A PRACTICE SCHOOL PROGRAM

REPORT ON

BLIND PEOPLE MESSAGE READER

K L UNIVERSITY

GreenFields, Vaddeswaram, Guntur District-520 022

2012-2013

By

SRI KANTH VAKA

(09101056)

BACHELOR OF TECHNOLOGY

In

ELECTRONNICS AND COMMUNICATION ENGINEERING

At

VYAS INFORMATICS Private Ltd.

(Nellore, Andhra Pradesh, India)

A PRACTICE SCHOOL PROGRAM
REPORT ON

BLIND PEOPLE MESSAGE READER

A project report submitted in partial fulfillment of the requirements for the award of the degree of
BACHELOR OF TECHNOLOGY
In
ELECTRONICS AND COMMUNICATION ENGINEERING
K L UNIVERSITY

Mr.M.V.D.PRASAD
Asst. Professor
K L University
(Technical Guide)

Mr. V H K Kishore
Design Engineer
Vyas Informatics, Ltd.
(Company Guide)

Mr. Raghavendra
Asst. Professor
K L University
(Operational Guide)
By
SRI KANTH VAKA
(09101056)
Department of Electronics and Communication Engineering
IV/IV B. Tech

A PRACTICE SCHOOL PROGRAM (2012-2013)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

K L UNIVERSITY

CERTIFICATE

This is to certify that this project work is being presented in this entitled: **"BLIND PEOPLE MESSAGE READER"** is bonafied by V.Sri Kanth (09101056) submitted in partial fulfillments of the requirements for the award degree of Bachelor of Technology and submitted to the Practice School Division, Department of Electronics and Communication Engineering at K L University in the academic year 2012-2013.

Mr.M.V.D.PRASAD
(Technical Guide)
Associate Professor.
K L University

Dr. HABIBULLA KHAN
Head of Department
E.C.E
K L University.

ACKNOWLEDGMENT

I have taken efforts in completing Practice School Program Successfully. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

I take this opportunity to express my profound gratitude and deep regards to **Dr. K. Rajasekhara Rao** (Dean Student Welfare) for providing me the greatest opportunity to have industrial exposure and to carryout live projects.

I take immense pleasure in thanking **Ram Kumar Reddy** (CEO, VYAS INFORMATICS Pvt. Ltd.) for having permitted me to carry out this project work. Also, I would wish to express my gratitude to **Prof. M. Bhaskar Rao** (Director, Practice school, K L University) for providing me an opportunity to do my Practice School Program in VYAS INFORMATICS private Ltd.

I would like to specially thank **Dr.HABIBULLA KHAN** (Head of the Department, CSE, K L University), **Mr. P. Venkateswara Rao** (Regional Coordinator, Practice School, Vijayawada) and Operational Guides **Mr. B Suresh, Mr. Raghavendra** for their help and support during the program.

I wish to express my deep sense of gratitude to our Technical Guide, **Mr.M.V.D.PRASAD**(Asst. Professor K L University) for his guidance and useful suggestions which helped us in completing the project work, in time.

It is my privilege to thank **Mr. V H K Kishore (**Company Guide**) and Mr. Karthik Reddy** (Lead Engineer) who were the source of inspiration and for their timely guidance in the conduct of our project.

Words are inadequate in offering my sincere thanks **to Mr.M.Sahithi** (Project Manager) and my team members for their encouragement, guidance and cooperation during our stay in the company.

Lastly I would also like to thank my parents for encouraging me all the way through my project.

TABLE OF CONTENTS

1. LIST OF FIGURES--7
2. EXECUTIVE SUMMARY---8
3. PROFILE OF THE INDUSTRY--------------------------------------9
4. PROFILE OF THE COMPANY------------ -----------------------11
5. OBJECTIVES---------------------------- ------------------------------11
6. CONTENT--12
7. INRODUCTION--29
8. RELATED WORK IN TEXT TO SPEECH------------------------34
9. SYNTHESIZER TECHNOLOGIES--------------------------------38
   - 9.1.CONCATENATIVE SYNTHESIS----------------------------39
     - ➢ 9.1.1 DIPHONE SYNTHESIS -------------------------------39
     - ➢ 9.1.2 UNIT SELECTION SYNTHESIS---------------------39
     - ➢ 9.1.3 DOMAIN SPECIFIC SYNTHESIS------------------40
   - 9.2 .FORMANT SYNTHESIS-------------------------------------40
   - 9.3.ARTICULATORY SYNTHESIS------------------------------41
   - 9.4 .HMM BASED SYNTHESIS-----------------------------------41
   - 9.5. SINEWAVE SYNTHESIS --------------------------------------41

10. WORKING OF TTS SYSTEM-------------------------------------41

11. TEXT ANALYSIS---46

- **11.1.AUTOMATIC PHOTOZENISATION----------------------47**
- **11.2.PROSODY GENERATION------------------------------------48**
- **11.3.RULE BASED SYNTHESIZERS------------------------------51**
- **11.4.CONCATINATIVE SYNTHESIZERS----------------------51**

12.CHALLENGES--57

- **12.1.TEXT NORMALISATION CHALLENGES---------------57**
- **12.2. TEXT TO PHENOME CHALLENGES--------------------58**
- **12.3.EVALUATION CHALLENGES------------------------------58**

13.DEDICATED HARDWARE---59

14.APPLICATIONS---62

15.RESULTS AND FINDINGS---65

- **15.1 CODE--66**

16. LIMITATIONS--73

17. DISCUSSIONS--73

18. CONCLUSION--74

19. BIBLIOGRAPHY---75

1.LIST OF FIGURES

EMBEDDED SOFTWARE VS HARDWARE	14
EMBEDDED MARKET	15
EMBEEDED SYSTEM	16
DTMF KEYPAD LAYOUT	21
DTMF STRUCTURE	24
FREQUENCIES USED FOR SIGNAL	25
DTMF BLOCK DIAGRAM	25
OVERVIEW OF TEXT PROCESSING	32
OVERVIEW OF TYPICAL TTS	32
SPEECH SYNTHESIZER	38
FUNCTIONAL DIAGRAM OF TTS	42
NLP MODULE OF TTS	45
INFORMATION PROVIDE BY INTONATION	49
CONCATENATION BASED SYNTHESIZER	52

2. EXECUTIVE SUMMARY

A Text-To-Speech (TTS) synthesizer is a computer-based system that should be able to read any text aloud, whether it was directly introduced in the computer by an operator .There is a fundamental difference between the system we are about to discuss here and any other talking machine (as a cassette-player for example) in the sense that we are interested in the automatic production of new sentences. This definition still needs some refinements. Systems that simply concatenate isolated words or parts of sentences, denoted as Voice Response Systems, are only applicable when a limited vocabulary is required (typically a few one hundreds of words), and when the sentences to be pronounced respect a very restricted structure, as is the case for the announcement of arrivals in train stations for instance. In the context of TTS synthesis, it is impossible (and luckily useless) to record and store all the words of the language. It is thus more suitable to define Text-To-Speech as the automatic production of speech, through a grapheme-to-phoneme transcription of the sentences to utter.

At first sight, this task does not look too hard to perform. After all, is not the human being potentially able to correctly pronounce an unknown sentence, even from his childhood? We all have, mainly unconsciously, a deep knowledge of the reading rules of our mother tongue. They were transmitted to us, in a simplified form, at primary school, and we improved them year after year. However, it would be a bold claim indeed to say that it is only a short step before the

computer is likely to equal the human being in that respect. Despite the present state of our knowledge and techniques and the progress recently accomplished in the fields of Signal Processing and Artificial Intelligence, we would have to express some reservations. As a matter of fact, the reading process draws from the furthest depths, often unthought-of, of the human intelligence

3. PROFILE OF THE INDUSTRY

Vyas Informatics has been established with the primary objective of providing unconventional engineering training and consulting services for the valued customers. We provide comprehensive Engineering training in many global delivery models such as online, onsite, offshore that answer all the training needs of individuals and organizations across the globe.

Vyas Informatics is the pioneer in the arena of professional engineering training and customized engineering consulting services with its strategically developed strong team of trainers, faculty, subject matter experts and good training infrastructure and supporting applications. The domains here include ELECTRONICS, VLSI, EMBEDDED and so on. All the electronic goods and components required to do a particular project are always available at all branches of VYAS INDIA.

Vyas Software Solutions is providing leading edge solutions for years to come. Backed by a strong inspirations and financial base, Vyas Software solutions continues to acquire expertise and invest in the aggressive research and development that fuels innovation. Online Training includes:

- **User Experience**

Our online training programs include online applications with effective user interface, detailed coverage of course curriculum, online interaction with trainers, interactive problem solving techniques and subject matter experts (SME).

- **Interactivity**

All of our online courses are both interactive and informative in nature. The advanced interactive techniques make the learning more effective.

- **Real-time Examples**

Our online training involves real-time examples from industry and corporate organizations to learn concepts and to know the application of the same for better understanding.

- **Technology Infrastructure**

Our online training programs are powered with advanced technology infrastructure such as online applications, dedicated web servers, exclusive database engines, interactive audio, animation and video streaming applications.

SWOT ANALYSIS OF THE COMPANY

STRENGTHS:

- Work friendly environment.
- Good Technical Guides.
- To train every student into industry ready professional.
- Expertise at partner level in consultancy

WEAKNESS

- Shortage of consultants at operating level rather than partner level
- Absence of important skills.
- Poor acess to distribution.

OPPURTUNITIES

- Increasing awareness.
- Attracting more clients.
- Well established position with a well-defined market niche.

THREATS

- Large consultancies operating at a minor level.
- Other small consultancies looking to invade the marketplace.
- Technological advances.
- New distribution markets.

4. PROFILE OF THE COMPANY

CEO	**Mr. Ram Kumar Reddy**

Lead Engineer	**Mr. Karthik Reddy**
Design Engineer	**Mr. V H K Kishore**
Project Trainer	**Mr. M.Sahithi**

5. OBJECTIVES

Right from day 1 of the practice school program (PSP), I took an active at VYAS INFORMATICS, NELLORE. Knowing that PSP gives me a very good work experience, I along with my team members utilized this opportunity very well. Some of the objectives that include are:

- Training on Embedded Systems must be done Sucessfully
- Specific tasks given by the company , that must be done by students
- Workshop on KEIL and ARDUINO with PROJECT DEMONSTRATION at JNIT College of Engineering.
- Real-time Project on Embedded Systems
- Published a Research paper on International Journals.
- Conducted Embedded-classes for Engineering Students on behalf of company.

6. CONTENT

1. **PROFILE OF THE EMBEDDED INDUSTRY**

Our day-to-day to life is becoming more dependent on Embedded system and Digital techniques. Embedded technologies are bonding into our daily activities even without our knowledge. For example refrigerator, washing machine, microwave oven, air conditioner, television sets, DVD Players, and music systems that we use in home are built around embedded systems. Evan in our vehicles the presence of specialized embedded systems vary from intelligent head lamps , engine controllers , ignition controllers , to complex air bag control system to protect you in case of severe accident. By forecasting the above examples I can say that **Embedded Industry** is playing and going to play a vital role in the coming future.

People will experience the power of Embedded Industry and enjoy the features and comfort provided by them. Most of us are totally unaware or ignorant of the intelligent embedded systems giving us so much comfort and security. They are always sitting in a hidden place and are dedicated to their assigned tasks.

1.1 EMBEDDED INDUSTRY Vs GENERAL COMPUTING INDUSTRY:

The computing revolution began with the general purpose computing requirements. Later it was realized that the general computing requirements are not sufficient for the embedded computing requirements. The Embedded computing requirements demand something special in terms of response to stimuli**,** meeting the computational deadlines, power efficiency, limited memory available etc.

A Personal Computer is a general purpose computer which has a very flexible application layer to allow for many different types of applications to be installed. Some examples of such software would range from entertainment, multimedia, to business & productivity software. PC's can have a mix of applications to suit a user's needs. Personal computers always have a user facing component as the applications they run are meant to be interfaced with by a person.

An embedded system is typically a less powerful computer which is very limited in its scope and purpose, usually designed and deployed for one application, and as such they only contain enough memory for one useful application. It is not uncommon for embedded systems to have no user interface, as many applications are never meant to accept user input or give human readable output. An example would be the types of computers that you would find embedded into a car. The computer controlling your anti-lock braking system interacts with various sensors and actuators to make and execute decisions without human interaction and is extremely specialized and miniaturized to do so.

1.2 HISTORY OF EMBEDDED INDUSTRY:

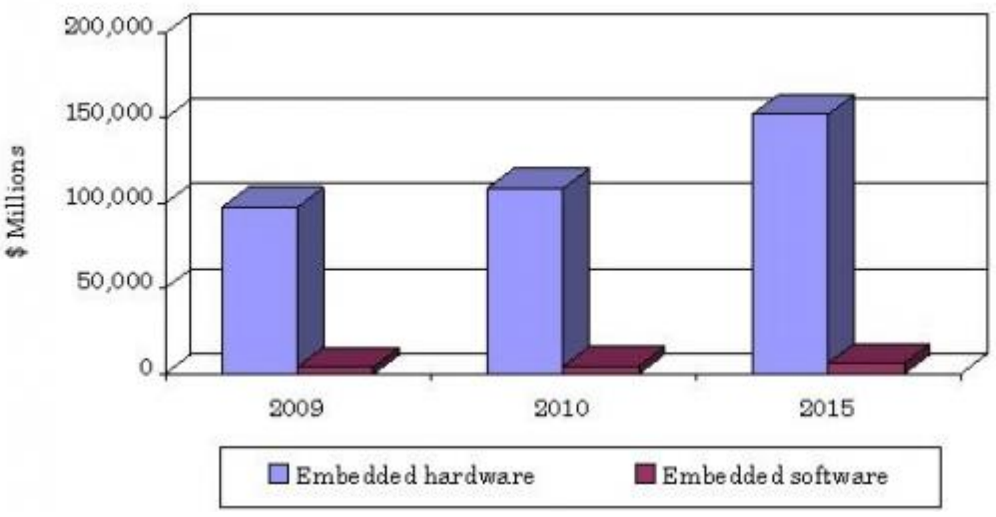
In the earliest years of computers in the 1930–40s, computers were sometimes dedicated to a single task, but were far too large and expensive for most kinds of tasks performed by embedded computers of today. Over time however, the concept of programmable controllers evolved from traditional electromechanical sequencers, via solid state devices, to the use of computer technology. One of the first recognizably modern embedded systems was the Apollo Guidance Computer, developed by Charles Stark Draper at the MIT Instrumentation Laboratory. At the project's inception, the Apollo guidance computer was considered the riskiest item in the Apollo project as it employed the then newly developed monolithic integrated circuits to reduce the size and weight. An early mass-produced embedded system was the Autonetics D-17 guidance computer for the Minuteman missile, released in 1961. It was built from transistor logic and had a hard disk for main memory. When the Minuteman II went into production in 1966, the D-17 2 was replaced with a new computer that was the first high-volume use of integrated circuits. This program alone reduced prices on quad nand gate ICs from $1000/each to $3/each, permitting their use in commercial products.

Since these early applications in the 1960s, embedded systems have come down in price and there has been a dramatic rise in processing power and functionality. The first microprocessor for example, the Intel 4004, was designed for calculators and other small systems but still required many external memory and support chips. In 1978 National Engineering Manufacturers Association released a "standard" for programmable microcontrollers, including almost any computer-based controllers, such as single board computers, numerical, and event-based controllers. As the cost of microprocessors and microcontrollers fell it became feasible to replace expensive knob-based analog components such as potentiometers and variable capacitors with up/down buttons or knobs read out by a microprocessor even in some consumer products. By the mid-1980s, most of the common previously external system components had been integrated into the same chip as the processor and this modern form of the microcontroller allowed an even more widespread use, which by the end of the decade were the norm rather than the exception for almost all electronics devices.

The integration of microcontrollers has further increased the applications for which embedded systems are used into areas where traditionally a computer would not have been considered. A general purpose and comparatively low-cost microcontroller may often be programmed to fulfill the same role as a large number of separate components. Although in this context an embedded system is usually more complex than a traditional solution, most of the complexity is contained within the microcontroller itself. Very few additional components may be needed and most of the design effort is in the software. The intangible nature of software makes it much easier to prototype and test new revisions compared with the design and construction of a new circuit not using an embedded processor.

1.3 EMBEDDED MARKET TRENDS:

The worldwide market for embedded technology was $101.6 billion in 2009, and was $113 billion in 2010. The market will exhibit steady growth at a compound annual growth rate (CAGR) of 7% over the next 5 years and BCC expects the market to reach $158.6 billion by 2015.



