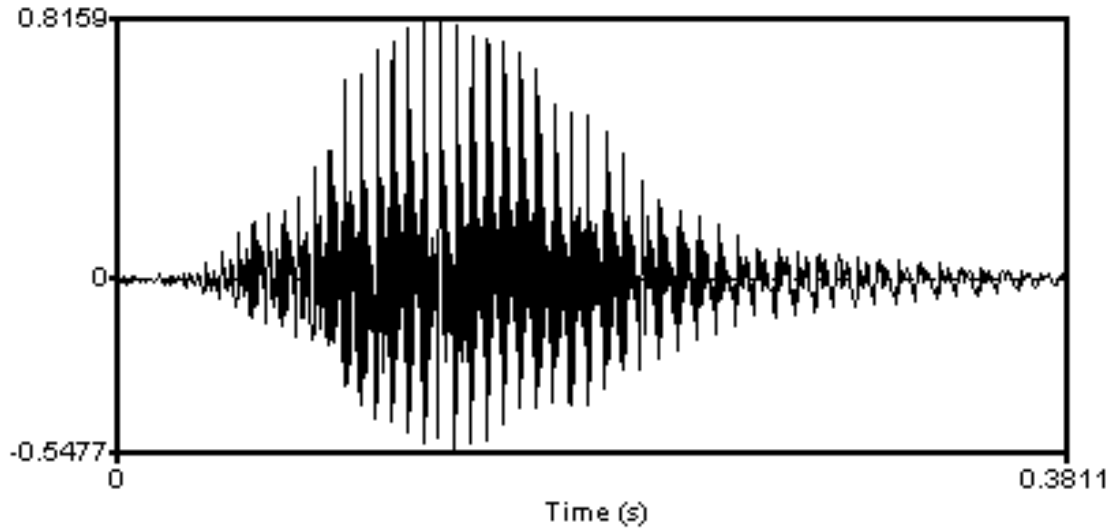


What is pitch?

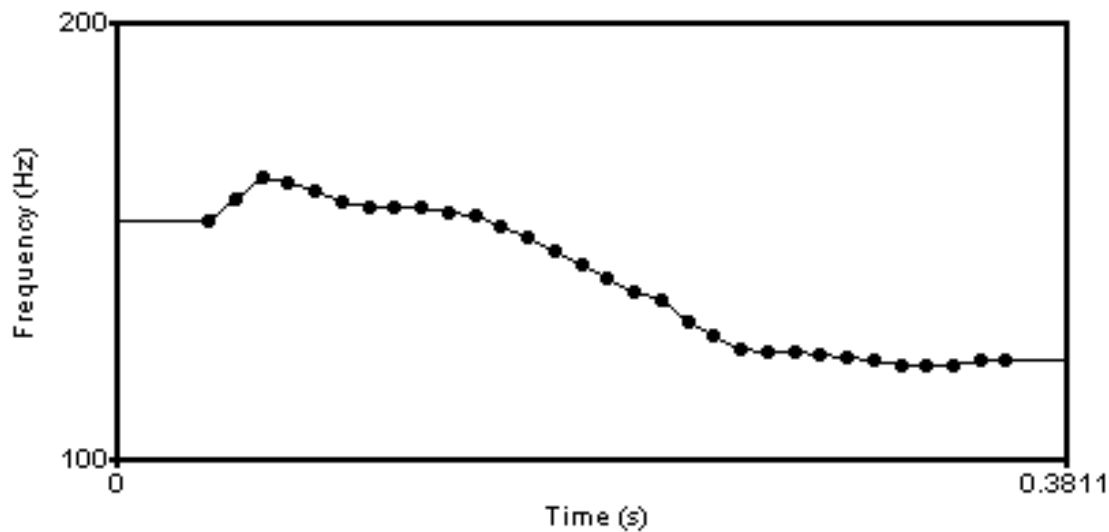
- *Pitch: (only applicable to voiced speech)*
 - Fundamental frequency (F0)
 - One period of speech data.



