Project EPR400/402 **First semester report**	**August 2022**	**Note:**

Student to complete this section					
Ramedies	**I.**	**Mr.**	**16023405**		
Design of a levitation train			**Study leader:** **Mr. A. Oloo**		
Degree programme enrolled for: Computer Engineering	**Project number:**	A05	Revision number:	0	
Student declaration: I understand what plagiarism is and that I have to complete my project on my own. I am fully aware of the University's policy in this regard. I have not used another student's past or present written work to submit as my own. I declare that this report is my own original work. Where other people's work has been used (either from a printed source, internet or any other source), this has been properly acknowledged and referenced in accordance with departmental requirements.	________________ Student signature	5 August 2022 Date			

Declaration by language editor	
I have been allowed adequate time to read this document carefully and to make corrections where necessary (date received indicated below). To the best of my knowledge, correct formatting, spelling and grammar are used throughout the document.	
______________________________ N. Ramedies (language editor)	5 August 2022 Date

Table of contents

1. Literature study

Text-to-speech (TTS) synthesis is useful in many applications, particularly for aiding people to hear text being read, e.g., in situations of partial blindness. In more modern applications of speech synthesis, it is used for providing accessibility to speech among other applications. To accomplish a system that is capable of TTS synthesis, a combination of natural language processing (NLP) and a linguistics analysis has to be performed on the received text. In addition to this, digital signal processing (DSP) should be performed, to provide audio waveforms from signals. The general format of a TTS synthesizer thus comprises of the main functional units illustrated in Figure 1. Often in TTS systems, the NLP function is referred to as the front-end and the DSP function is referred to as the back-end of the system.

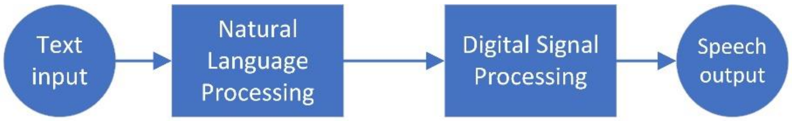


Figure 1 High level functional block diagram of TTS system

Natural language processing is a fundamental, key component of text-to-speech synthesis. The front-end development of TTS systems involves NLP, with the objective of attaching prosody or sound information to the input text. Zhigang provides information on how the components involved in the front-end development work together, to provide the necessary output to the DSP unit [1]. The paper also provides information on DSP and non-DSP techniques, for generating signal waveforms from the derived prosody information. Reichel describes in more detail some text preprocessing modules associated with NLP for TTS synthesis [2]. These modules are defined as text normalization, part-of-speech tagging, grapheme-to-phoneme conversion and word stress. The paper provides further detail concerning the lower-level implementation of these modules. An important consideration, is that most of these implementations are data-driven and thus bring up the need for dictionaries or lexicons to be used.

The aforementioned research papers do not explicitly mention linguistics analysis techniques, but a modified implementation. In general, linguistics analysis techniques that should be avoided for this system implementation, are; rules-based and neural network implementations, as they are data-driven and time consuming. In the book; "*Theory and applications of digital speech processing*" [3], a better insight to linguistics analysis and the desired output is provided. It also provides various TTS and DSP knowledge, along with a few design choices to consider.

The back-end of a TTS synthesizer, involves DSP of the prosody information, generated from the front-end of the system, to produce synthesized text. Similarly to NLP, one could consider implementing the back-end, using the two approaches briefly mentioned [1], i.e., the data-driven- and the rule-driven approach. The former usually makes use of synthesis methods, such as concatenation, unit selection, Hidden Markov Models (HMM) and deep neural networks (DNN). The data-driven synthesizers do not make use of any digital signal processing and are not flexible. They also require a lot of storage space and most importantly, data. Rule-driven approaches often make use of articulatory synthesis or formant synthesis. These synthesizers make use of DSP techniques and while they produce a synthetic speech, they are both flexible solutions and are based on derived synthesis models. Articulatory synthesis is a direct simulation of the physical process of human pronunciation, while formant synthesis, is an acoustic simulation. Articulatory synthesis also presents a lot of difficulty in controlling parameters, such as prosodic features, which is undesirable, therefore, a formant synthesizer might provide the most adequate solution for the system's needs.