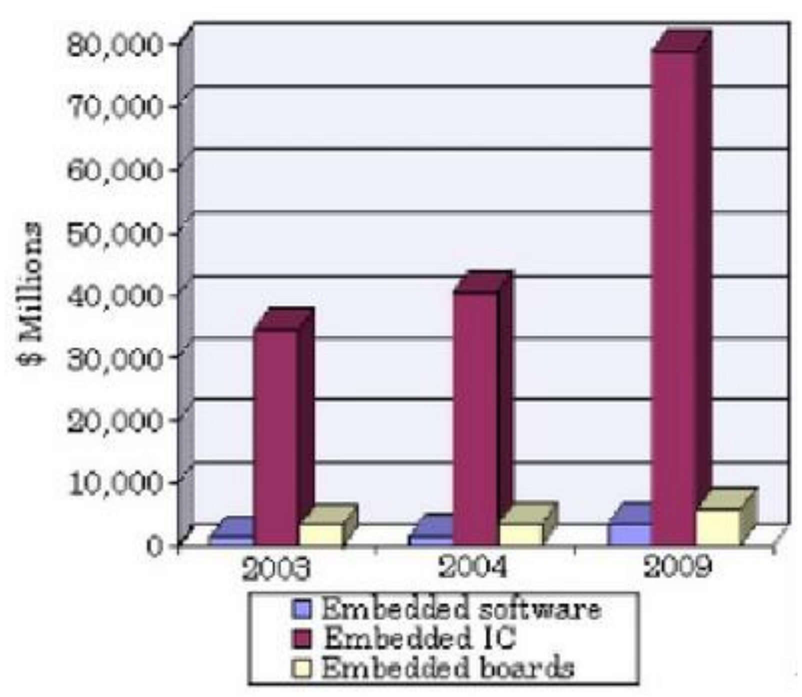
The embedded hardware segment represents the majority of the overall market with a $108.8 billion share in 2010. BCC expects this market to reach $152.4 billion by 2015 increasing at a CAGR of 7%.

The software segment accounts for $4.2 billion in revenue in 2010. This segment is expected to grow at a CAGR of 7.8% and will reach $6.1 billion by 2015.

1.4 EMBEDDED SOFTWARE & TOOLS MARKET:

The market highlights from ongoing coverage of the Embedded Software & Tools market include: The rising competition from open source platforms such as Android has created additional mainstream alternatives to commercial operating systems, with these alternate solutions available from all sides of the embedded industry; All industries cite outsourcing various tasks (e.g. software application development, middleware development, test) as part of their current project development; As the smart phone OS battle continues to intensify, independent device manufacturers such as HTC, LG, and Samsung have avoided committing their entire mobile roadmap to a single platform, instead opting for dual strategies centered around Android and Windows Phone 7; Mobile & Embedded Virtualization (MEV) has emerged as an alternative to address time-to-market pressures and cost-reduction requirements, in the face of mounting complexity now associated with many new embedded systems. Embedded Software & Tools Market research program will continue to analyze these (and other) emerging trends as well as provide data and insights that will help you to identify, define, prioritize and attack the best market opportunities.

Past years Embedded market:



1.5 MAJOR APPLICATION AREAS OF EMBEDDED INDUSTRY:

We are living in a world where embedded systems play a vital role in our day-to-day life, starting from home to the computer industry, where most of the people find their job for livelihood. Embedded technology has acquired a new dimension from its first generation model.

Consumer Electronics: Camcorders, Cameras, etc.

Household Appliances: Television, DVD players, Washing machine, fridge, microwave oven, etc.

Home Automation and Security systems: Air conditioners, sprinklers, intruder detection alarms, closed circuit television cameras, fire alarms, etc.

Automotive Industry: Anti-lock breaking system, engine control, Ignition systems, Automatic navigation systems, etc.

Telecom: Cellular telephones, telephone switches, Handset Multi-media applications, etc.

Computer Peripherals: Printers, scanners, Fax machines, etc.

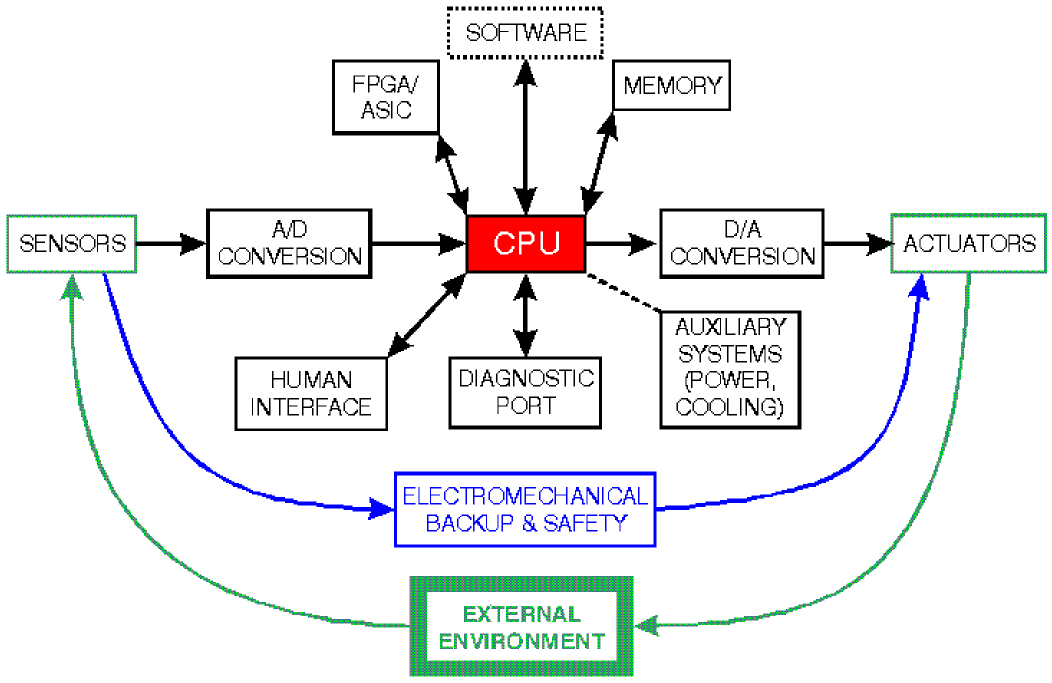
Computer Networking Systems: Network Routers, switches, hubs, Firewalls, etc.

Health care: Different kinds of scanners, EEG, ECG Machines, etc.

Measurements and Instrumentation: Digital Multi-meters, Digital CRO's, Logic analyzers, PLC systems, etc.
Banking and Retail: Automatic Teller Machines (ATM) and currency counters, Point of sales (POS).
Card Readers: Bar code, Smart card Readers Hand held devices, etc.

➢ WHAT IS EMBEDDED SYSTEM

An embedded system is a special-purpose computer system, which is completely encapsulated by the device it controls. An embedded system has specific requirements and performs pre-defined tasks, unlike a general-purpose personal computer. Embedded systems control many devices in common use today.



An embedded system is a programmed hardware device. A programmable hardware chip is the 'raw material' and it is programmed with particular applications. This is to be understood in comparison to older systems with full functional hardware or systems with general purpose hardware and externally loaded software. Embedded systems are a combination of hardware and software which facilitates mass production and variety of application.

A combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a dedicated function.

In some cases, embedded systems are part of a larger system or product, as in the case of an anti lock braking system in a car.

* EMBEDDED SYSTEM is a combination of Hardware and Software.

* An Embedded system is a system, that has a computing device embedded into it. * These are the controllers, processors, arrays or other hardware using dedicated (embedded) logic or programming (code) called “firmware” or a “microkernel.

* Embedded systems are designed around a µC which integrates Memory & Peripherals.

➢ **WHY EMBEDDED SYSTEM**

It is EMBEDDED because the Micro Controller is ‘inside’ some other system.For Example a Micro Controller is ‘EMBEDDED’ into your TV, car, or appliance.The consumer need not think about how to make it perform or process.

* Avoids lots of Electronics Components.

* Built in rich Features.

* Reduces the cost, space.

* Less Down Time for Maintenance.

* Probability of Failure is reduced.

* Easy interface with Computers.

➢ **CHARACTERISTICS OF AN EMBEDDED SYSTEM** • Sophisticated functionality • Real-Time Operation • Low Manufacturing Cost • Low Power Consumption • Eliminates Necessity of Complex Circuitry • Smarter Products

• Smaller Size • User Friendly • State of the Art Technology

➢ **TYPES OF EMBEDDED SYSTEMS**

1. General Computing • Applications similar to desktop computing, but in an embedded package. • Video games, set- top boxes, wearable computers, automatic tellers.

2. Control Systems • Closed- loop feedback control of real- time system • Vehicle engines, chemical processes, nuclear power, flight control 3.Signal Processing • Computations involving large data streams • Radar, Sonar, video compression 4.Communication & Networking • Switching and information transmission • Telephone system, Internet.

- **FEATURES OF AN EMBEDDED SYSTEM** 1.Real-Time Operation • Reactive: computations must occur in response to external events • Correctness is partially a function of time 2.Small Size, Low Weight • Hand- held electronics and Transportation applications. 3. Low Power • Battery power for 8+ hours (laptops often last only 2 hours)

4. Harsh environment

• Heat, vibration, shock, power fluctuations, RF interference, lightning.

5. Safety- critical operation

• Must function correctly and must not function in correctly.

- **Underwent EMBEDDED SYSTEMS training at the company.**

From the next day of the start of the PSP i.e. from 18th December, I along with my teammates, we were engaged in basics of EMBEDDED SYSTEMS training at the company. Ms. Sahithi, who is currently working as a trainee at VYAS taught us this course .She is a very good teacher who spends more time on giving us practical examples rather than explaining much theory.

Under her guidance, I have learned how to use and implement the EMBEDDED SYSTEMS codes using KEIL&ARDUINO software .We were trained up with much of the real time examples. We also learned how to write EMBEDDED SYSTEMS codes. In order to implement the codes, we need to use KEIL&ARDUINO software. Till then we don't know what is it and after we started using the software and implementing code in KEIL&ARDUINO, we felt the real work of a core engineer.

- **COMPLETED IEEE PROJECT BY NAME** DTMF:

DUAL TONE MULTI FREQUENCY SIGNALLING

ABSTRACT:

Dual-tone multi-frequency signaling (**DTMF**) is used for telecommunication signalling over analog telephone lines in the voice-frequency band between telephone handsets and other communications devices and the switching center. The version of DTMF that is used in push-button telephones for tone dialing is known as **Touch-Tone**. It was developed by Western Electric and first used by the Bell system in commerce, using that name as a registered trademark. DTMF is standardized by ITU-T Recommendation. It is also known in the UK as *MF4*.

Other multi-frequency systems are used for internal signaling within the telephone network.

- **<u>A Teaching Session Conducted by us on ARDUINO SOFTWARE to the Company clients</u>**

In our ON JOB ENGINEERING TRAINING our company asked us to do a Teaching session on **“**HOW TO USE ARDUINO SOFTWARE**”** for the clients in the company and it is called as “KT”, a Knowledge Transfer”.

So a teaching Session was conducted by us on ARDUINO SOFTWARE which includes how to implement the codes, how to write the codes, different types of writing the code. These three concepts were fully explained by us by giving support for each and every one. So this Teaching session has been conducted by us in a week. So we explained the basics first and have explained the concepts with examples. And the feedback came by the Company Employees.

- **<u>Workshop Conducted on KEIL and AURDINO software</u>**

Our Company has sent us to a college called Jawaharlal Nehru Institute of Technology (JNIT), Hyderabad to train the final years of that college on EMBEDDED SYSTEMS. This Training session is being conducting by us very effectively and there is a very good response came from the students.

Actually the workshop is conducted to explain the concepts of KEIL and AURDINO software to the Final Year students of JNIT College. Later we need to show the demonstration of their Projects (IEEE) and need to explain each and every module in their project.

Based on the schedule assigned to us we have started taking classes for the students on EMBEDDED basics. Even the response of the students was very good on our classes. So we are giving better output based on the requirements given by the Company. And I teach them how to implement and use the KEIL and AURDINO software by giving some practical examples on every Day. During the lab sessions I am giving some simple tasks to students on the programming. Every day our task is to explain the concepts thoroughly to them and I am doing my task very efficiently with my team.

- **<u>To publish a paper based on IEEE papers</u>**

During our Practice school programme we need to publish a journal. So based on the tasks I have selected a Base paper on **“vlsi**” concept**.**

VLSI is thus a technology that can be harnessed for various applications especially now a days

- **ACHIEVEMENTS**

ACHIEVEMENTS INCLUDES:

- EMBEDDED SYSTEM and WIRELESS COMMUNICATION fundamentals Training that includes Basics, application.
- Obtain Knowledge on KEIL and ARDUINO software.
- Demos on KEIL and ARDUINO codes.
- Learnt how to write the codes and implementation.
- Train the students on EMBEDDED SYSTEMS and particular projects.

TASKS

- To give a Demo Classes on KEIL and ARDUINO basics to all Employees in our Company.
- Train and Deal the concepts of Projects to Final year students .

DUAL TONE MULTI FREQUENCY SIGNALLING

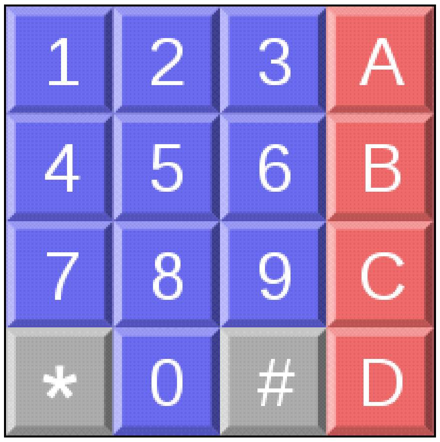
ABSTRACT:

Dual-tone multi-frequency signaling (**DTMF**) is used for telecommunication signalling over analog telephone lines in the voice-frequency band between telephone handsets and other communications devices and the switching center. The version of DTMF that is used in push-button telephones for tone dialing is known as **Touch-Tone**. It was developed by Western Electric and first used by the Bell system in commerce, using that name as a registered trademark. DTMF is standardized by ITU-T Recommendation. It is also known in the UK as *MF4*.

Other multi-frequency systems are used for internal signaling within the telephone network.

EXISTED SYSTEM:

Introduced by AT&T in 1963, the Touch-Tone system using the telephone keypad gradually replaced the use of rotary dial and has become the industry standard for landline service.



DTMF KEYPAD LAYOUT

Multi-frequency signaling is a group of signalling methods that use a mixture of two pure tone (pure sine wave) sounds. Various MF signalling protocols were devised by the Bell System and CCITT. The earliest of these were for in-band signalling between switching centres, where long-distance telephone operators used a 16-digit keypad to input the next portion of the destination telephone number in order to contact the next downstream long-distance telephone operator. This semi-automated signalling and switching proved successful in both speed and cost effectiveness. Based on this prior success with using MF by specialists to establish long-distance telephone calls, *Dual-tone multi-frequency* (DTMF) signalling was developed for the consumer to signal their own telephone-call's destination telephone number instead of talking to a telephone operator.

The DTMF system uses eight different frequency signals transmitted in pairs to represent 16 different numbers, symbols and letters. The engineers had envisioned phones being used to access computers, and surveyed a number of companies to see what they would need for this role. This led to the addition of the number sign (#, sometimes called 'octothorpe,' 'pound' or 'diamond' in this context - 'hash' or 'gate' in the UK) and asterisk or "star" (*) keys as well as a group of keys for menu selection: A, B, C and D. In the end, the lettered keys were dropped from most phones, and it was many years before these keys became widely used for vertical service codes such as *67 in the United States of America and Canada to suppress caller ID.
They were used before dialling the phone in order to give some calls priority, cutting in over existing calls if need be. The idea was to allow important traffic to get through every time. The levels of priority available were Flash Override (A), Flash (B), Immediate (C), and Priority (D), with Flash Override being the highest priority. Pressing one of these keys gave your call priority, overriding other conversations on the network. Pressing C, Immediate, before dialling would make the switch first look for any free lines, and if all lines were in use, it would disconnect any non-priority calls, and then any priority calls. Flash Override will kick every other call off the trunks between the origin and destination. Consequently, it was limited to the White House Communications Agency.