Pitch Contour



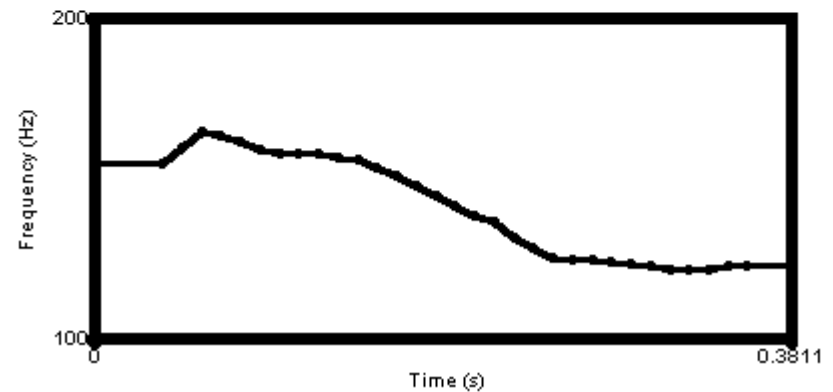
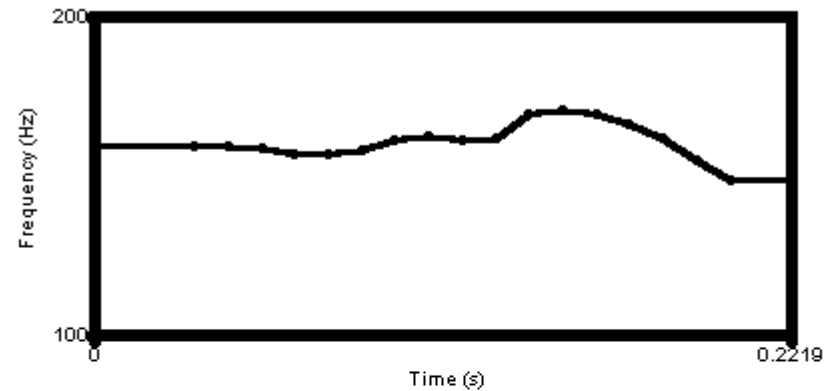
Wave Form