In 1979, Dennis H. Klatt published a report on formant synthesis containing a realized speech synthesizer, in order to provide a flexible research tool, for studying aspects of speech [4]. This paper demonstrates the result of using digital resonators in parallel or cascade configurations, while allowing the user to specify variable control parameter data. The paper also contains synthesizer design descriptions, motivations, computer requirements and strategies for imitating speech utterances. The paper thus provides a good baseline expectancy, when considering a formant synthesizer for speech output. A group of researchers published a document, describing an implementation of a TTS system, based on the Klatt synthesizer model, which was modified for the Portuguese language [5]. It also uses a model based on *Acoustic Theory of Speech Production*, developed by Fant [6]. The model of the source is the LF model described by Fant. The paper provides knowledge on a combination of acoustics and synthesis, in the form of a source, tract, and graphical interface, to produce prototype speech.

The intercom system to be designed and implemented will make use of the theories and models provided from past sources. Text input will be preprocessed, to generate prosody, by implementing a general NLP unit. The procedure will consist of; text normalization, part of speech tagging, and grapheme to phoneme conversion. The difference between the proposed implementation and the previous ones, is that it will not be a data-driven design, or a rule-based implementation. The implementation would consist of sentence parsing, which is more algorithmic based and requires access to dictionaries and lexicons. An addition to the NLP unit will introduce a text-based, profanity filtering system. The system will simply remove explicit profanity by comparing with a self-defined database. The system will also include a feature that will extract text from emails and thus, an email client will receive the input for the NLP unit.

TTS synthesis will be implemented, using the traditional DSP approach. The approach considered, is the formant synthesizer produced by Klatt. The reasoning is, that the proposed system will be designed and be flexible to imitate the frequency spectrogram of several individuals' voices to be used in a teaching environment. Data collection and storage will also be an issue, if a DSP approach is not taken. The difference between previous approaches and the intended approach is that the source is not modelled on an LF model, but instead, it is modelled on an individual's voice. However, acoustics theory on speech production may assist in extracting formant features.

3. Work breakdown and first semester progress

Table 1 contains the work breakdown of the tasks to be completed and the progress of these tasks as of the 25th July 2022.

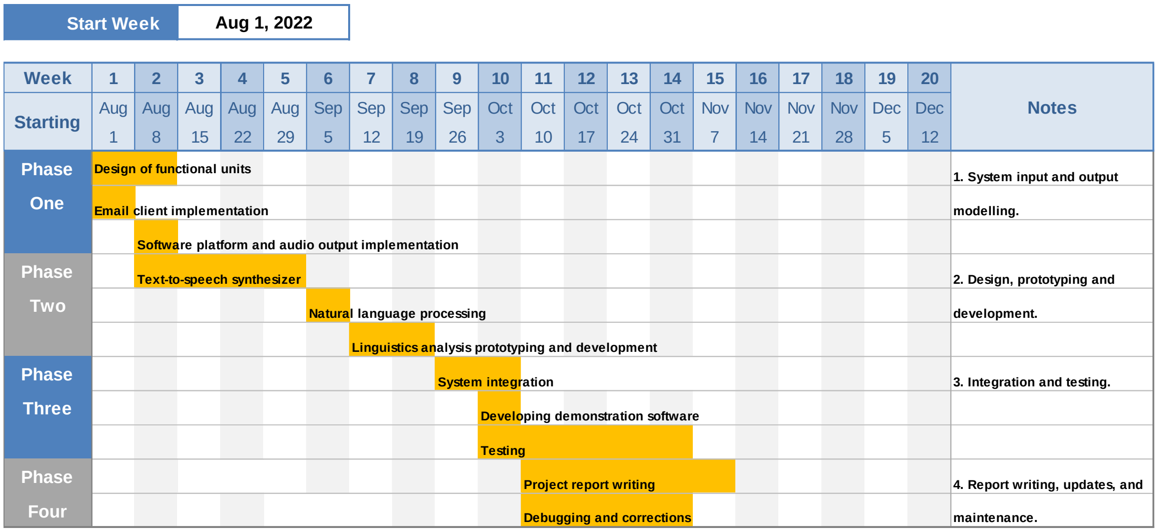
Task	**Progress**	**Brief description**
Preparation of Project Proposal	100% complete	Revision 2 is approved.
System concept design.	100% complete	The system overall design has been researched and concept algorithms have been decided on.
Design and simulate preliminary email client.	50% complete	Email client base has been implemented and now software specific requirements are being catered for.
Email client transfer to hardware platform.	0%	Still setting up hardware platform to begin software integration process.
Email client display feature.	0%	Design of the function is yet to be decided on.
Text-based profanity filtering system.	10%	Dictionaries have been created; further development of the filter requires a fully functioning Natural Language Processing (NLP) feature in place.
Text-to-speech synthesis.	5%	References for text-to-speech conversion has been found. Currently signal outputting as audio files and audio file concatenation is being tested. Audio recording of vowel formants will then be conducted.
Natural language processing.	5%	A dictionary was found that contains words with no phonemes. Currently phoneme dictionaries are being looked for as well as part of speech references. The tokenization process may then begin.
Linguistics analysis technique	5%	A linguistics analysis technique has been decided on. Linguistics analysis programming will take place after NLP is complete and a dictionary is found.
Hardware integration.	100% complete	Drivers for a WiFi module has been successfully installed on the system and it has internet connection. Audio drivers are also successfully installed. The hardware is capable of input through the internet and output through USB speakers.
System software set up	50% complete	An operating system has been