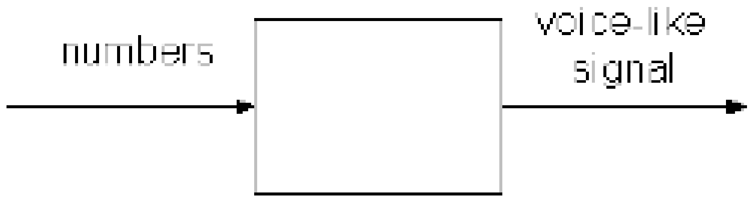
ARDUINO PROGRAM FOR DTMF:

```
#define IRQ 2
#define d0 3
#define d1 4
#define d2 5
#define d3 6
byte k;
void setup()
{
Serial.begin(9600);
pinMode(IRQ, INPUT);
pinMode(d0, INPUT);
pinMode(d1, INPUT);
pinMode(d2, INPUT);
pinMode(d3, INPUT);
}
void loop()
{
k = digitalRead(IRQ);
if ( k == 1 ){
Serial.print("Key : ");
char key = read_cod();
Serial.println(key);
delay(80);
}
}
byte read_cod()
{
byte data;
char key;
byte D0,D1,D2,D3;
D0 = digitalRead(d0);
D1 = digitalRead(d1);
D2 = digitalRead(d2);
D3 = digitalRead(d3);
bitWrite(data,0,D0);
bitWrite(data,1,D1);
bitWrite(data,2,D2);
bitWrite(data,3,D3);
if ( data == 1 ) key = '1';
if ( data == 2 ) key = '2';
if ( data == 3 ) key = '3';
if ( data == 4 ) key = '4';
if ( data == 5 ) key = '5';
if ( data == 6 ) key = '6';
if ( data == 7 ) key = '7';
if ( data == 8 ) key = '8';
if ( data == 9 ) key = '9';
if ( data == 10 ) key = '0';
if ( data == 11 ) key = '*';
if ( data == 12 ) key = '#';
if ( data == 13 ) key = 'A';
if ( data == 14 ) key = 'B';
if ( data == 15 ) key = 'C';
if ( data == 16 ) key = 'D';
return key;
}
```

DETAILED EXPLANATION:

The telephone network is designed to carry voice signals. Nonetheless, it is frequently asked to carry other types of signals. A simple and ubiquitous example is telephone numbers. Your telephone has to communicate with the phone company central office the phone number you are intending to call. It has to do that over circuits designed to carry voice signals. Moreover, you may connect to a long-distance carrier distinct from your local service provider before supplying the phone number you want to call. Or you may connect to some service that asks you to enter your credit card number or account number, or asks you to respond to certain questions by pressing buttons on your telephone keypad.

A system is needed that has the following structure:



It converts sequences of numerical digits into signals that will easily traverse circuits designed for voice. The internationally standardized way to do that is *DTMF signaling*, or dual-tone, multi-frequency. DTMF signaling converts decimal digits (and the symbols '*' and '#') into sounds that share enough essential characteristics with voice to easily traverse circuits designed for voice.

Here is an illustration of DTMF. Enter a phone number in the entry box, and then type return.

The sound you hear is the DTMF signal for the number you entered. The number can include any of the digits {0,1, ... ,9} plus the symbols '*' and '#'. The above applet also understands the special symbol ',' (a comma), which will produce a pause of approximately one second. All other symbols are ignored. The DTMF coder is thus a function that maps a phone number into a voice-like signal. Let *Digits* = {0, 1, ..., 9, '*', '#'} be the set of digits that a telephone keypad can produce. Let *Indices* = {1, ..., *N*}. An *N* digit phone number, therefore, is a function

PhoneNumber: *Indices* → *Digits*.

Of course, not all sequences of digits are valid phone numbers, so the set of all valid *N* digit phone numbers is a subset of the set of *N*-digit sequences,

PhoneNumbers ⊂ [*Indices* → *Digits*].

Recall that sound is

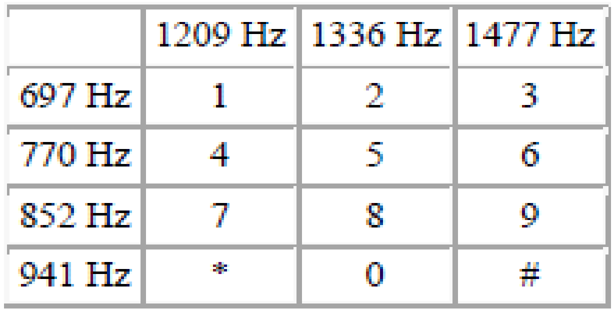
Sound : *Time* → *Pressure*.

The set of all sounds is the function space

Sounds = [*Time* → *Pressure*].

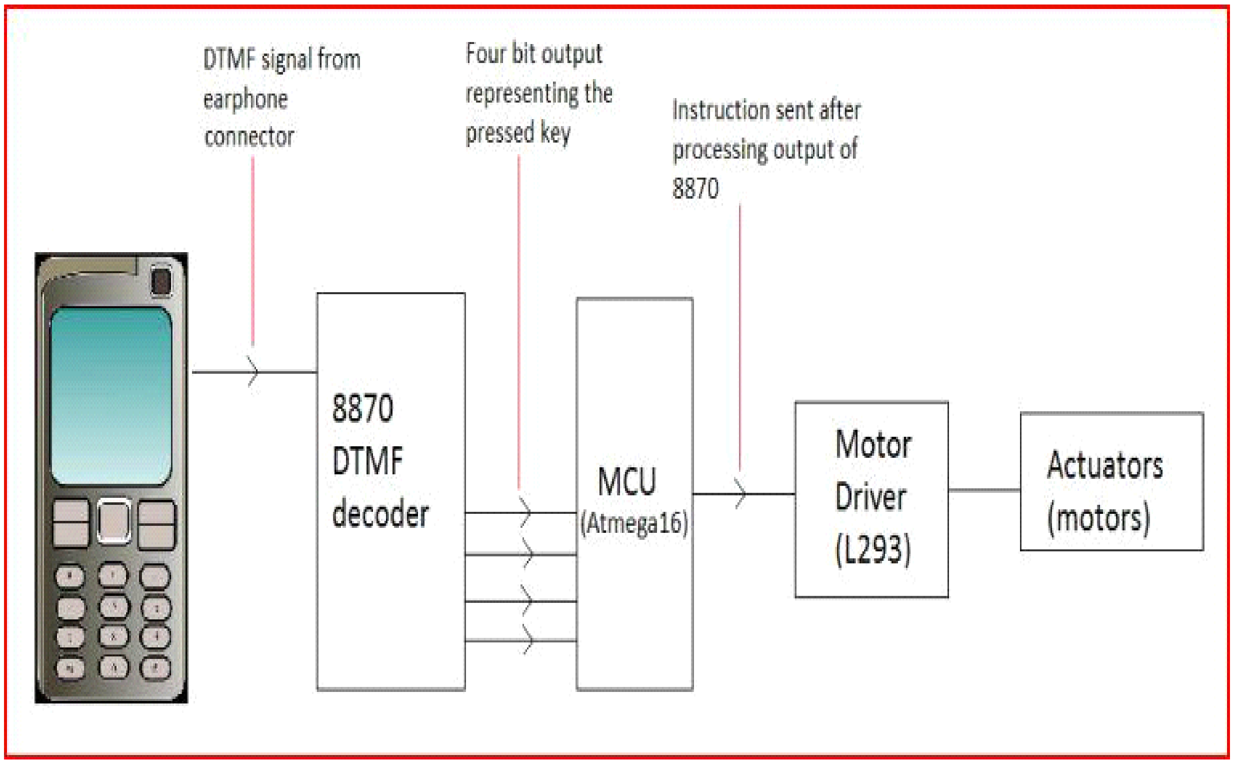
Thus, the above system is a function that maps a function *PhoneNumber* into a function *Sound*. I.e., the DTMF signaling system is a function

DTMF : *PhoneNumbers* → *Sounds*.
There are twelve DTMF signals, each of which is made up of two tones from the following selection: 697 Hz, 770 Hz, 852 Hz, 941 Hz, 1209 Hz, 1336 Hz, 1477Hz.
The tones are divided into two groups (low and high), and each DTMF signal uses one from each group. This prevents any harmonics from being misinterpreted as part of the signal.
The following table shows the frequencies used for each signal



	1209 Hz	1336 Hz	1477 Hz
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

PROPOSED SYSTEM:
Now, this DTMF signalling is used for guiding robots. The block diagram is as follows:


DTMF signal from earphone connector
Four bit output representing the pressed key
Instruction sent after processing output of 8870
8870 DTMF decoder
MCU (Atmega16)
Motor Driver (L293)
Actuators (motors)

CODE FOR MOBILE ROBO :

```
const int in1 = 2;
const int in2 = 4;
const int in3 = 7;
const int in4 = 12;
const int data0 = 14;
const int data1 = 15;
const int data2 = 16;
const int data3 = 17;
void front(void)
{
digitalWrite(in1, HIGH); // Motor 1 o/p
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH); // Motor 2 o/p
digitalWrite(in4, LOW);
}
void back(void)
{
digitalWrite(in1, LOW); // Motor 1 o/p
digitalWrite(in2, HIGH);
digitalWrite(in3, LOW); // Motor 2 o/p
digitalWrite(in4, HIGH);
}
void left(void)
{
digitalWrite(in1, LOW); // Motor 1 o/p
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH); // Motor 2 o/p
digitalWrite(in4, LOW);
}
void right(void)
{
digitalWrite(in1, HIGH); // Motor 1 o/p
digitalWrite(in2, LOW);
digitalWrite(in3, LOW); // Motor 2 o/p
digitalWrite(in4, LOW);
}
void stopall(void)
{
```

```
digitalWrite(in1, LOW); // Motor 1 o/p
digitalWrite(in2, LOW);
digitalWrite(in3, LOW); // Motor 2 o/p
digitalWrite(in4, LOW); }
void setup()
{
Serial.begin(9600);// opens serial port, sets data rate to 9600 bps
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);
pinMode(data0, INPUT);
pinMode(data1, INPUT);
pinMode(data2, INPUT);
pinMode(data3, INPUT);
}
void loop()
{
if( LOW == digitalRead(data3) && LOW == digitalRead(data2) && HIGH ==
digitalRead(data1) && LOW == digitalRead(data0) ) // check for value 2
{
front();
}
else if( LOW == digitalRead(data3) && HIGH == digitalRead(data2) && LOW ==
digitalRead(data1) && LOW == digitalRead(data0) ) // check for value 4
{
left();
} else if( LOW == digitalRead(data3) && HIGH == digitalRead(data2) && HIGH ==
digitalRead(data1) && LOW == digitalRead(data0) ) // check for value 6
{
right();
}
else if( HIGH == digitalRead(data3) && LOW == digitalRead(data2) && LOW ==
digitalRead(data1) && LOW == digitalRead(data0) ) // check for value 8
{
back();
}
else if( LOW == digitalRead(data3) && HIGH == digitalRead(data2) && LOW ==
digitalRead(data1) && HIGH == digitalRead(data0) ) // check for value 5
{ stopall();
}
else
{
// do nothing
}
}
```

CONCLUSION:

The Mobile robo is an application of DTMF signalling. More bots can be developed involving DTMF controlling. The robot can be used for the military operation, spy robot

7.INTRODUCTION

Stephen hawking is one of the most famous people using speech synthesis to communicate

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a **speech synthesizer**, and can be implemented in software or hardware products. A **text-to-speech (TTS)** system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcription into speech.

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database Systems differ in the size of the stored speech units; a system that stores phones or diphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood. An intelligible text-to-speech program allows people with visual impairements or reading disabilities to listen to written works on a home computer. Many computer operating systems have included speech synthesizers since the early 1990s.

The intent of this review is to trace the history of progress toward he development of system for converting text to speech, giving credit along the way to those responsible for the important ideas that have led to present successes. Emphasis is placed on the theory behind current algorithms.
The account of this theory, in conjunction with an extensive Bibliography, can serve to bring someone new to the field "up to speed" fairly rapidly, even though to some extent Existing commercial systems are hidden behind a cloud of Proprietary trade secrets. Perceptual data that measure the Intelligibility of current systems are summarized, and a brief Attempt is made to estimate the potential of the technology for practical application, especially in areas of social need. A final purpose of this undertaking is to identify the weakest links in present systems for the conversion of unrestricted Text to fluent, intelligible, natural sounding speech. The hope is that this critical review will focus future research in directions having the greatest payoff . The reader should be aware that the author is not an impartial outside observer, but rather an active participant in the field who has many biases that will no doubt color the review. The steps involved in converting text to speech are illustrated in Fig. 1 (Allen, 1976). First, a set of modules must Analyze the text to determine the underlying structure of the Sentence and the phonemic composition of each word.
Then, a second set of modules transforms this abstract linguistic representation into a speech waveform. These processes have interesting connections to linguistic theory, models of speech production, and the acoustic-phonetic characterization of language (experimental phonetics),as well as to a topic that Vanderslice (1968) calls" synthetic elocution," or the art of effective reading out loud. The review will focus on the conversion of English text to speech. Systems for other languages will not be reviewed unless they Have contributed to the evolution of systems

for English. It might seem more practical to store natural waveforms Corresponding to each word of English, and to simply concatenate them to produce sentences, particularly considering the low cost and large capacity of new laser disk technology. However, such an approach is doomed to failure because a spoken sentence is very different from a sequence of words uttered in isolation. In a sentence, words are as short as half their duration when spoken in isolation—making concatenated speech seem painfully slow. The sentence stress pattern, rhythm, and intonation, which depend on syntactic and semantic factors, are disruptively unnatural when words are simply strung together in a concatenation scheme. Finally, words blend together at an articulatory level in ways that are important to their perceived naturalness and intelligibility. The only satisfactory way to simulate these effects is to go through an intermediate syntactic, phonological, and phonetic transformation. A second problem with approaches that attempt to store Representations for whole words is that the number of words That can be encountered in free text is extremely large, due in part to the existence of an unbounded set of proper names [e.g., the Social Security Administration (1985) estimates that there are over 1.7 million different surnames in their files],a s well as the existence of general rules that permit the formation of larger words by the addition of prefixes and suffixes to root words, or by compounding. Also, new words are being coined every day. It was hoped that a system employing prerecorded words might spell out such items for the listener, but this has proven to be less than satisfactory. Modern systems to be described below have fairly powerful fall-back procedures to be used when an unfamiliar word is encountered**.**

For expository reasons the review is organized backwards with respect to Fig.1.Only after we have some idea of the nature of the input information required by the synthesis routines will the second section take up the analysis of text**.**

Linguistic framework

A recent trend in linguistics has been to describe a language such as English in generative terms, the goal being to specify rules for the generation of any legitimate sentence of the language (Chomsky and Halle, 1968). I have summarized and simplified this view somewhat in Fig. 2 to indicate how it might be applied to the problem of synthesis. Linguists believe that a sentence can be represented by a sequence of discrete lements, called phonemes that are drawn from a small set of about 40 such sound building blocks for English(see Table IV). These abstract phonemic symbols might be thought to represent articulatory target configurations or gestures. Thus a word like "beam" consists of three phonemes, the /b/characterized by lip closure, the vowel /i/ characterized by a high fronted tongue position, and the nasal /m/characterized by both lip closure and opening of the velar port to the nasal passages. The psychological reality
of the phoneme as a unit for representing how words are to be spoken is attested to by collections of speech errors in which phonemic exchanges are common(Fromkin, 1971) . Linguists have also found it useful to be able to refer to the Components or features of a phoneme, such as the fact that /b/ and /m/ are both + LABIAL, while only /m/ is + NASAL. Rules describing how words change pronunciation in certain sentence contexts are often stated most efficiently in terms of features Phoneme strings form larger units such as syllables, words, phrases, and

clauses. These structure should be indicated in the underlying representation for an utterance, bemuse aspects of how a sentence is pronounced depend on the locations of these types of boundaries. Each syllable of a word in a sentence can be assigned a strength or stress level. Differences in assigned stress make some syllables stand out from the others. The stress pattern has an effect on the durations of sounds and on the pitch changes over an utterance (fundamental frequency of vocal cord vibrations, or f0)- The phonological component of the grammar converts phonemic Representations and information about stress and Boundary types into (1) a string of phonetic segments plus (2) a superimposepd attern of timing, intensity, and f0 motions-the latter three aspects being known as sentence prosody.

History of TTS:

Long before electronic signal processing was invented, there were those who tried to build machines to create human speech. Some early legends of the existence of "speaking heads" involved Gerbert of Aurillac (d. 1003 AD), Albertus Magnus (1198–1280), and Roger Bacon (1214–1294).

In 1779, the Danish scientist Christian Kratzenstein, working at the Russian Academy of Sciences, built models of the human vocal tract that could produce the five long vowel sounds (in International Phonetic Alphabet notation, they are [a], [e], [i], [o] and [u]. This was followed by the bellows-operated "acoustic-mechanical speech machine" by Wolfgang von Kempelen of Vienna, Austria, described in a 1791 paper. This machine added models of the tongue and lips, enabling it to produce consonants as well as vowels. In 1837, Charles Wheatstone produced a "speaking machine" based on von Kempelen's design, and in 1857, M. Faber built the "Euphonia". Wheatstone's design was resurrected in 1923 by Paget.