Pitch Contour

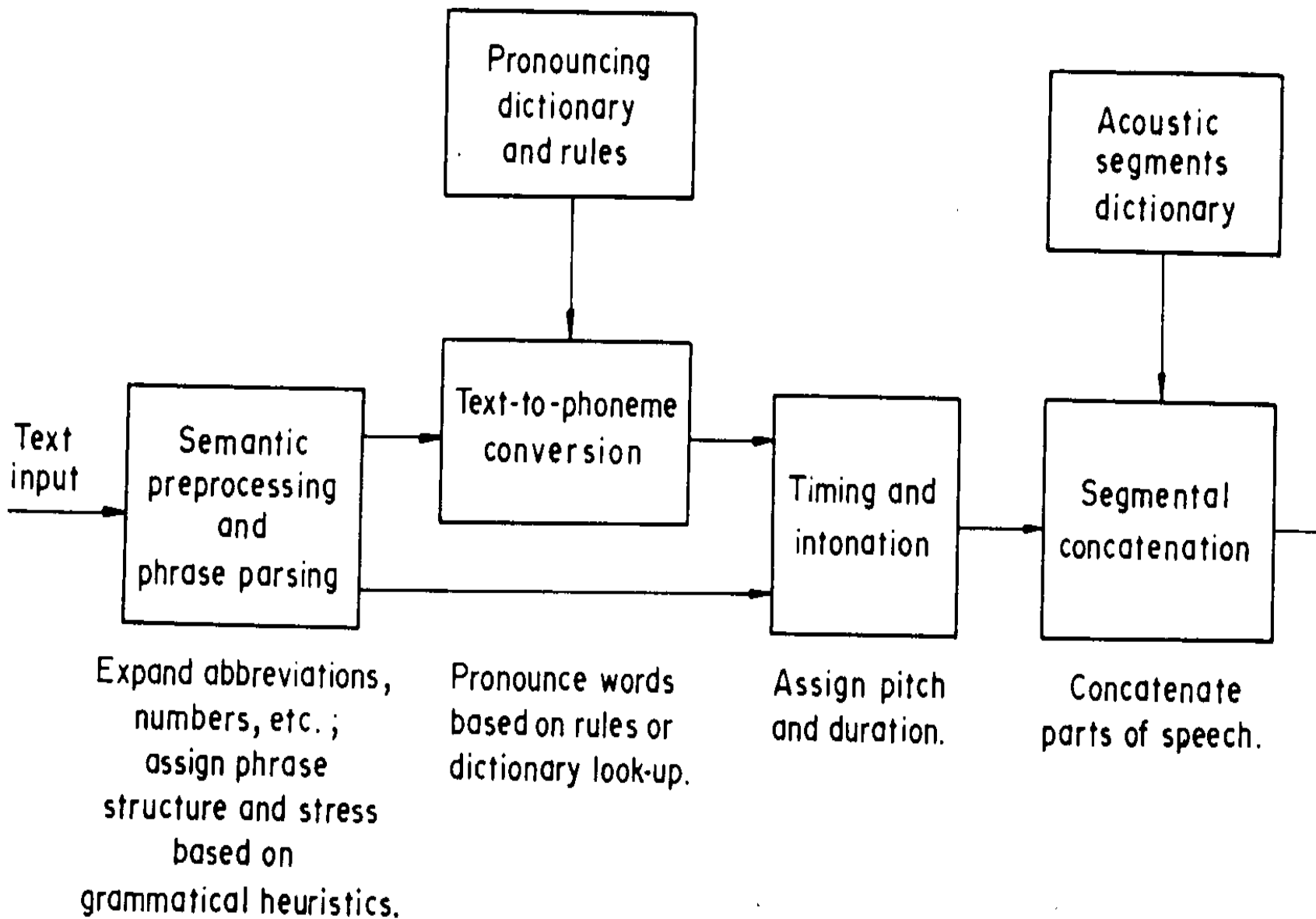
Pitch Contour

- **Example:**
 - Same syllable may have different pitch contour in different occasions



Goals and challenges of text to speech.

- **Generate speech from any given text.**
- **Goal: Generate natural speech, like human speech.**
 - Timber (Spectrum)
 - Prosody
 - **Linguistic Level:** Stress, Intonation, Rhythm, Tone...
 - **Acoustic level:** Pitch(F0), Duration(Timing), Amplitude(Energy, intensity) ...
- **Challenges:**
 - Text understanding, prosody generation, synthesis method.



Principal elements of text-to-speech conversion system

Synthesis by concatenating phonemes

- **Generate synthesizer's *control parameters* from a *phonetic transcription* of utterance.**
 - The utterance to be synthesized is represented by a *string of phonemes*
 - It is the input to *train the synthesizer*.
- ***Synthesized speech is constructed based on a set of rules:***
 - This is called *synthesis by rules*
 - a *look-up table* storing the parameters
 - data and rules for generating *transitions between neighboring sounds*
 - data and rules for *allophonic variations*
 - a way to assign *prosodic pattern*.

Concatenating larger units

- *Diphones* - units span 2 sounds, from the centre of one phone to another.
- *Other larger units for concatenation:*
 - syllable
 - demi-syllable
 - word
- *Syllable* - a syllable consists of an initial consonant C_i , followed by a vowel or diphthong V and the a final cluster C_f ie C_iVC_f
- *Syllable is not suitable, because of the strong co-articulation between adjacent syllables.*
 - The number of syllables is also too large, about 10,000 for English

•Demi-syllables

is more suitable.

- There are 800 initials and 1200 finals.
- Interpolation of parameters at demi-syllable boundaries is also simple as co-articulation there is weak.

•Word

the largest multi-phonemic unit in concatenation.

- Co-articulation between words are weak.
- **The problem is an extremely large number of words.**

The Naturalness - Prosodic Features

- ***Intonation and accent:***

- Are most important prosodic features.
- They relate to:
 - frequencies,
 - loudness,
 - duration.

- ***Basic intonation component:***

- in between pauses
 - (speech uttered in one breath),
- **pitch frequency** is usually high at the onset
- gradually decreases towards the end
 - to the decrease in **sub-glottal pressure**

- ***The **accent components** of the pitch pattern are determined by the **accent position** for each word or syllable.***

- ***In the next slide, we will cover two approaches of **speech synthesis by concatenation*****

Linear Predictive Synthesizers

- The actual signal can be reconstructed if the error function $e[n]$ is known

- We can model the error function as a period unit sample generator :
 - at a pitch frequency in the case of a **voiced speech**
 - or a random number generator in the case of **unvoiced speech**.

$$\tilde{x}[n] = \sum_{k=1}^p a_k x[n-k] \quad p = \text{order of LPC}$$

Error function :

$$x[n] = e[n] + \tilde{x} = e[n] + \sum_{k=1}^p a_k x[n-k]$$

- The synthetic speech will be give as

- A time-varying **set of control parameters** are required

- They give:

- the pitch-period,
- voiced/unvoiced decision,
- **G**,
- predictor coefficients.