		installed on the hardware platform. It has been updated and has a working version of the programming language installed. IDE installation is currently being looked into. Once a simple code can be run, software can be developed on a separate system.
Software integration	0%	Software functions not yet fully developed.
Audio recording	10%	The software for recording has been installed and tested with arbitrary audio. Audio recordings will commence when TTS synthesis begins.
Multiple user capability	0%	After complete design of a single system, the system will attempt to include other users.
Testing and debugging	0%	
Writing of final report	0%	

Table 1 Work breakdown and Progress

3. Project plan (second semester)

Figure 2 is the proposed project plan schedule of the completion of the system for the duration of the project.



Start Week	Aug 1, 2022

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Notes
Starting	Aug 1	Aug 8	Aug 15	Aug 22	Aug 29	Sep 5	Sep 12	Sep 19	Sep 26	Oct 3	Oct 10	Oct 17	Oct 24	Oct 31	Nov 7	Nov 14	Nov 21	Nov 28	Dec 5	Dec 12	
Phase One	Design of functional units	■																			1. System input and output modelling.
	Email client implementation																				
		Software platform and audio output implementation																			
Phase Two		Text-to-speech synthesizer	■	■	■																2. Design, prototyping and development.
						Natural language processing															
							Linguistics analysis prototyping and development	■													
Phase Three									System integration	■											3. Integration and testing.
										Developing demonstration software											
										Testing	■	■	■	■							
Phase Four											Project report writing	■	■	■	■						4. Report writing, updates, and maintenance.
											Debugging and corrections	■	■	■							

Figure 2 Project Plan schedule

4. References

[1] Y. Zhigang, “An overview of speech synthesis technology,” in *Eighth International Conference on Instrumentation and Measurement, Computer, Communication and Control*, Beijing, China, 2018.

[2] U. D. Reichel and H. R. Pfitzinger, “Text Preprocessing for Speech Synthesis,” in *TC-Star Speech to Speech Translation Workshop*, Munich, Germany, 2006.

[3] L. R. Rabiner and R. W. Schafer, Theory and Applications of Digital Speech Processing - First Edition, Santa Barbara: Pearson Higher Education, 2011.

[4] D. H. Klatt, “Software for a cascade/parallel formant synthesizer,” Massachusetts Institute of Technology, Cambridge, Massachusetts, 1979.

[5] L. M. T. Jesus, F. A. C. Vaz and J. C. Principe, “An Implementation of the Klatt Speech Synthesizer,” *Revista Do Detua,* vol. 2, no. 1, 1997.

[6] G. Fant, Acoustic Theory of Speech Production, 1960.