In the 1930s, Bell Labs developed the VOCODER, a keyboard-operated electronic speech analyzer and synthesizer that was said to be clearly intelligible. Homer Dudley refined this device into the VODER, which he exhibited at the 1939 New York World's Fair.

The Pattern playback was built by Dr. Franklin S. Cooper and his colleagues at Haskins Laboratories in the late 1940s and completed in 1950. There were several different 9 versions of this hardware device but only one currently survives. The machine converts pictures of the acoustic patterns of speech in the form of a spectrogram back into sound. Using this device, Alvin Liberman and colleagues were able to discover acoustic cues for the perception of phonetic segments (consonants and vowels).

Early electronic speech synthesizers sounded robotic and were often barely intelligible.The quality of synthesized speech has steadily improved, but output from contemporary speech synthesis systems is still clearly distinguishable from actual human speech.

Overview of text processing:

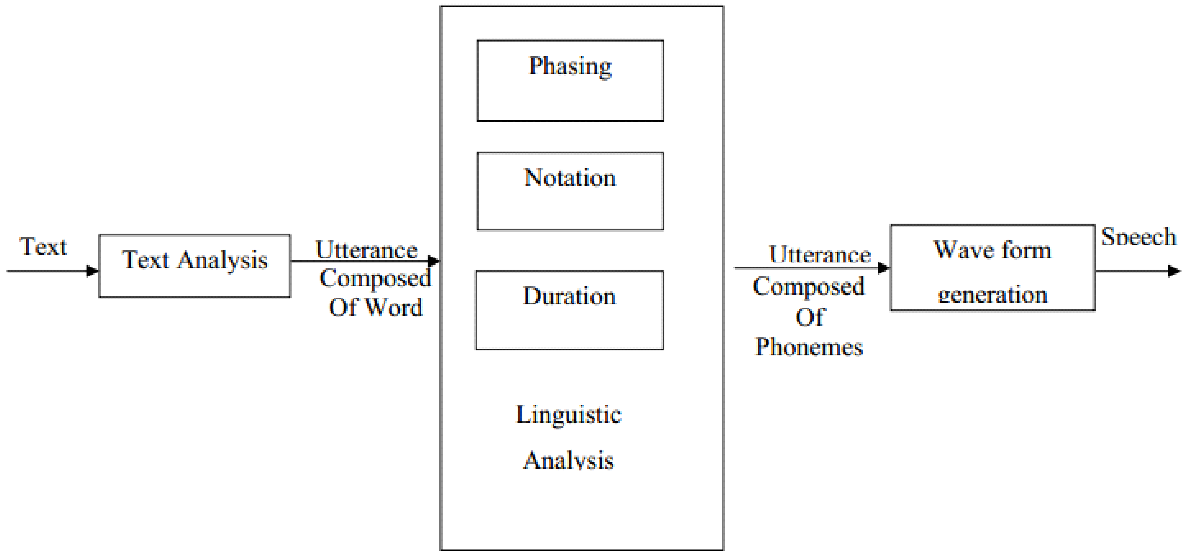


Fig 1.6: Text processing

Overview of a typical TTS system

	Automatic announcement Menu 0:00 A synthetic voice announcing an arriving train in Sweden.
	Sample of Microsoft Sam Menu 0:00

	Microsoft Windows XP's default speech synthesizer voice saying "The quick brown fox jumps over the lazy dog 1,234,567,890 times. soi"

A text-to-speech system (or "engine") is composed of two parts: a front-end and a backend. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, pre-processing, or tokenization. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end often referred to as the synthesizer, then converts the symbolic linguistic representation into sound.

8.RELATED WORK IN TEXT TO SPEECH

William A. Ainsworth 1973 proposed a system for converting English text to speech. He investigated the feasibility of converting English text into speech using an inexpensive computer and a small amount of stored data. He segmented text into breath groups, the orthography is converted into a phonemic representation, lexical stress is assigned to appropriate syllables, then the resulting string of symbols in converted by synthesis by rule in the parameter values for controlling an analogue speech synthesizer. The algorithms for performing these conversions are described in details and evaluated independently, and the intelligibility of the resulting synthetic speech in assed by listening tests. In his approach a typical seven word sentence contains less than one phonetic error. This level of performance is probably achieved because most of the longer words in English are pronounced according to rule, whereas the common words are pronounced irregularly. The present set of rules ensures that the most common irregular words are treated as special cases, and the correct phonemic translation is generated.Fushikida, Katsunobu et.al 1982 developed a low cost, compact text to speech synthesizer using formant speech synthesizer LSI for the personal computer. This speech synthesis system based on synthesis by rule using glottal pole formant model and CV, VC speech segment compilation technique. It can synthesize any arbitrary Japanese speech from the input text. They obtained 99% word intelligibility in the listening tests. It should be possible to apply this system for other languages, by modifying the synthesis program and parameters .

Susan R. Hertz 1986 proposed English text to speech conversion with delta. Delta is a computer system for developing text to speech rules for any language. He described a set of delta rules for English, on the basis of the input text. The English rules build multilevel linguistic constituents of the utterance (e.g. morphs, syllables, and phonemes) and the low-level synthesizer parameter patterns. In his proposed rules syllable and phoneme tokens are generated and used them to derive target and transitioned patterns for a formant synthesizer, another rule set might take a different approach, perhaps generating diphone tokens, and extracting LPC coefficients for them from the dictionary. Another rule set might use a formant synthesizer, but generate the formant values on the basis of articulatory streams. The possibilities are endless.

Cecil H. Coker et.al 1990 proposed two powerful alternatives to letter to sound rules for speech synthesis which are Morphology and Rhyming. Most speech synthesizers have tended to depend on letter-to-sound rules for most words, and resort to a small ‘‘exceptions Dictionary’’ of about 5000 words to cover the more serious gaps in the letter to-sound rules. The Bell Laboratories Text-to-Speech system which is moving to an extreme dictionary-based approach cuts the error rate by at least an order of magnitude. The pronunciation problem has traditionally been divided into two very separate modules: letter-to-sound rules and the exceptions dictionary. They focused on letter-to sound rules which work from first principles. In contrast, they resort to letter-to-sound rules only when all alternatives have been exhausted. The most reliable inference is table lookup. Failing that, the system tries to make as safe an inference as possible from similar words in the dictionary. Stress neutral morphology is considered fairly safe but rhyming is more dangerous, but far more this approach has a much smaller error rate than previous letter-to-sound systems.

Mark Tatham and Eric Lewis 1996 proposed that naturalness in synthetic speech is essentially the successful rendering of variability in the final acoustic signal, once they got the obvious factors such as limiting the domain of discourse within which the system is to operate. The obvious factors of rhythm and intonation there is the more difficult question of modeling the variability in human speech. They discussed how SPRUCE, a high-level text- to-speech synthesis system, incorporates several different types of variability. In SPRUCE we identify and treat distinctly several sources of variability in human speech, adhering carefully to contemporary speech production theory. They believed that this approach renders transparent the sources of naturalness, and at the same time enables us to manipulate what they felt to be an important interplay between the various types of variability .

Chen Fang and Yuan Baozong 1996 developed the intelligent speech production system with text generation intelligent speech production system is different from ordinary speech production systems. They considered not only how to convert text into speech but also how to generate the necessary text in text-to- speech conversion. The system gets the right test by the step of topic selection, test planning. test organization, grammar realization and text generation According to the generated test the system at last generates speech which have good quality in naturalness and intelligibility using Chinese Text-to-Speech Conversion System. The paper introduced the structure and main techniques of intelligent speech production system with text generation. It has wide application prospect in intelligent demonstration system, expert system, inquiring system, auto report generation system, and translation system to raise the intelligent level in man machine interaction [14].Nobuyuki Katae and Shinta Kimara 1996 developed natural prosody generation for domain specific text to speech system. In this method, sentences are composed by inlaying a variable word into each slot in prepared sentence structures. This method can be used for domain specific text-to-speech applications that don't require so many number of sentence structures but many words .

Leija L. et.al 1996 developed a system of text reading and translation to voice for blind persons. The developed system falls into text to speech reading machines and can give to the blind, capacity autonomous of reading text once he learned drive it. The read characters are obtained through an optic lector and a computer program based in pattern recognition by means of an artificial neural network. The system used a central processing unit (CPU) from a personal computer PC, containing only a hard disk, a scanner unit, and a sound blaster audio card. The main key in this project was the words recognition by back propagation neural network which is now very useful, and the use of standards cards. Since the first Kurzweil machine (1) text to speech in 1978 with a prize of 30,000 dollars to the modern computer with a storing high capacity and great velocity and the development techniques of pattern recognition with neural networks, today it is possible the development of relatively inexpensive systems, recognizing the importance of the developed algorithms for pattern recognition, which facilitates the development of aids for blind persons.

Rivarol Vergin, Douglas O’Shoughnessy et.al 1997 proposed an approach to increase the possibilities of speech modifications while preserving most of the speech quality of the original signal. They examined one of the most used methods to modify the current aspect of a speech signal that was the one based on a modification of its apparent pitch contour. The first step of this technique was an estimation of the pitch epochs, followed by a pitch-scale transformation to modify the apparent gender of a current speaker. The apparent rate and intensity contour of the original signal were preserved through the use of a time scale modification. The resulting synthesis signal had an apparent pitch contour relatively different from the input signal. This technique when combined with the pitchscale and time-scale modification procedure, allowed reaching this goal while preserving most of the speech quality of the original signal .

A. P. Breen 1997 suggested that the next generation of synthesis systems will inevitably take more account of the type of information being presented to them, that the interfaces to such systems will become more generic and that the type of processing conducted as part of the synthesis process will become more diffuse and data orientated. He also suggested that advances in speech synthesis can best be achieved when developed within a complete multi-modal spoken language framework .

Leija L. et.al 1999 have proposed a reader instrument for translating text to speech using commercial components as a multimedia computer, a flat bed scanner unit, a control and recognition of characters program, the instrument is used to read printed text by blind people. In their proposed work the character recognition is based on a block of preclassification and artificial neuronal back propagation network, the character are read, through the scanner text from elemental teaching books, and the program locates the words recognized in a abase of recorded words to emit them via voice through the multimedia system. Their system had a limitation; it works with the books of text used in elemental education. The kind of characters is Arial in size of 12 pixels, with a performance of 93% .

Mingli Song, Chun Chen et.al 2003 developed 3D Realistic Talking Face Co-Driven by Text and Speech system. The text is translated into a sequence of visemes transcription. And time vector of the sequence is extracted from the speech corresponding to the text after it is Segmented into phonetic sequence. A muscle based viseme vector is defined for static viseme. And then, with the time vector and the static visemes's sequence, dynamic visemes are generated through time-related dominance function. Finally, according to the frame rate to be rendered, intermediate frames are interpolated between key frames to make the animation result looks more natural and realistic than those obtained based onthe text or speech-driven only.

Soumyajit Dey et.al 2007 proposed architectural optimizations for text to speech synthesis in embedded system. The increasing processing power of embedded devices has created the scope for certain applications that could previously be executed in desktop environments only, to migrate into handheld platforms. An important feature of 24the computing systems of modern times is their support for applications that interact with the user by synthesizing natural speech output. In their work, the performance of a Text to Speech Synthesis application is evaluated on embedded processor architectures and medications in the underlying hardware platform are proposed for real time performance improvement of the concerned application .

Dmitri Bitouk and Shree K. Nayar 2008 proposed a complete framework for creating a speech enabled avatar from a single image of a person. They used a generic facial motion model which represented deformation of a prototype face during speech. They developed a HMM-based facial animation algorithm which took into account both lexical stress and co-articulation. Their algorithm produced realistic animations of the prototype facial surface from either text to speech. They showed several examples of avatars that were driven by text to speech inputs. Such avatars were animated from text or speech input with the help of a novel motion synthesis algorithm. They developed a method for synthesizing eye gaze motion from a photograph. They demonstrated that their approach can also be used to build volumetric displays that feature speech-enabled 3D avatars

Text to speech:

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer, and can be implemented in software or hardware. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech. Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diaphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output. The quality of a speech synthesizer is judged by its similarity to the human voice, and by its ability to be understood. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written works on a home computer.

Electronics devices:

The first computer-based speech synthesis systems were created in the late 1950s, and the first complete text-to-speech system was completed in 1968. In 1961, physicist John Larry Kelly, Jr and colleague Louis Gerstman used an IBM 704 computer to synthesize speech, an event among the most prominent in the history of Bell Labs. Kelly's voice recorder synthesizer (vocoder) recreated the song "Daisy Bell", with musical accompaniment from Max Mathews. Coincidentally, Arthur C. Clarke was visiting his friend and colleague John Pierce at the Bell Labs Murray Hill facility. Clarke was so impressed by the demonstration that he used it in the climactic scene of his screenplay for his novel 2001: A Space Odyssey, where the HAL 9000 computer sings the same song as it is being put to sleep by astronaut Dave Bowman. Despite the success of purely electronic speech synthesis, research is still being conducted into mechanical speech synthesizers.

9.SYNTHESIZER TECHNOLOGIES

The most important qualities of a speech synthesis system are naturalness and intelligibility. Naturalness describes how closely the output sounds like human speech, while intelligibility is the ease with which the output is understood. The ideal speech synthesizer is both natural and intelligible. Speech synthesis systems usually try to maximize both characteristics.

Fig 1.7: Speech synthesizer

The two primary technologies for generating synthetic speech waveforms are concatenative synthesis and formant synthesis. Each technology has strengths and weaknesses, and the intended uses of a synthesis system will typically determine which approach is used.

Concatenative synthesis:

Concatenative synthesis is based on the concatenation (or stringing together) of segments of recorded speech. Generally, concatenative synthesis produces the most naturalsounding synthesized speech. However, differences between natural variations in speech and the nature of the automated techniques for segmenting the waveforms sometimes result in audible glitches in the output. There are three main sub-types of concatenative synthesis.

- **Diphone synthesis**: Diphone synthesis uses a minimal speech database containing all the diphones (sound-to-sound transitions) occurring in a language. The number of diphones depends on the phonotactics of the language: for example, Spanish has about 800 diphones and German about 2500. In diphone synthesis, only one example of each diphone is contained in the speech database. At runtime, the target prosody of a sentence is superimposed on these minimal units by means of digital signal processing techniques 11such as linear predictive coding, PSOLA or MBROLA. The quality of the resulting speech is generally worse than that of unit-selection systems, but more natural-sounding than the output of formant synthesizers. Diphone synthesis suffers from the sonic glitches of concatenative synthesis and the robotic-sounding nature of formant synthesis, and has few of the advantages of either approach other than small size. As such, its use in commercial applications is declining, although it continues to be used in research because there are a number of freely available software implementations.
- **Unit selection synthesis**: Unit selection synthesis uses large databases of recorded speech. During database creation, each recorded utterance is segmented into some or all of the following: individual phones, diphones, half-phones, syllables, morphemes, words, phrases, and sentences. Typically, the division into segments is done using a specially modified speech recognizer set to a "forced alignment" mode with some manual correction afterward, using visual representations such as the waveform and spectrogram. An index of the units in the speech database is then created based on the segmentation and acoustic parameters like the fundamental frequency (pitch), duration, position in the syllable, and neighboring phones. At runtime, the desired target utterance is created by determining the best chain of candidate units from the database (unit selection). This process is typically achieved using a specially weighted decision tree.Unit selection provides the greatest naturalness, because it applies only a small amount of digital signal processing (DSP) to the recorded speech. DSP often makes recorded speech sound less natural, although some systems use a small amount of signal processing at the point of concatenation to smooth the waveform. The output from the best unit-selection systems is often indistinguishable from real human voices, especially in contexts for which the TTS system has been tuned. However, maximum naturalness typically require unit-selection speech databases to be very large, in some systems ranging into the gigabytes of recorded data, representing dozens of hours of speech. Also, unit selection algorithms have been

known to select segments from a place that results in less than ideal synthesis (e.g. minor words become unclear) even when a better choice exists in the database.

- **Domain-specific synthesis**: Domain-specific synthesis concatenates prerecorded words and phrases to create complete utterances. It is used in applications where the variety of texts the system will output is limited to a particular domain, like transit schedule announcements or weather reports. The technology is very simple to implement, and has been in commercial use for a long time, in devices like talking clocks and calculators. The level of naturalness of these systems can be very high because the variety of sentence types is limited, and they closely match the prosody and intonation of the original recordings.Because these systems are limited by the words and phrases in their databases, they are not general-purpose and can only synthesize the combinations of words and phrases with which they have been preprogrammed. The blending of words within naturally spoken language however can still cause problems unless the many variations are taken into account. For example, in non-rhotic dialects of English the "r" in words like "clear" is usually only pronounced when the following word has a vowel as its first letter. Likewise in French, many final consonants become no longer silent if followed by a word that begins with a vowel, an effect called liaison. This alternation cannot be reproduced by a simple word-concatenation system, which would require additional complexity to be context-sensitive.