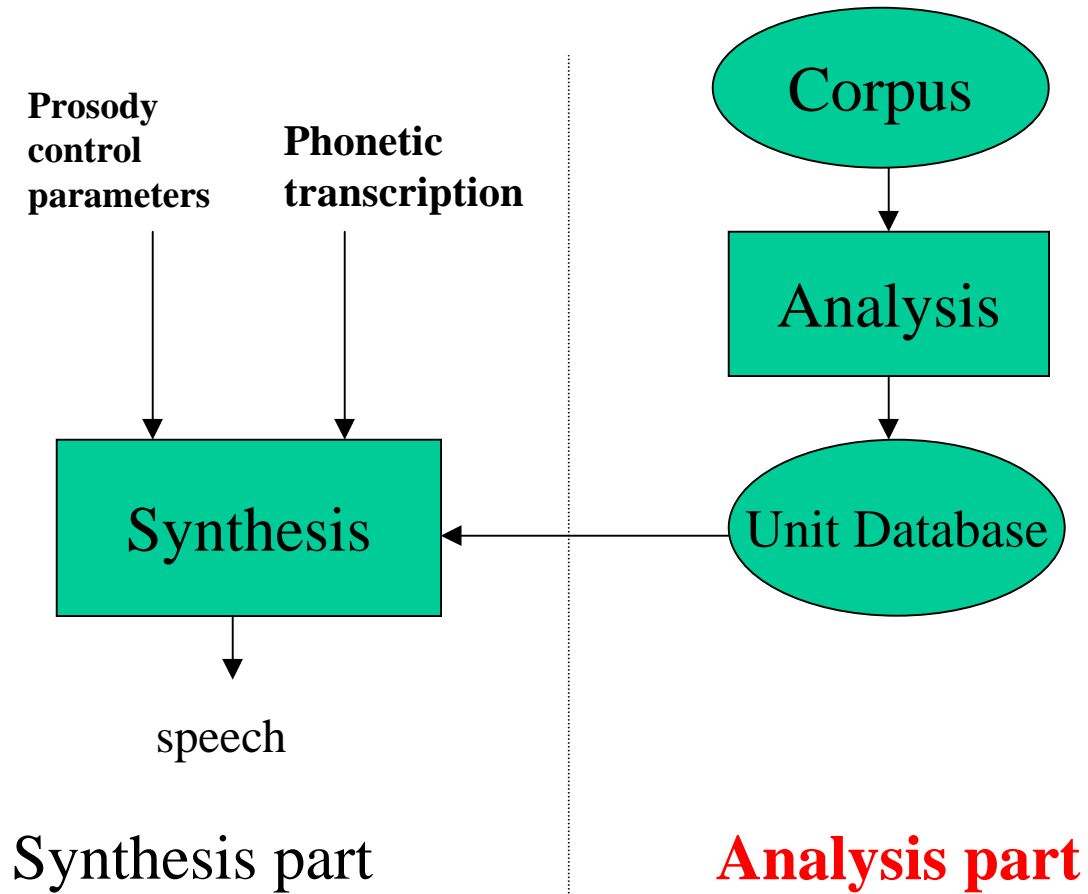
$$\tilde{x}[n] = \sum_{k=1}^p a_k x[n-k] + G \cdot u[n]$$

$u[n]$ = unit - step sequence and G is a gain control

PSOLA

- *Pitch Synchronous OverLap-Add*
- *A very popular method to synthesize speech.*
- *Proposed at the end of 1980's.*

Frame of PSOLA synthesis



Pitch-Synchronous-Overlap-Add Scheme (PSOLA)

- Provides a method to modify the **pitch** and **duration** of a speech segment in time domain,
- Makes it possible to modify the **prosody** in word or in sentence when synthesizing speech using waveform concatenation technique.
- The waveform concatenation is done on the **consonant parts**.

- For parameter synthesis, the main method is based on LPC, including
 - Single-pulse excitation LPC,
 - regular-pulse excitation LPC and
 - multi-pulse excitation LPC.
- It is easy to adjust the parameters and control synthesizer for high synthesized speech quality **by rules**.
- It needs **less resource** than waveform synthesis.

LPC = Linear Predictive Coding

Generating synthesized speech – Pitch-Synchronous-Overlap-Add Scheme

- ***In LPC approach, the speech is re-constructed from the source speech directly.***
 - There is little we can do on the individual wave data. It works on units-by-unit.
 - In PSOLA approach, the synthesized speech is generated by **modifying** the sound speech, **wave-by-wave**.
 - PSOLA thus allows **finer control** on the synthesized speech.
- ***PSOLA provides a method to **modify the pitch and duration** of a speech segment **in time domain or frequency domain**.***
 - Current work is on the TD-PSOLA, Time Domain PSOLA.
- ***this makes it possible to **modify the prosody** in word or in sentence when synthesizing speech **using waveform concatenation** technique.***
- ***The waveform concatenation is done **on the consonant** parts.***
 - This is because there is no distinct frequency characteristics in the consonant parts (these are normally unvoiced sound).