Formant synthesis:

Formant synthesis does not use human speech samples at runtime. Instead, the synthesized speech output is created using an acoustic model. Parameters such as fundamental frequency, voicing, and noise levels are varied over time to create a waveform of artificial speech. This method is sometimes called rules-based synthesis; however, many concatenative systems also have rules-based components.Many systems based on formant synthesis technology generate Artificial, roboticsounding speech that would never be mistaken for human speech. However, maximum naturalness is not always the goal of a speech synthesis system, and formant Synthesis systems have advantages over concatenative systems. Formant-synthesized speech can be reliably intelligible, even at very high speeds, avoiding the acoustic glitches that commonly plague concatenative systems. High-speed synthesized speech is used by the visually impaired to quickly navigate computers using a screen reader. Formant synthesizers are usually smaller programs than concatenative systems because they do not have a database of speech samples. They can therefore be used in embedded systems, where memory and microprocessor power are especially limited. Because formant-based systems have complete control of all aspects of the output speech, a wide variety of prosodies and intonations can be output, conveying not just questions and statements, but a variety of emotions and tones of voice.

Examples of non-real-time but highly accurate intonation control in formant synthesis include the work done in the late 1970s for the Texas Instruments toy Speak & Spell, and in the early

1980s Sega arcade machines. Creating proper intonation for these projects was painstaking, and the results have yet to be matched by real-time text-to-speech interfaces.

Articulatory synthesis: Articulatory synthesis refers to computational techniques for synthesizing speech based on models of the human vocal tract and the articulation processes occurring there. The first articulatory synthesizer regularly used for laboratory experiments was developed at Haskins Laboratories in the mid-1970s by Philip Rubin, Tom Baer, and Paul Mermelstein. This synthesizer, known as ASY, was based on vocal tract models developed at Bell Laboratories in the 1960s and 1970s by Paul Mermelstein, Cecil Coker, and colleagues.Until recently, articulatory synthesis models have not been incorporated into commercial speech synthesis systems. A notable exception is the NeXT-based system originally developed and marketed by Trillium Sound Research, a spin-off company of the University of Calgary, where much of the original research was conducted. Following the demise of the various incarnations of NeXT (started by Steve Jobs in the late 1980s and merged with Apple Computer in 1997), the Trillium software was published and the work continuing as gnuspeech. The system, first marketed in 1994, provides full articulatory-based text-to-speech conversion using a waveguide or transmission-line analog of the human oral and nasal tracts controlled by Carré's "distinctive region model".

HMM-based synthesis

HMM-based synthesis is a synthesis method based on hidden markov models, also called Statistical Parametric Synthesis. In this system, the frequency spectrum (vocal tract), fundamental frequency(vocal source), and duration (prosody) of speech are modeled simultaneously by HMMs. Speech waveforms are generated from HMMs themselves based on the maximum likelyhood criterion.

Sinewave synthesis:

sinewave synthesis is a technique for synthesizing speech by replacing the formants(main bands of energy) with pure tone whistles.

10.WORKING OF TTS SYSTEM

From now on, it should be clear that a reading machine would hardly adopt a processing scheme as the one naturally taken up by humans, whether it was for language analysis or for speech production itself. Vocal sounds are inherently governed by the partial differential equations of fluid mechanics, applied in a dynamic case since our lung pressure, glottis tension, and vocal and nasal tracts configuration evolve with time. These are controlled by our cortex, which takes advantage of the power of its parallel structure to extract the essence of the text read: its meaning. Even though, in the current state of the engineering art, building a Text-To-Speech synthesizer on such intricate models is almost scientifically conceivable (intensive research on articulatory synthesis, neural networks, and semantic analysis give evidence of it), it would result anyway in a machine with a very high degree of (possibly avoidable) complexity, which is not always compatible with economical criteria. After all, flies do not flap their wings!

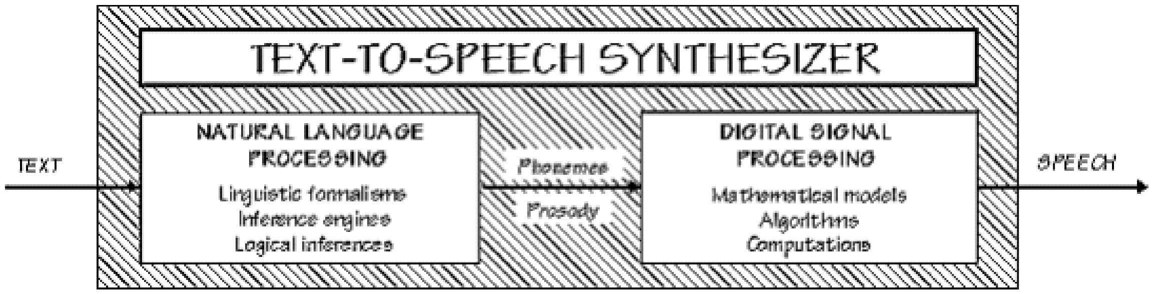


Fig 4.1 General function diagram of a TTS system

Figure 4.1 introduces the functional diagram of a very general TTS synthesizer. As for human reading, it comprises a Natural Language Processing module (NLP), capable of producing a phonetic transcription of the text read, together with the desired intonation and rhythm (often termed as prosody), and a Digital Signal Processing module (DSP), which transforms the symbolic information it receives into speech. But the formalisms and algorithms applied often manage, thanks to a judicious use of mathematical and linguistic knowledge of developers, to short-circuit certain processing steps. This is occasionally achieved at the expense of some restrictions on the text to pronounce, or results in some reduction of the "emotional dynamics" of the synthetic voice (at least in comparison with human performances), but it generally allows to solve the problem in real time with limited memory requirements.

GSM:

The GSM standard was developed as a replacement for first generation (1G) analog cellular networks, and originally described a digital, circuit switched network optimized for full duplex voice telephony. This was expanded over time to include data communications, first by circuit switched transport, then packet data transport via GPRS (General Packet Radio Services) and EDGE (Enhanced Data rates for GSM Evolution or EGPRS).

GSM is a cellular network, which means that cell phones connect to it by searching for cells in the immediate vicinity. There are five different cell sizes in a GSM network—macro, micro, pico, femto and umbrella cells. The coverage area of each cell varies according to the implementation environment. Macro cells can be regarded as cells where the base station antenna is installed on a mast or a building above average roof top level. Micro cells are cells whose antenna height is under average roof top level; they are typically used in urban areas. Picocells are small cells whose coverage diameter is a few dozen metres; they are mainly used indoors. Femtocells are cells designed for use in residential or small business environments and connect to the service provider’s network via a broadband internet connection.

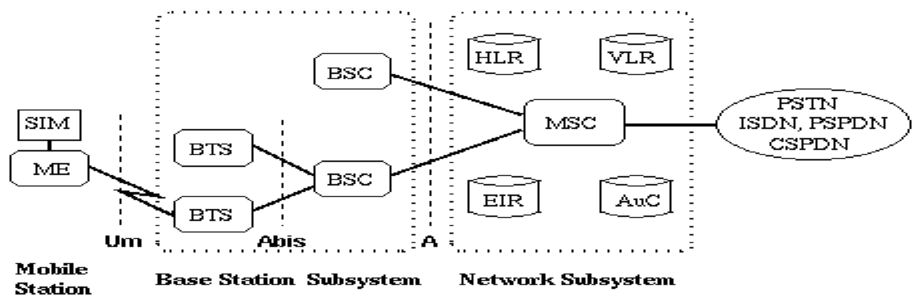
Umbrella cells are used to cover shadowed regions of smaller cells and fill in gaps in coverage between those cells.

GSM networks operate in a number of different carrier frequency ranges (separated into GSM frequency ranges for 2G and UMTS frequency bands for 3G), with most 2G GSM networks operating in the 900 MHz or 1800 MHz bands. Where these bands were already allocated, the 850 MHz and 1900 MHz bands were used instead (for example in Canada and the United States). In rare cases the 400 and 450 MHz frequency bands are assigned in some countries because they were previously used for first-generation systems.

GSM was designed with a moderate level of service security. The system was designed to authenticate the subscriber using a pre-shared key and challenge-response. Communications between the subscriber and the base station can be encrypted. The development of UMTS introduces an optional Universal Subscriber Identity Module (USIM), that uses a longer authentication key to give greater security, as well as mutually authenticating the network and the user – whereas GSM only authenticates the user to the network (and not vice versa). The security model therefore offers confidentiality and authentication, but limited authorization capabilities, and no non-repudiation.

ARCHITECTURE OF GSM NETWORK

A GSM network is composed of several functional entities, whose functions and interfaces are specified. Figure 1 shows the layout of a generic GSM network. The GSM network can be divided into three broad parts. The Mobile Station is carried by the subscriber. The Base Station Subsystem controls the radio link with the Mobile Station. The Network Subsystem, the main part of which is the Mobile services Switching Center (MSC), performs the switching of calls between the mobile users, and between mobile and fixed network users. The MSC also handles the mobility management operations. Not shown is the Operations and Maintenance Center, which oversees the proper operation and setup of the network. The Mobile Station and the Base Station Subsystem communicate across the Um interface, also known as the air interface or radio link. The Base Station Subsystem communicates with the Mobile services Switching Center across the A interface.



SIM Subscriber Identity Module	BSC Base Station Controller	MSC Mobile services Switching Center
ME Mobile Equipment	HLR Home Location Register	EIR Equipment Identity Register
BTS Base Transceiver Station	VLR Visitor Location Register	AuC Authentication Center

Mobile Station

The mobile station (MS) consists of the mobile equipment (the terminal) and a smart card called the Subscriber Identity Module (SIM). The SIM provides personal mobility, so that the user can have access to subscribed services irrespective of a specific terminal. By inserting the SIM card into another GSM terminal, the user is able to receive calls at that terminal, make calls from that terminal, and receive other subscribed services.

The mobile equipment is uniquely identified by the International Mobile Equipment Identity (IMEI). The SIM card contains the International Mobile Subscriber Identity (IMSI) used to identify the subscriber to the system, a secret key for authentication, and other information. The IMEI and the IMSI are independent, thereby allowing personal mobility. The SIM card may be protected against unauthorized use by a password or personal identity number.

Base Station Subsystem

The Base Station Subsystem is composed of two parts, the Base Transceiver Station (BTS) and the Base Station Controller (BSC). These communicate across the standardized Abis interface, allowing (as in the rest of the system) operation between components made by different suppliers.

The Base Transceiver Station houses the radio tranceivers that define a cell and handles the radio-link protocols with the Mobile Station. In a large urban area, there will potentially be a large number of BTSs deployed, thus the requirements for a BTS are ruggedness, reliability, portability, and minimum cost.

The Base Station Controller manages the radio resources for one or more BTSs. It handles radio-channel setup, frequency hopping, and handovers, as described below. The BSC is the connection between the mobile station and the Mobile service Switching Center (MSC).

Network Subsystem

The central component of the Network Subsystem is the Mobile services Switching Center (MSC). It acts like a normal switching node of the PSTN or ISDN, and additionally provides all the functionality needed to handle a mobile subscriber, such as registration, authentication, location updating, handovers, and call routing to a roaming subscriber. These services are provided in conjuction with several functional entities, which together form the Network Subsystem. The MSC provides the connection to the fixed networks (such as the PSTN or ISDN). Signalling between functional entities in the Network Subsystem uses Signalling System Number 7 (SS7), used for trunk signalling in ISDN and widely used in current public networks.

The Home Location Register (HLR) and Visitor Location Register (VLR), together with the MSC, provide the call-routing and roaming capabilities of GSM. The HLR contains all the administrative information of each subscriber registered in the corresponding GSM network, along with the current location of the mobile. The location of the mobile is typically in the form of the signalling address of the VLR associated with the mobile station. The actual routing procedure will be described later. There is logically one HLR per GSM network, although it may be implemented as a distributed database.

The Visitor Location Register (VLR) contains selected administrative information from the HLR, necessary for call control and provision of the subscribed services, for each mobile currently located in the geographical area controlled by the VLR. Although each functional entity can be implemented as an independent unit, all manufacturers of switching equipment to date implement the VLR together with the MSC, so that the geographical area controlled by the MSC corresponds to that controlled by the VLR, thus simplifying the signalling required. Note that the MSC contains no information about particular mobile stations --- this information is stored in the location registers.The other two registers are used for authentication and security purposes. The Equipment Identity Register (EIR) is a database that contains a list of all valid mobile equipment on the network, where each mobile station is identified by its International Mobile Equipment Identity (IMEI). An IMEI is marked as invalid if it has been reported stolen or is not type approved. The Authentication Center (AuC) is a protected database that stores a copy of the secret key stored in each subscriber's SIM card, which is used for authentication and encryption over the radio channel

The NLP component:

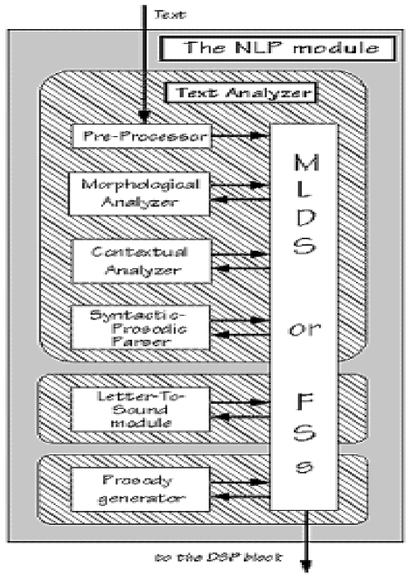


Fig 4.2: The NLP module of a general Text to Speech conversion system.

Figure 4.2 introduces the skeleton of a general NLP module for TTS purposes. One immediately notices that, in addition with the expected letter-to-sound and prosody generation blocks, it comprises a morpho-syntactic analyzer, underlying the need for some syntactic processing in a high quality Text-To-Speech system. Indeed, being able to reduce a given sentence into something like the sequence of its parts-of-speech, and to further describe it in the form of a syntax tree, which unveils its internal structure, is required for at least two reasons:

A) Accurate phonetic transcription can only be achieved provided the part of speech category of some words is available, as well as if the dependency relationship between successive words is known.

B) Natural prosody heavily relies on syntax. It also obviously has a lot to do with semantics and pragmatics, but since very few data is currently available on the generative aspects of this dependence, TTS systems merely concentrate on syntax. Yet few of them are actually provided with full disambiguation and structuration capabilities.

<u>11.TEXT ANALYSIS</u>

The text analysis block is itself composed of:

- A pre-processing module, which organizes the input sentences into manageable lists of words. It identifies numbers, abbreviations, acronyms and idiomatic and transforms them into full text when needed. An important problem is encountered as soon as the character level: that of punctuation ambiguity (including the critical case of sentence end detection). It can be solved, to some extent, with elementary regular grammars.
- A morphological analysis module, the task of which is to propose all possible part of speech categories for each word taken individually, on the basis of their spelling. Inflected, derived, and compound words are decomposed into their elementary graphemic units (their morphs) by simple regular grammars exploiting lexicons of stems and affixes.
- The contextual analysis module considers words in their context, which allows it to reduce the list of their possible part of speech categories to a very restricted number of highly probable hypotheses, given the corresponding possible parts of speech of neighboring words. This can be achieved either with n-grams, which describe local syntactic dependences in the form of probabilistic finite state automata (i.e. as a Markov model), to a lesser extent with multi-layer perceptrons (i.e., neural networks) trained to uncover contextual rewrite rules, or with local, non-stochastic grammars provided by expert linguists or automatically inferred from a training data set with classification and regression tree.

- Syntactic-prosodic parser, which examines the remaining search space and finds the text structure (i.e. its organization into clause and phrase-like constituents) which more closely relates to its expected prosodic realization.

Automatic phonetization:

The Letter-To-Sound (LTS) module is responsible for the automatic determination of the phonetic transcription of the incoming text. It thus seems, at first sight, that its task is as simple as performing the equivalent of a dictionary look-up! From a deeper examination, however, one quickly realizes that most words appear in genuine speech with several phonetic transcriptions, many of which are not even mentioned in pronunciation dictionaries. Namely:

A) Pronunciation dictionaries refer to word roots only. They do not explicitly account for morphological variations (i.e. plural, feminine, conjugations, especially for highly inflected languages, such as French), which therefore have to be dealt with by a specific component of phonology, called morphophonology.