PSOLA - What do we need?

The following need be done

- **1. choose the basic unit of synthesis.**
- **2. record speech.**
- **3. build a speech feature database for PSOLA:**
 - this is a speech database with the **pitch-synchronous** mark.
 - for LPC, a speech feature database by **LCP analysis**, including **LPC coefficients**, **pitch**, **gain** and **excitation pulse**,
 - and using **Vector Quantization** if necessary.
- **4. Write a program for the synthesis model.**
- **5. Build a synthesis rule dictionary, including**
 - tone modification rule
 - stress rule
 - **light-tone rule**
 - energy rule
 - **er-colored final rule**
 - prosodic rule
 - duration rule
 - stop rule
 - intonation model

Advantages of PSOLA

- Use *pre-stored real speech* as synthesis units:

- keep speech *natural*

- **Use pre-stored real speech as synthesis units:**

- **Generate most natural speech as the original speech waves are kept and used.**

- **In LPC approach, the original waves are not kept.**

- **Instead, a set of LPC coefficients are used in synthesis.**

- *Synthesis by analysis:*

- Analyzing speech to create synthesis unit database.

- *Low computation cost*

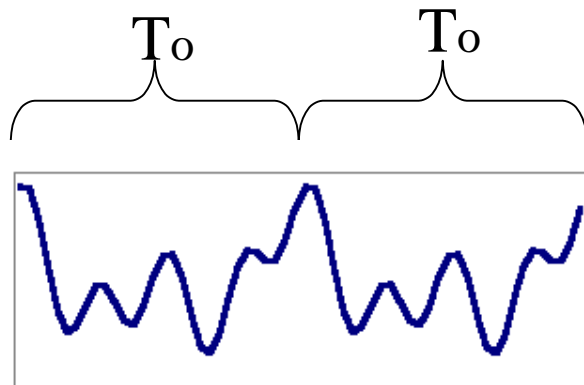
Advantages of PSOLA

- *Pitch level operations provide finer control on the synthesized speech*
 - Easy to change pitch period.
 - Easy to increase and decrease duration of speech.
 - Small synthesis unit database.

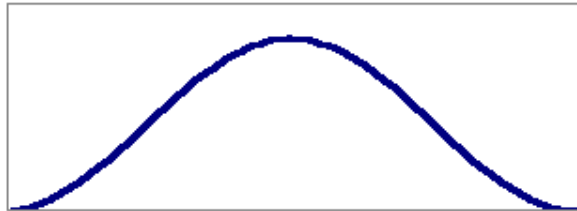
PSOLA Synthesis (1)

- *Input:*
 - Length of each part of speech
 - Pitch variation over time
- *Unvoiced part:*
 - Copy, no pitch change need.
- *Voice part:*
 - Extend a pitch two periods.
 - Multiply by a window function
 - Overlap and add.

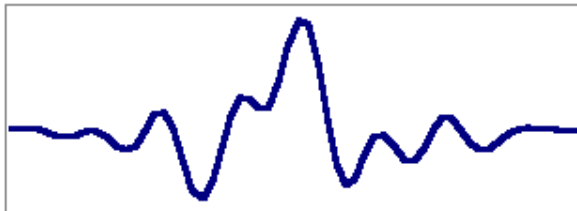
PSOLA Synthesis(2)



Two periods of a pitch
(T_0 : Pitch length)