B) Some words actually correspond to several entries in the dictionary, or more generally to several morphological analyses, generally with different pronunciations. This is typically the case of heterophonic homographs, i.e. words that are pronounced differently even though they have the same spelling, as for 'record', constitute by far the most tedious class of pronunciation ambiguities. Their correct pronunciation generally depends on their part-of-speech and most frequently contrasts verbs and non-verbs , as for 'contrast' (verb/noun) or 'intimate' (verb/adjective), although it may also be based on syntactic features, as for 'read' (present/past)

C) Pronunciation dictionaries merely provide something that is closer to a phonemic transcription than from a phonetic one (i.e. they refer to phonemes rather than to phones). As denoted by Withgott and Chen : "while it is relatively straightforward to build computational models for morphophonological phenomena, such as producing the dictionary pronunciation of 'electricity' given a base form 'electric', it is another matter to model how that ronunciation actually sounds". Consonants, for example, may reduce or delete in clusters, a phenomenon termed as consonant cluster simplification, as in 'softness' in which [t] fuses in a single gesture with the following.

D) Words embedded into sentences are not pronounced as if they were isolated. Surprisingly enough, the difference does not only originate in variations at word boundaries (as with phonetic liaisons), but also on alternations based on the organization of the sentence into non-lexical units, that is whether into groups of words (as for phonetic lengthening) or into non-lexical parts thereof (many phonological processes, for instance, are sensitive to syllable structure).

E) Finally, not all words can be found in a phonetic dictionary: the pronunciation of new words and of many proper names has to be deduced from the one of already known words. Clearly, points A and B heavily rely on a preliminary morphosyntactic (and possibly semantic) analysis of the sentences to read. To a lesser extent, it also happens to be the case for point C as well, since reduction processes are not only a matter of contextsensitive phonation, but they also rely on morphological structure and on word grouping, that is on morphosyntax. Point D puts a strong demand on sentence analysis, whether syntactic or metrical, and point E can be partially solved by addressing morphology and/or by finding graphemic analogies between words.It is then possible to organize the task of the LTS module in many ways (Fig.4.3), often roughly classified into dictionary-based and rule-based strategies, although many intermediate solutions exist.

Fig 4.3: Dictionary-based (left) versus rule-based (right) phonetization.

Dictionary-based solutions consist of storing a maximum of phonological knowledge into a lexicon. In order to keep its size reasonably small, entries are generally restricted to morphemes, and the pronunciation of surface forms is accounted for by inflectional, derivational, and compounding morphophonemic rules which describe how the phonetic transcriptions of their morphemic constituents are modified when they are combined into words. Morphemes that cannot be found in the lexicon are transcribed by rule. After a first phonemic transcription of each word has been obtained, some phonetic postprocessing is generally applied, so as to account for co-articulatory smoothing phenomena. This approach has been followed by the MITTALK system from its very first day. A dictionary of up to 12,000 morphemes covered about 95% of the input words. The AT&T Bell Laboratories TTS system follows the same guideline, with an augmented morpheme lexicon of 43,000 morphemes. A rather different strategy is adopted in rule-based transcription systems, which transfer most of the phonological competence of dictionaries into a set of letter-to-sound (or grapheme-to-phoneme) rules. This time, only those words that are pronounced in such a

particular way that they constitute a rule on their own are stored in an exceptions dictionary. Notice that, since many exceptions are found in the most frequent words, a 44 reasonably small exceptions dictionary can account for a large fraction of the words in a running text. In English, for instance, 2000 words typically suffice to cover 70% of the words in text. It has been argued in the early days of powerful dictionary-based methods that they were inherently capable of achieving higher accuracy than letter-to-sound rules, given the availability of very large phonetic dictionaries on computers. On the other hand, considerable efforts have recently been made towards designing sets of rules with a very wide coverage (starting from computerized dictionaries and adding rules and exceptions until all words are covered. Clearly, some trade-off is inescapable. Besides, the compromise is language-dependent, given the obvious differences in the reliability of letter-to-sound correspondences for different languages.

Prosody generation

The term prosody refers to certain properties of the speech signal which are related to audible changes in pitch, loudness, and syllable length. Prosodic features have specific functions in speech communication (see Fig. 4.4). The most apparent effect of prosody is that of focus. For instance, there are certain pitch events which make a syllable stand out within the utterance, and indirectly the word or syntactic group it belongs to will be highlighted as an important or new component in the meaning of that utterance. The presence of a focus marking may have various effects, such as contrast, depending on the place where it occurs, or the semantic context of the utterance.

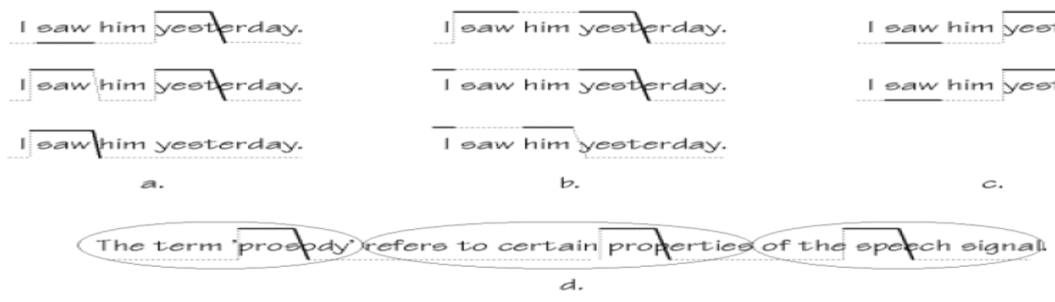

I saw him yesterday.
a.
b.
c.
The term 'prosody' refers to certain properties of the speech signal.
d.

Fig 4.4: Different kinds of information provided by intonation (lines indicate pitch movements; solid lines indicate stress).

a. Focus or given/new information;

b. Relationships between words (saw-yesterday; I-yesterday; I-him)

c. Finality (top) or continuation (bottom), as it appears on the last syllable;

d. Segmentation of the sentence into groups of syllables.

Although maybe less obvious, there are other, more systematic or general functions. Prosodic features create a segmentation of the speech chain into groups of syllables, or, put the other way round, they give rise to the grouping of syllables and words into larger chunks. Moreover, there are prosodic features which indicate relationships between such groups, indicating that two or more groups of syllables are linked in some way. This grouping effect is hierarchical, although not necessarily identical to the syntactic structuring of the utterance.The key idea is that the "correct" syntactic structure, the one that precisely requires some semantic and pragmatic insight, is not essential for producing such a prosody. With these considerations in mind, it is not surprising that commercially developed TTS system have emphasized coverage rather than linguistic sophistication, by concentrating 46 their efforts on text analysis strategies aimed to segment the surface structure of incoming sentences, as opposed to their syntactically, semantically, and pragmatically related deep structure. The resulting syntactic-prosodic descriptions organize sentences in terms of prosodic groups strongly related to phrases (and therefore also termed as minor or intermediate phrases), but with a very limited amount of embedding, typically a single level for these minor phrases as parts of higher-order prosodic phrases (also termed as major or intonational phrases, which can be seen as a prosodic-syntactic equivalent for clauses) and a second one for these major phrases as parts of sentences, to the extent that the related major phrase boundaries can be safely obtained from relatively simple text analysis methods. In other words, they focus on obtaining an acceptable segmentation and translate it into the continuation or finality marks of Fig. 4.c, but ignore the relationships or contrastive meaning of Fig. 4.4.a and b.

A (minor) prosodic phrase = a sequence of chinks followed by a sequence of chunks in which chinks and chunks belong to sets of words which basically correspond to function and content words, respectively, with the difference that objective pronouns (like 'him' or 'them') are seen as chunks and that tensed verb forms (such as 'produced') are considered as chinks.

The DSP component :

Intuitively, the operations involved in the DSP module are the computer analogue of dynamically controlling the articulatory muscles and the vibratory frequency of the vocal folds so that the output signal matches the input requirements. In order to do it properly, the DSP module should obviously, in some way, take articulatory constraints into account, since it has been known for a long time that phonetic transitions are more important than stable states for the understanding of speech. This, in turn, can be basically achieved in two ways:

- Explicitly, in the form of a series of rules which formally describe the influence of phonemes on one another;
- Implicitly, by storing examples of phonetic transitions and co-articulations into a speech segment database, and using them just as they are, as ultimate acoustic units (i.e. in place of phonemes).Two main classes of TTS systems have emerged from this alternative, which quickly turned into synthesis philosophies given the divergences they present in their means and objectives: synthesis-by-rule and synthesis-by-concatenation.

Rule-based synthesizers:

Rule-based synthesizers are mostly in favor with phoneticians and phonologists, as they constitute a cognitive, generative approach of the phonation mechanism. The broad spreading of the Klatt synthesizer, for instance, is principally due to its invaluable assistance in the study of the characteristics of natural speech, by analytic listening of rule-synthesized speech. What is more, the existence of relationships between articulatory parameters and the inputs of the Klatt model make it a practical tool for investigating physiological constraints.For historical and practical reasons (mainly the need for a physical interpretability of the model), rule synthesizers always appear in the form of formant synthesizers. These describe speech as the dynamic evolution of up to 60 parameters [Stevens 90], mostly related to formant and anti-formant frequencies and bandwidths together with glottal waveforms. Clearly, the large number of (coupled) parameters complicates the analysis stage and tends to produce analysis errors. What is more, formant frequencies and bandwidths are inherently difficult to estimate from speech data. The need for intensive trials and errors in order to cope with analysis errors, makes them time-consuming systems to develop (several years are commonplace). Yet, the synthesis quality achieved up to now reveals typical buzzyness problems, which originate from the rules themselves: introducing a high degree of naturalness is theoretically possible, but the rules to do so are still to be discovered.

Rule-based synthesizers remain, however, a potentially powerful approach to speech synthesis. They allow, for instance, to study speaker-dependent voice features so that switching from one synthetic voice into another can be achieved with the help of specialized rules in the rule database. Following the same idea, synthesis-by-rule seems to be a natural way of handling the articulatory aspects of changes in speaking styles (as opposed to their prosodic counterpart, which can be accounted for by concatenationbased synthesizers as well).

Concatenative synthesizers:

As opposed to rule-based ones, concatenative synthesizers possess a very limited knowledge of the data they handle: most of it is embedded in the segments to be chained up. This clearly appears in figure 4.5, where all the operations that could indifferently be used in the context of a music synthesizer (i.e. without any explicit reference to the inner nature of the sounds to be processed) have been grouped into a sound processing block, as opposed to the upper speech processing block whose design requires at least some understanding of phonetics.

- Database operation: A series of preliminary stages have to be fulfilled before the synthesizer can produce its first utterance. At first, segments are chosen so as to minimize future concatenation problems. A combination of diphones (i.e. units that begin in the middle of the stable state of a phone and end in the middle of the following one), halfsyllables, and triphones (which differ from diphones in that they include a complete central phone) are often chosen as speech units, since they involve most of the transitions and co-articulations while requiring an affordable amount of memory. When a complete list of segments has emerged, a corresponding list of words is carefully completed, in such a way that each segment appears at least once (twice is better, for security). Unfavorable positions like inside stressed syllables or in strongly reduced (i.e. over-coarticulated) contexts, are excluded. A corpus is then digitally recorded and stored, and the elected segments are spotted, either manually with the help of signal visualization tools, or automatically thanks to segmentation algorithms, the decisions of which are checked and corrected interactively. A segment database finally centralizes the results, in the form of the segment names, waveforms, durations, and internal sub-splitting. In the case of diphones, for example, the position of the border between phones should be stored, so as to be able to modify the duration of one half-phone without affecting the length of the other one.Segments are then often given a parametric form, in the form of a temporal sequence of vectors of parameters collected at the output of a speech analyzer and stored in a 50 parametric segment database. The advantage of using a speech model originates in the fact that:

• The upper left hatched block corresponds to the development of the synthesizer (i.e. it is processed once for all). Other blocks correspond to run-time operations. Languagedependent operations and data are indicated by a flag.

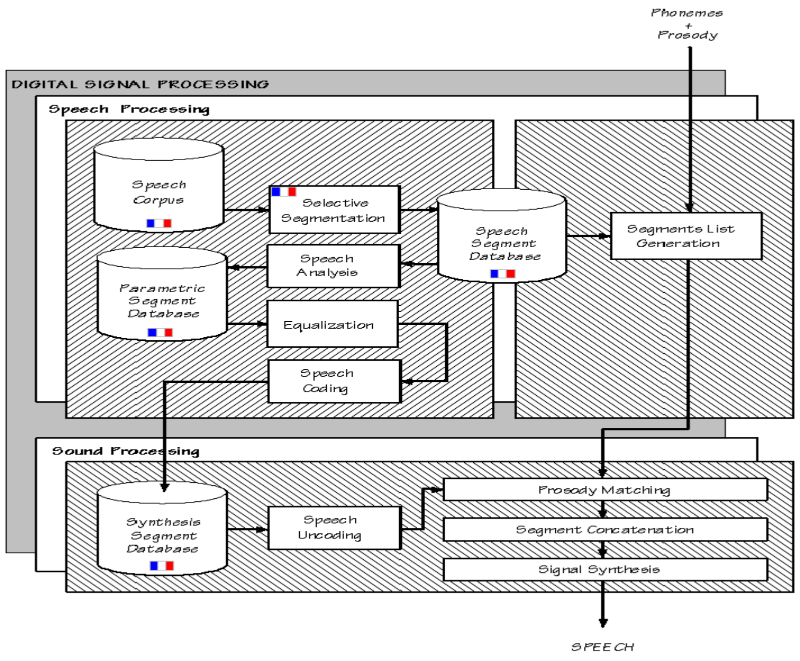


Figure 4.5: A general concatenation-based synthesizer.

• Well chosen speech models allow data size reduction, an advantage which is hardly negligible in the context of concatenation-based synthesis given the amount of data to be stored. Consequently, the analyzer is often followed by a parametric speech coder.

• A number of models explicitly separate the contributions of respectively the source and the vocal tract, an operation which remains helpful for the pre-synthesis operations: prosody matching and segments concatenation.Indeed, the actual task of the synthesizer is to produce, in real-time, an adequate sequence of concatenated segments, extracted from its parametric segment database and the prosody of which has been adjusted from their stored value, i.e. the intonation and the duration they appeared with in the original speech corpus, to the one imposed by the language processing module. Consequently, the respective parts played by the prosody matching and segments concatenation modules are considerably alleviated when input segments are presented in a form that allows easy modification of their pitch, duration, and spectral envelope, as is hardly the case with crude waveform samples. Since segments to be chained up have generally been extracted from different words, that is in different phonetic contexts, they often present amplitude and timbre mismatches. Even in the case of stationary vocalic sounds, for instance, a rough sequencing of parameters typically leads to audible discontinuities. These can be coped with during the constitution of the synthesis segments database, thanks to an equalization in which related endings of segments are imposed similar amplitude spectra, the difference being distributed on their neighborhood. In practice, however, this operation, is restricted to amplitude parameters: the equalization stage smoothly modifies the energy levels at the beginning and at the end of segments, in such a way as to eliminate amplitude mismatches (by setting the energy of all the phones of a given phoneme to their average value). In contrast, timbre conflicts are better tackled at run-time, by smoothing individual couples of segments when necessary rather than equalizing them once for all, so that some of the phonetic variability naturally introduced by co-articulation is still maintained. In practice, amplitude equalization can be performed either before or after speech analysis (i.e. on crude samples or on speech parameters).Once the parametric segment database has been completed, synthesis itself can begin.

- Speech Synthesis: A sequence of segments is first deduced from the phonemic input of the synthesizer, in a block termed as segment list generation, which interfaces the NLP and DSP modules. Once prosodic events have been correctly assigned to individual segments, the prosody matching module queries the synthesis segment database for the actual parameters, adequately uncoded, of the elementary sounds to be used, and adapts them one by one to the required prosody. The segment concatenation block is then in charge of dynamically matching segments to one another, by smoothing discontinuities. Here again, an adequate modelization of speech is highly profitable, provided simple interpolation schemes performed on its parameters approximately correspond to smooth acoustical transitions between sounds. The resulting stream of parameters is finally presented at the input of a synthesis block, the exact counterpart of the analysis one. Its task is to produce speech.
- Segmental Quality: The efficiency of concatenative synthesizers to produce high quality speech is mainly subordinated to the type of segments chosen.
  - Segments should obviously exhibit some basic properties:

    o They should allow to account for as many co-articulatory effects as possible.

    o Given the restricted smoothing capabilities of the concatenation block, they should be easily connectable.

    o Their number and length should be kept as small as possible.

o On the other hand, longer units decrease the density of concatenation points, therefore providing better speech quality. Similarly, an obvious way of accounting for articulatory phenomena is to provide many variants for each phoneme. This is clearly in contradiction with the limited memory constraint. Some trade-off is necessary. Diphones are often chosen. They are not too numerous (about 1200 for French, including lots of phoneme sequences that are only encountered at word boundaries, for 3 minutes of speech, i.e. approximately 5 Mbytes of 16 bits samples at 16 kHz) and they do 53incorporate most phonetic transitions. No wonder then that they have been extensively used. They imply, however, a high density of concatenation points (one per phoneme), which reinforces the importance of an efficient concatenation algorithm. Besides, they can only partially account for the many co-articulatory effects of a spoken language, since these often affect a whole phone rather than just its right or left halves independently. Such effects are especially patent when somewhat transient phones, such as liquids and (worst of all) semi-vowels, are to be connected to each other. Hence the use of some larger units as well, such as triphones.

2. The model of speech signal, to which the analysis and synthesis algorithms refer. The models used in the context of concatenative synthesis can be roughly classified into two groups, depending on their relationship with the actual phonation process. Production models provide mathematical substitutes for the part respectively played by vocal folds, nasal and vocal tracts, and by the lips radiation. Their most representative members are Linear Prediction Coding (LPC) synthesizers. On the contrary, phenomenological models intentionally discard any reference to the human production mechanism. Among these

pure digital signal processing tools, spectral and time-domain approaches are increasingly encountered in TTS systems. Two leading such models exist: the hybrid Harmonic/Stochastic (H/S) model and the Time-Domain Pitch-Synchronous-OveraLapAdd (TD-PSOLA) one. The latter is a time-domain algorithm: it virtually uses no speech explicit speech model. It exhibits very interesting practical features: a very high speech quality (the best currently available) combined with a very low computational cost (7 operations per sample on the average). The hybrid Harmonic/stochastic model is intrinsically more powerful than the TD-PSOLA one, but it is also about ten times more computationally intensive. PSOLA synthesizes are now widely used in the speech synthesis community. The recently developed MBROLA algorithm even provides a timedomain algorithm which exhibits the very efficient smoothing apabilities of the H/S model (for the spectral envelope mismatches that cannot be avoided at concatenation points) as well as its very high data compression ratios (up to 10 with almost no additional computational cost) while keeping the computational complexity of PSOLA.

One of the practical usage if TTS converter is **grapheme to phoneme**

conversion for text-to-speech synthesis in French

Text-to-speech (TTS) synthesis systems usually involve three main submodules applyingin sequence. The first stage of the synthesis, traditionally referred to as grapheme-tophoneme conversion, consists of translating a written utterance into the correspondingstream of phonemes (including, for some languages, the encoding of lengthenedphonemes, of lexically stressed syllables, of syllable boundaries, etc.). The second stage consists of computing a series of prosodic markers to be attached to this phonemicstring. The last stage deals with the actual production of the speech waveforms. Speechsynthesis can be viewed as a chain: the output quality depends on the quality of individual components. It thus makes sense to conduct a specific evaluation for eachpart of this chain.In language such as French (likewise, in English), grapheme-to-phoneme (GP)conversion is diYcult. A first diYculty is that the French orthographical system isoverly complex and contains, mainly for historical reasons, a large number of irregularities (Catach, 1984; Belrhali, 1995). As a result, any accurate rule-based description of the correspondence between graphemes and phonemes needs to incorporatea fairly large number of very specific rules and exceptions. These rules are furtherobscured by a number of so-called heterophonous homographs, i.e. word forms whosepronunciation varies according to the environment. A second diYculty is that thephonology of French presents some intriguing aspects, whose linguistic description isstill subject to open controversies; these aspects need nonetheless be accounted for ina GP conversion system. A typical problem is the problem of the mute-e (or schwa),which may be either uttered or dropped (Larreur & Sorin, 1990), both word-internallyand at the junction between successive words. Other problems of contextual variabilityoccur in the case of glides, which may be uttered in a syllabic manner (diæresis) or not(synæresis), and in the case of liaisons. Liaison, here, refers to the phonetic realizationof a word final consonant in the context of a following word initial vowel or mute-h,which can be compulsory, forbidden, or optional. An example of each of thesepossibilities is given in the sentence:les enfants ont e´coute´(the children have listened)Liaison is compulsory between les and enfants, forbidden between enfants and ont,and optional between ont and e´coute´. In any case, predicting the accurate realizationof these variable phonemes requires a subtle analysis of their phonological, morphological, and even syntactical environment. Finally, as is the case in most languages,extra-lexical items such as proper names, numbers and abbreviations, which are frequently found in real-world texts, also raise complex problems. A review of thesediYculties, and of the solutions that have been put forward in the context of automatedGP conversion, can be found, for instance, in Auberge´ (1991), Be´chet and El-Be`ze(1996), Boula de Mareu¨il (1997), Divay (1990), Dutoit (1993), Gaudinat and Wherli(1997), Keller (1997), Lacheret-Dujour (1990), Laporte (1988), Marty (1992), O'Shaughnessy (1984) and Yvon (1996).The extent to which these problems are solved, or still impair the quality of TTS, ishowever poorly appreciated. Therefore, the evaluation of GP conversion is an importantproblem for evaluating TTS systems. Quite an abundant literature exists about TTSevaluation methods (Silverman, Basson & Levas, 1990; Carlson, Granstro¨m & Nord,1990; van Bezooijen & Pols, 1990; van Santen, 1993; Kraft & Portele, 1995; Sorin &Emerard, 1996; Klaus, Fellbaum & Sotscheck, 1997; Benoıˆt, 1997; Pols & Jekosch,1997). However, to the knowledge of the authors, no specific methods, corpora or stateof-the-art reports are currently available for GP conversion, and in particular for GPconversion in French.The aim of this paper is to report joint experiments conducted in the French-speakingacademic community on evaluation of GP conversions in French, for TTS synthesis.The systems involved are all relying on a rule-based approach; nonetheless, they diVergreatly in the number of rules involved in GP conversion, which ranges between 500and about 4000. In all the systems, rules may be superseded by look-up in exceptions lexica. These lexica may contain as few as a handful of very specific words, or as manyas thousands of irregular word forms. These figures may be diYcult to interpret, asthere is no clear cut distinction between rules and exceptions. The systems also diVerin the amount of linguistic pre-processing of the text: while some take advantage of afairly accurate general purpose syntactic analysis module, others rely onad hocheuristicsto perform morpho-syntactic disambiguation. Some systems also use specific modulesfor pronouncing proper names. However, this is not the general case. Precise descriptionof the systems involved in this test is out of the scope of the present paper, and thereader is invited to refer to the paper cited above for more details.The keypoints and main features of the present work are the following:Objective evaluation: For a linguistic task such as GP conversion, it seems better to useautomatic evaluation tools, based on enriched text corpora, rather than subjectivetests, which are better suited to the assessment of complete TTS systems.Diagnostic approach: Following the grid proposed in Gibbon, Moore and Winski (1997:chapter 12), we preferred a diagnostic approach. System developers need tosystematically and objectively evaluate the behaviour of their programs at a verydetailed level of precision, in order to concentrate their eVorts on the most defective part of their system. Surprisingly enough, very little has been done so far toprovide system developers with suitable, i.e. “objective”, diagnostic evaluationmethodologies.International evalution: Eight diVerent systems were tested in the experiments. They were provided by teams from Canada, Belgium, France and Switzerland in the following universities or research institute: ENST (Paris), LIMSI (Orsay), ICP (Grenoble), LIA (Avignon), INRS (Que´bec), TCTS (Mons), LAIP (Lausanne) and LATL (Geneva). LPL (Aix-en-Provence) was in charge of organization and corpora development. Methodology design: One of the main issues, and one of the most diYcult tasks, for GP converter evaluation was the design of a suitable methodology. This is discussed in detail below. Corpus design: The corpus format and content designed for these experiments will be useful for other systems as well, since they will be at the disposal of the scientific community.Strictly speaking, GP conversion refers to the process of converting a stream of orthographical symbols into an appropriate symbolic representation of the corresponding sequence of sounds. This output usually takes the form of a series of phonemic symbols. The usefulness of an automated GP conversion device has been demonstrated in various natural language and speech processing applications, such as the correction of spelling errors, speech synthesis and large-vocabulary speech recognition. Depending on the target application, the specifications of a GP converter are slightly diVerent, and this should be taken into account in the design of the evaluation methodology. For instance, whereas speech synthesis systems usually output one single phonemic string for each input, GP converters used in the context of speech recognition ought to produce multiple pronunciations of their input, in order to model properly the speech variability. In this project, the evaluation methodology is tuned to the specific task of speech synthesis.Even in this context, GP conversion modules can be given quite diVerent tasks. This task can be as restricted as the pronunciation of isolated items from a potentially large list, as is typically the case for reverse directory inquiry systems, or can consist of the pronunciation ofrichly annotated linguistic representations, as in the context of dialogue systems, or more generally, concept-to-speech systems. Text-to-speech systems put other kinds of requirements on their GP conversion module, since their input may potentially be any kind of written text (news articles, e-mail, etc.). Ideally, an evaluation of GP conversion systems should include a specific experimental design for each of these tasks. Our experiments are nevertheless restricted to newspaper or book reading. This paper is organized as follows. In the next section, we present the methodology. Section 3 describes the corpus design and content. Section 4 presents the results obtained for the eight systems. Section 5 discusses the methodology and summarizes the results obtained, which give an accurate state-of-the-art of GP conversion in French. Directions for future studies are also envisaged.

12.CHALLENGES

12.1.Text normalization challenges

The process of normalizing text is rarely straightforward. Texts are full of heteronyms, numbers, and abbreviations that all require expansion into a phonetic representation. There are many spellings in English which are pronounced differently based on context. For example, "My latest project is to learn how to better project my voice" contains two pronunciations of "project".

Most text-to-speech (TTS) systems do not generate semantic representations of their input texts, as processes for doing so are not reliable, well understood, or computationally effective. As a result, various heuristic techniques are used to guess the proper way to disambiguate homographs, like examining neighboring words and using statistics about frequency of occurrence.

Recently TTS systems have begun to use HMMs (discussed above) to generate "parts of speech" to aid in disambiguating homographs. This technique is quite successful for many cases such as whether "read" should be pronounced as "red" implying past tense, or as "reed" implying present tense. Typical error rates when using HMMs in this fashion are usually below five percent. These techniques also work well for most European languages, although access to required training corpora is frequently difficult in these languages.

Deciding how to convert numbers is another problem that TTS systems have to address. It is a simple programming challenge to convert a number into words (at least in English), like "1325" becoming "one thousand three hundred twenty-five." However, numbers occur in many different contexts; "1325" may also be read as "one three two five", "thirteen twenty-five" or "thirteen hundred and twenty five". A TTS system can often infer how to expand a number based on surrounding words, numbers, and punctuation, and sometimes the system provides a way to specify the context if it is ambiguous.Roman numerals can also be read differently depending on context. For example "Henry VIII" reads as "Henry the Eighth", while "Chapter VIII" reads as "Chapter Eight".

Similarly, abbreviations can be ambiguous. For example, the abbreviation "in" for "inches" must be differentiated from the word "in", and the address "12 St John St." uses the same abbreviation for both "Saint" and "Street". TTS systems with intelligent front ends can make educated guesses about ambiguous abbreviations, while others provide the same result in all cases, resulting in nonsensical (and sometimes comical) outputs, such as "co-operation" being rendered as "company operation".

12.2.Text-to-phoneme challenges

Speech synthesis systems use two basic approaches to determine the pronunciation of a word based on its spelling, a process which is often called text-to-phoneme or grapheme-to-phoneme conversion (phoneme) is the term used by linguists to describe distinctive sounds in a language). The simplest approach to text-to-phoneme conversion is the dictionary-based approach, where a large dictionary containing all the words of a language and their correct pronunciations is stored by the program. Determining the correct pronunciation of each word is a matter of looking up each word in the dictionary and replacing the spelling with the pronunciation specified in the dictionary. The other approach is rule-based, in which pronunciation rules are applied to words to determine their pronunciations based on their spellings. This is similar to the "sounding out", or synthetic phonics, approach to learning reading.

Each approach has advantages and drawbacks. The dictionary-based approach is quick and accurate, but completely fails if it is given a word which is not in its dictionary. As dictionary size grows, so too does the memory space requirements of the synthesis system. On the other hand, the rule-based approach works on any input, but the complexity of the rules grows substantially as the system takes into account irregular spellings or pronunciations. (Consider that the word "of" is very common in English, yet is the only word in which the letter "f" is pronounced [v].) As a result, nearly all speech synthesis systems use a combination of these approaches.

Languages with a phonemic orthography have a very regular writing system, and the prediction of the pronunciation of words based on their spellings is quite successful. Speech synthesis systems for such languages often use the rule-based method extensively, resorting to dictionaries only for those few words, like foreign names and borrowings, whose pronunciations are not obvious from their spellings. On the other hand, speech synthesis systems for languages like English, which have extremely irregular spelling systems, are more likely to rely on dictionaries, and to use rule-based methods only for unusual words, or words that aren't in their dictionaries.

Evaluation challenges

The consistent evaluation of speech synthesis systems may be difficult because of a lack of universally agreed objective evaluation criteria. Different organizations often use different speech data. The quality of speech synthesis systems also depends to a large degree on the quality of the production technique (which may involve analogue or digital recording) and on the facilities used to replay the speech. Evaluating speech synthesis systems has therefore often been compromised by differences between production techniques and replay facilities.

Recently, however, some researchers have started to evaluate speech synthesis systems using a common speech dataset.

Prosodics and emotional content

A study in the journal *Speech Communication* by Amy Drahota and colleagues at the University of Portsmouth, UK, reported that listeners to voice recordings could determine, at better than chance levels, whether or not the speaker was smiling.It was suggested that identification of the vocal features that signal emotional content may be used to help make synthesized speech sound more natural.

13.DEDICATED HARDWARE

- Votrax
  - SC-01A(analogformant)http://en.wikipedia.org/wiki/File:TextSpeak_Embedded_Text_to_Speech_on_a_Chip.jpg
  - SC-02 / SSI-263 / "Artic 263"
  - General Instrument SP0256_AL2(CTS256A-AL2)
- National Semiconductor DT1050 Digitalker (Mozer - Forrest Mozzer)
- Silicon Systems SSI 263 (analog formant)
- Texas Instruments LPC Speech Chips** TMS5110A ** TMS5200
- MSP50C6XX - Sold to Sensory.,Inc. in 2001

Current (as of 2013)

- Magnevation SpeakJet (www.speechchips.com) TTS256 Hobby and experimenter.
- Epson S1V30120F01A100 (www.epson.com) IC DECTalk Based voice, Robotic, Eng/Spanish
- Textspeak TTS-EM (www.textspeak.com) ICs, Modules and Industrial enclosures in 24 languages. Human sounding, Phoneme based.

Mattel

The Mattel Intellivision game console, which is a computer that lacks a keyboard, offered the Intellvoice Voice Synthesis module in 1982. It included the SP0256 Narrator speech synthesizer chip on a removable cartridge. The Narrator had 2kB of Read-Only Memory (ROM), and this was utilized to store a database of generic words that could be combined to make phrases in Intellivision games. Since the Orator chip could also accept speech data from external memory, any additional words or phrases needed could be stored inside the cartridge itself. The data consisted of strings of analog-filter coefficients to modify the behavior of the chip's synthetic vocal-tract model, rather than simple digitized samples.

SAM

Also released in 1982, Software automatic mouth was the first commercial all-software voice synthesis program. It was later used as the basis for Macintalk. The program was available for non-Macintosh Apple computers (including the Apple II, and the Lisa), various Atari models and the Commodore 64. The Apple version preferred additional hardware that contained DACs, although it could instead use the computer's one-bit audio output (with the addition of much distortion) if the card was not present. The Atari made use of the embedded POKEY audio chip. Speech playback on the Atari normally disabled interrupt requests and shut down the ANTIC chip during vocal output. The audible output is extremely distorted speech when the screen is on. The Commodore 64 made use of the 64's embedded SID audio chip.

Atari

Arguably, the first speech system integrated into an operating system was the 1400XL/1450XL personal computers designed by Atari,Inc. using the Votrax SC01 chip in 1983. The 1400XL/1450XL computers used a Finite State Machine to enable World English Spelling text-to-speech synthesis.Unfortunately, the 1400XL/1450XL personal computers never shipped in quantity.

The Atari ST computers were sold with "stspeech.tos" on floppy disk.