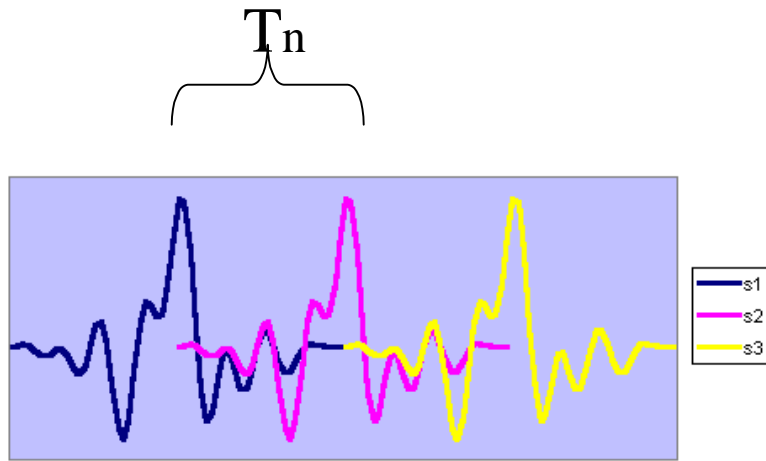


Window function

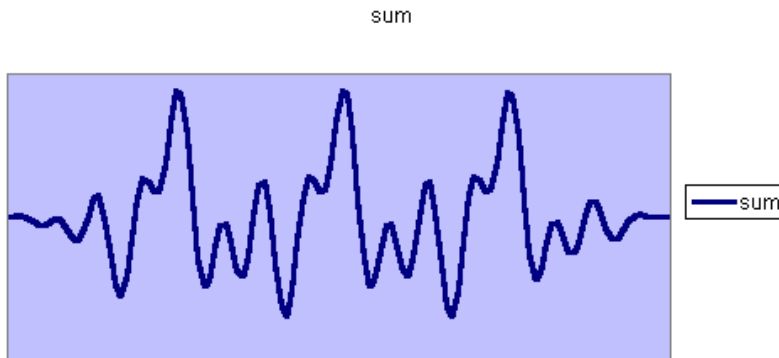


Multiplied result
(windowed signal)

PSOLA Synthesis(3)



Overlap windowed signals
(T_n : New pitch duration)



Result of addition
(synthesized speech)

PSOLA Synthesis(4)

- *Voice part: Modification*

- How to change pitch contour.

- Change the offset when overlapping.

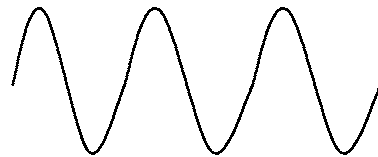
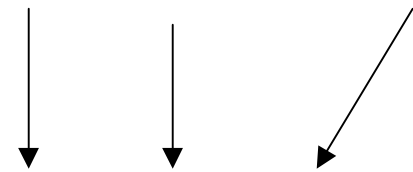
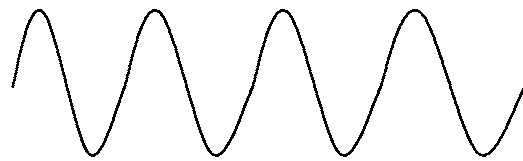
- How to change length of speech.

- Insert or delete pitches(change number of pitches).

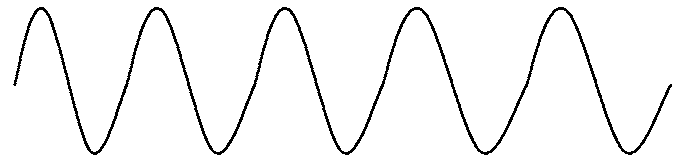
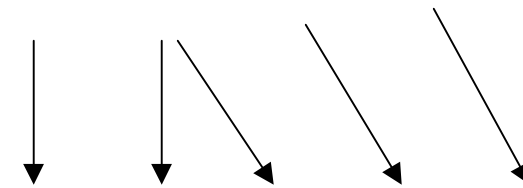
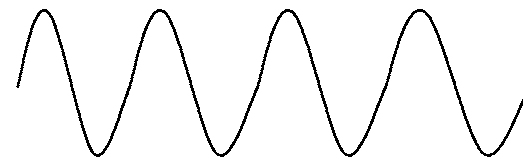
- How to change energy.

- Multiply a factor to change amplitude.

PSOLA synthesis(5)



Delete pitch to reduce duration



Insert pitch to increase duration

Synthesized Speech Examples:

• **1: Synthetic**

Human

• **2: Synthetic**



• **3: Synthetic**



Human

• **4: Synthetic**



• **5: Synthetic**



Speech Synthesis Links

- [The Pattern Playback \(Haskins Laboratories\)](#)
- [Current speech synthesis from Haskins](#)
- [Articulatory Synthesis at Haskins](#)
- [SineWave Synthesis at Haskins](#)
- [Bell Laboratories \(Lucent Technologies\)](#)
- [YorkTalk](#)
- [Speech Synthesis Museum](#) (speech links from the University of Birmingham)

Sources

***1. Associate Professor Lua Kim Teng,
School of Computing, NATIONAL
University of Singapore***

2. Jean Anderson

***3. Stephen Woodruff, University of
Glasgow, Language Centre***