Apple

The first speech system integrated into an operating system that shipped in quantity was Apple computer's MacInTalk. The software was licensed from 3rd party developers Joseph Katz and Mark Barton (later, SoftVoice, Inc.) and an early version was featured during the 1984 introduction of the Macintosh computer. This January demo, which used speech synthesis based on the Software Automatic Mouth, or SAM software, required 512 kilobytes of RAM memory. As a result, it could not run in the 128 kilobytes of RAM the first Mac actually shipped with.So, the demo was accomplished with a prototype 512k Mac, although those in attendance were not told of this and the synthesis demo created considerable excitement for the Macintosh. In the early 1990s Apple expanded its capabilities offering system wide text-to-speech support. With the introduction of faster PowerPC-based computers they included higher quality voice sampling. Apple also introduced sppech recognition into its systems which provided a fluid command set. More recently, Apple has added sample-based voices. Starting as a curiosity, the speech system of Apple Macintosh has evolved into a fully supported program, plainTalk, for people with vision problems. Voiceover was for the first time featured in Mac OS X Tiger (10.4). During 10.4 (Tiger) & first releases of 10.5 (Leopard) there was only one standard voice shipping with Mac OS X. Starting with 10.6 (Snow Leopard), the user can choose out of a wide range list of multiple voices. VoiceOver voices feature the taking of realistic-sounding breaths between sentences, as well as improved clarity at high read rates over PlainTalk. Mac OS X also includes say, command-line based application that converts text to audible speech. The AppleScript Standard Additions includes a say verb that allows a script to use any of the installed voices and to control the pitch, speaking rate and modulation of the spoken text.

The Apple iOS operating system used on the iPhone, iPad and iPod Touch uses Voice over speech synthesis for accessibility.Some third party applications also provide speech synthesis to facilitate navigating, reading web pages or translating text.

AmigaOS

The second operating system to feature advanced speech synthesis capabilities was AmigaOS, introduced in 1985. The voice synthesis was licensed by Commodore International from SoftVoice, Inc., who also developed the original MacinTalk text-to-speech system. It featured a complete system of voice emulation for American English, with both male and female voices and "stress" indicator markers, made possible through the Amiga's audio chipset.The synthesis system was divided into a narrator device, which was responsible for modulating and concatenating phonemes, and a translator library which translated English text to phonemes via a set of rules. AmigaOS also featured a high-level "Speak Handler", which allowed command-line users to redirect text output to speech. Speech synthesis was occasionally used in third-party programs, particularly word processors and educational software. The synthesis software remained largely unchanged from the first AmigaOS release and Commodore eventually removed speech synthesis support from AmigaOS 2.1 onward.

Despite the American English phoneme limitation, an unofficial version with multilingual speech synthesis was developed. This made use of an enhanced version of the translator library which could translate a number of languages, given a set of rules for each language.

Microsoft Windows

Modern Windows desktop systems can use SAPI 4 and SAPI 5 components to support speech synthesis and speech recognition. SAPI 4.0 was available as an optional add-on for Windows 95 and Windows 98. Windows 2000 added Narrator, a text–to–speech utility for people who have visual handicaps. Third-party programs such as CoolSpeech, Textaloud and Ultra Hal can perform various text-to-speech tasks such as reading text aloud from a specified website, email account, text document, the Windows clipboard, the user's keyboard typing, etc. Not all programs can use speech synthesis directly.[Some programs can use plug-ins, extensions or add-ons to read text aloud. Third-party programs are available that can read text from the system clipboard.

Microsft Speech Server is a server-based package for voice synthesis and recognition. It is designed for network use with web applications and call centers.

Text-to-Speech (**TTS**) refers to the ability of computers to read text aloud. A **TTS Engine** converts written text to a phonemic representation, then converts the phonemic representation to waveforms that can be output as sound. TTS engines with different languages, dialects and specialized vocabularies are available through third-party publishers.

Android

Version 1.6 of Android added support for speech synthesis (TTS).

Internet

Currently, there are a number of applications, plugins and gadgets that can read messages directly from an e-mail clinet and web pages from a web server or Google Toolbar such as Text-To-Voice which is an add-on to Firefox. Some specialized software can narrate RSS-feeds. On one hand, online RSS-narrators simplify information delivery by allowing users to listen to their favourite news sources and to convert them to podcasts. On the other hand, on-line RSS-readers are available on almost any PC connected to the Internet. Users can download generated audio files to portable devices, e.g. with a help of podcast receiver, and listen to them while walking, jogging or commuting to work.

A growing field in Internet based TTS is web-based assistive technology e.g. 'Browsealoud' from a UK company and Readspeaker. It can deliver TTS functionality to anyone (for reasons of accessibility, convenience, entertainment or information) with access to a web browser. The non-profit project Pediaphon was created in 2006 to provide a similar web-based TTS interface to the Wikipedia.

Other work is being done in the context of the W3C through the W3C Audio Incubator Groupwith the involvement of The BBC and Google Inc

14.APPLICATIONS

Each and every synthesizer is the result of a particular and original imitation of the human reading capability, submitted to technological and imaginative constraints that are characteristic of the time of its creation. The concept of high quality TTS synthesis appeared in the mid eighties, as a result of important developments in speech synthesis and natural language processing techniques, mostly due to the emergence of new technologies (Digital Signal and Logical Inference Processors). It is now a must for the speech products family expansion. Potential applications of High Quality TTS Systems are indeed numerous. Here are some examples:

- **Telecommunications services.** TTS systems make it possible to access textual information over the telephone. Knowing that about 70 % of the telephone calls actually require very little interactivity, such a prospect is worth being considered. Texts might range from simple messages, such as local cultural events not to miss (cinemas, theatres), to huge databases which can hardly be read and stored as digitized speech. Queries to such information retrieval systems could be put through the user's voice (with the help of a speech recognizer), or through the telephone keyboard (with DTMF systems). One could even imagine that our (artificially) intelligent machines could speed up the query when needed, by providing lists of keywords, or even summaries. In this connection, AT&T has recently organized a series of consumer tests for some promising telephone services. They include: Who's Calling (get the spoken name of your caller before being connected and hang up to avoid the call), Integrated Messaging (have your electronic mail or facsimiles being automatically read over the telephone), Telephone Relay Service (have a telephone conversation with speech or hearing impaired persons thanks to ad hoc text-to-voice and voice-to-text conversion), and Automated Caller Name and Address (a computerized version of the "reverse directory"). These applications have proved acceptable, and even popular, provided the intelligibility of the synthetic utterances was high enough. Naturalness was not a major issue in most cases.

- **Language education**. High Quality TTS synthesis can be coupled with a Computer Aided Learning system, and provide a helpful tool to learn a new language. To our knowledge, this has not been done yet, given the relatively poor quality available with commercial systems, as opposed to the critical requirements of such tasks.

- **Aid to handicapped persons.** Voice handicaps originate in mental or motor/sensation disorders. Machines can be an invaluable support in the latter case: with the help of an especially designed keyboard and a fast sentence assembling program, synthetic speech can be produced in a few seconds to remedy these impediments. strophysician Stephen Hawking gives all his lectures in this way. The aforementioned Telephone Relay Service is another example. Blind people also widely benefit from TTS systems, when coupled with Optical Recognition Systems (OCR), which give them access to written information. The market for speech synthesis for blind users of personal computers will soon be invaded by mass-market synthesizers bundled with sound cards.

- **Talking books and toys**. The toy market has already been touched by speech synthesis. Many speaking toys have appeared, under the impulse of the innovative 'Magic Spell' from Texas Instruments. The poor quality available inevitably restrains the educational ambition of such products. High Quality synthesis at affordable prices might well change this.

- **Vocal Monitoring.** In some cases, oral information is more efficient than written messages. The appeal is stronger, while the attention may still focus on other visual sources of information. Hence the idea of incorporating speech synthesizers in measurement or control systems.

- **Multimedia, man-machine communication**. In the long run, the development of high quality TTS systems is a necessary step (as is the enhancement of speech recognizers) towards more complete means of communication between men and computers. Multimedia is a first but promising move in this direction.

- **Fundamental and applied research**. TTS synthesizers possess a very peculiar feature which makes them wonderful laboratory tools for linguists: they are completely under control, so that repeated experiences provide identical results (as is hardly the case with human beings). Consequently, they allow to investigate the efficiency of into native and rhythmic models. A particular type of TTS systems, which are based on a description of the vocal tract through its resonant frequencies (its formants) and denoted as formant synthesizers, has also been extensively used by phoneticians to study speech in terms of acoustical rules. In this manner, for instance, articulatory constraints have been enlightened and formally described.

15.RESULTS AND FINDINGS

CODE

CODE:

```
% fclose(modem);
%delete(modem);
clc;
clear all;
%initialising Serial Port and Voice
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

ha = actxserver('SAPI.SpVoice');

text1= 'Please wait ... the system is initializing';
disp(text1);
invoke(ha,'speak',text1);

modem = serial('COM3');
set(modem,'BaudRate',9600);
set(modem,'DataBits',8);
set(modem,'Parity','none');
set(modem,'StopBits',1);
set(modem,'FlowControl','none');
fopen(modem);
text2= 'Initializing is over..';
disp(text2);
invoke(ha,'speak',text2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%

%initialising GSM Modem
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
text3= 'Initializing GSM Modem now';
disp(text3);
invoke(ha,'speak',text3);

testcondition = 0;
while testcondition==0   %continue this loop till gsm is responding with AT Commands

   fprintf(modem,'%s\r\n','at');
   pause(10);
   received = fgetl(modem)
```

```
    for i=1:length(received)
        if (received(1,i)=='O')
            if (received(1,(i+1))=='K')
                testcondition = 1
                break;
            end
        end

    end
    if(testcondition == 1)
        disp('GSM is responding with AT command....');
        invoke(ha,'speak','GSM is responding with AT command....');
    else
        disp('GSM is not responding with AT command....');
        invoke(ha,'speak','GSM is not noresponding with AT command....');

    end
    pause(1);

end
testcondition = 0
while 0== testcondition
    fprintf(modem,'%s\r\n','ate0');
    pause(1);
    received = fgetl(modem)



    for i=1:length(received)
        if (received(1,i)=='O')
            if (received(1,(i+1))=='K')
                testcondition = 1
                break;
            end
        end

    end
    if(1 == testcondition )
        disp('echoing is made off');
        invoke(ha,'speak','echoing is made off');
    else
        disp('echoing is not made off');
        invoke(ha,'speak','echoing is not made off');

    end
    pause(1);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%
testcondition = 0
while 0== testcondition
    fprintf(modem,'%s\r\n','AT&W');
    pause(1);
    received = fgetl(modem)



    for i=1:length(received)
        if (received(1,i)=='O')
            if (received(1,(i+1))=='K')
                testcondition = 1
                break;
            end
        end

    end

    if(1 == testcondition )
        disp('AT&W OK');
        invoke(ha,'speak','AT&W OK');
    else
        disp('AT&W NOT OK');
        invoke(ha,'speak','AT&W NOT OK');

    end
    pause(1);
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%
testcondition = 0
while 0== testcondition
   fprintf(modem,'%s\r\n','AT+CMGF=1');
   pause(1);
   received = fgetl(modem)



   for i=1:length(received)
      if (received(1,i)=='O')
         if (received(1,(i+1))=='K')
            testcondition = 1
            break;
         end
      end

   end

   if(1 == testcondition )
      disp('Message Format is SET');
      invoke(ha,'speak','Message Format is SET');
   else

      disp(' Message Format is NOT SET');
      invoke(ha,'speak',' Message Format is NOT SET');

   end
   pause(1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%
testcondition = 0
```

```
while 0== testcondition
    fprintf(modem,'%s\r\n','AT+CNMI=2,2,0,0,0');
    pause(1);
    received = fgetl(modem)



    for i=1:length(received)
        if (received(1,i)=='O')
            if (received(1,(i+1))=='K')
                testcondition = 1
                break;
            end
        end

    end

    if(1 == testcondition )
        disp('Incoming Message Format is SET');
        invoke(ha,'speak','Incoming Message Format is SET');
    else
        disp('Incoming Message Format is NOT SET');
        invoke(ha,'speak','Incoming Message Format is NOT SET');

    end
    pause(1);
end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%

text4 = 'Initializing GSM Modem is over';
disp(text4);
```

```
invoke(ha,'speak',text4);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
modem.TimeOut = 100;

    received = fgetl(modem)
    received = fgetl(modem)
    received = fgetl(modem)
    received = fgetl(modem)
    received = fgetl(modem)

invoke(ha,'speak','Send messages now to test this module:');


while 1
    readingover = 0;
    received = fgetl(modem)

     i=1;
     validity = size(received);
     if(validity)
        if (received(1,i)=='+')
            if (received(1,(i+1))=='C')
                if(received(1,(i+2))=='M')
                     if(received(1,(i+3))=='T')
                                                         invoke(ha,'speak','You got a message from ');
                                pause(1);
                                for j=i+10:i+19
                                    invoke(ha,'speak',received(1,j));


                                end
                                 pause(1);
                                 invoke(ha,'speak','Here is your message:');
                                 pause(1);

                                 received = fgetl(modem)
                                   validity1 = size(received);
                                    if(validity1)
                                        invoke(ha,'speak',received);
                                   % else
                                       %  invoke(ha,'speak','Empty');

                                    end
                                 readingover = 1;

                                  end

                     end
            end
          end

    if (0 == readingover)
    invoke(ha,'speak',received);
    end
     end

    end
```

16. LIMITATIONS

TASKS GIVEN BY THE COMPANY DURING THE PRACTICE SCHOOL PROGRAM

- ✓ To do some projects like DTMF,MOBILE ROBO.etc.
- ✓ To write the codes of basic projects in keil &arduino
- ✓ To give a Demo Classes on keil, arduino to all Employees in our Company.
- ✓ Train and Deal the concepts of Projects to Final students of JNIT College.
- ✓ Teach Embedded classes for B.TECH Students on behalf of company.

17. DISCUSSIONS

ACHIEVEMENTS INCLUDES:

- ✓ Embedded systems fundamentals Training that includes Basics.
- ✓ Obtain Knowledge on keil and arduino Overview.
- ✓ Developed some projects using keil and arduino softwares
- ✓ Demos on Conditional Statements (if-else, do-while, for, for-each, switch, break, continue).
- ✓ Gained knowledge fully on Emedded systems
- ✓ Projects Development using Keil and Arduino
- ✓ Train the students on Embedded systems
- ✓ To Do Real time Project on the Blind people message reader

TARGETS REACHED

- ✓ Designed Project with various Practical examples Mobile operated robot,DTMF
- ✓ Trained B.TECH students to get knowledge over their main projects.
- ✓ Conducted a 20 day workshop on Embedded systems in JNIT College
- ✓ Have done a Research Paper and published it on International Journal.
- ✓ Done a real-time project on Embedded systems.

18.CONCLUSION

This project focuses mainly on how blind people will know the messages by using GSM module, Laptop. This project is similar to text to speech conversion. In this text to speech conversion blind people who have lost their eyes due to some incidents. They can hear the messages through this. The message which is came to the GSM module is serially sent to the laptop and a code is written to it to hear. The code is written in such a manner that the message which is sent to the laptop serially is converted to speech and it can be heard by the people who are blind. The messages received to gsm module will be sent serially to laptop and using the code the audio will be heard and the blind people can know the messages which is the best possible solution.

19.BIBILOGRAPHY

- Kurzweil, Raymond (2005). The Singularity is Near. Penguin Books. ISBN 0-14-303788-9.
- Klatt, D. (1987) "Review of Text-to-Speech Conversion for English" Journal of the Acoustical Society of America **82**(3):737-93
- Lambert, Bruce (March 21, 1992). "Louis Gerstman, 61, a Specialist In Speech Disorders and Processes". New York Times.
- Arthur C. Clarke Biography at the Wayback Machine(archived December 11, 1997)
- Where "HAL" First Spoke (Bell Labs Speech Synthesis website)"Bell Labs. Retrieved 2010-02-17.
- Anthropomorphic Talking Robot Waseda-Talker Series
- TSI Speech+ & other speaking calculators
- History and Development of Speech Synthesis, Helsinki University of Technology, Retrieved on November 4, 2006
- van Santen, Jan P. H.; Sproat, Richard W.; Olive, Joseph P.; Hirschberg, Julia (1997). Progress in Speech Synthesis. Springer
- Mattingly, Ignatius G. (1974). "Speech synthesis for phonetic and phonological models". In Sebeok, Thomas A. Current Trends in Linguistics (Mouton, The Hague) **12**: 2451–2487.
- "Speech synthesis". World Wide Web Organization.
- Wikipedia
- L.F. Lamel, J.L. Gauvain, B. Prouts, C. Bouhier, R. Boesch. Generation and Synthesis of Broadcast Messages,Proceedings ESCA-NATO Workshop and Applications of Speech Technology, September 1993.