Speech-controlled robot

Physical Computing
Project
Group 7E: Beatrix Zöllner, Jaspreet Singh, Tsungai Chipato

Table of Contents

Introduction and Motivation

We shall build a robot that can recognize speech. The robot that we shall be using for our project is the LEGO Mindstorm NXT Robot. We shall use a microphone to record our voice in order to take samples. In addition, we shall write MATLAB code that shall be based on recognising our saved samples shall will be able to process and evaluate our samples. Following the evaluation of the samples,the user shall now be able to send commands to the robot from the microphone.

The code shall be written in such a way that it will take input from the microphone as an audio signal. It shall identify four voice commands “stop”, “left”, “right” and “forward” spoken by the user. The program shall filter all of samples and user input whilst removing the silence from all our recordings to achieve the most accurate recording. If signal correlates with the samples (i.e. one of the words we sampled is said) then the LEGO Mindstorms NXT Robot shall move like the command says.
We aim to make this a system that can recognise a person's speech by their voice “print” so that this system is specific to that user. In this way this technology could be used worldwide and is not confined to one country or language.

This implementation can be used as a result to be a functional voice recognition wheelchair. However, the basis of this can be used as a simple voice recognition system and can be used for other functionalities that could require voice recognition for example a smart home with its functionalities like switching on lights, based on voice recognition.

The outcome of our project is build a system that can aid physically handicapped people. In addition, this system could also be used in cases whereby the user could have been injured or are ageing and will need extra assistance to move freely. Particularly in the case of quadriplegic persons and whereby they are paralyzed from the neck downwards. This type of system will be useful for them as it will help these persons to not be as reliant on others helping them, and gain independence from that without them having to use.

Who and how will use it?
Our program (software) in our system is capable of recognising the speech of a specific person correctly and the robot will correspond according to the command.
In real life: The system could be implemented in a speech-controlled wheelchair, for instance. This system could thus be used by a person who is not able to move anymore and therefore has to sit in a wheelchair. This wheelchair could be controlled by the voice of that person only.

A project plan with a schedule with a task division and a risk analysis

Date	**Activity**
Monday, 27 November 2017	(Jaspreet) Literature study.
Wednesday, 29 November 2017	(All) Collecting data by taking samples
Thursday, 30 November 2017	(All) Collecting data by taking samples and trying to find a way to recognize speech using the recorded samples.

Wednesday, 06 December 2017	(Tsungai and Beatrix) Using mel cepstrum coefficients to distinguish between the different words. Also, a while-loop to test if the function (we have so far) works. We found the melcepstrum of fast. (Jaspreet) Updating document.
Thursday, 07 December 2017	(All) We found the melcepst of stop and right and go the voice recognition for stop, fast and right. Then, we aligned the waves to start at the same moment in time.
Tuesday, 12 December 2017	(Jaspreet) Worked on the document. (Tsungai and Beatrix) Worked on improving the code.
Wednesday, 13 December 2017	(All) We tested whether we could control the robot by speech.
Thursday, 14 December 2017	(All) Presentation and demo. (All) Worked on the final document
Friday, 15 December 2017	(All) worked on the final document

Task division

Our tasks were divided as follows:
Jaspreet was in charge of updating the tasks that we had completed.
Beatrix and Tsungai were in change of creating the presentation.

Risk analysis

Week 1	Taking samples and finding features to distinguish between different commands.
Week 2	Writing a while-loop to test if the system can recognize commands and writing code for the robot.
Week 3	Test the robot and do the presentation.

Literature Study

The following will be research papers that we discovered that involve speech recognition.We found three papers whereby the researchers from the papers use different techniques in order to classify and distinguish different types of speech.Our project had similar ideas to these research papers, However, our implementation was a simplified version.

The first paper is written by Masato Nishimori, Takeshi Saitoh and Ryosuke Konishi from Kagawa University, Japan. They designed a system for a speech-controlled wheelchair. The system can recognize five basic reaction commands, four short moving reaction commands and two verification commands. The verification commands are used in order to avoid the wheelchair from moving the wrong way. The system makes of use a headset microphone and a laptop. To recognize the words, a grammar-based recognition parser named “Julian” is used. The recognition performance of Julian was tested to test the system. The recognition rate for reaction commands was 98,3% and it was 97,0% for verification commands.[1]

The second article is written by Yasir Ali Memon, Imaaduddin Motan, Muhammad Ali Akbar, Sarmad Hameed, Moez Ul Hasan from Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Pakistan. They designed a system for a robot that is also controlled by speech. The system can recognize ten commands. The Android app they designed to recognize speech records speech and it sends the command in text form to the robot. The robot's receiver is able to receive the data which is sent by the app, using Bluetooth connection. Then, the text is to the robot character by character. Thereafter, a will use the characters from a single word.[2][3]

The third article is written by Megha Muralidharan, P. T. Jabir, Vinod Pottakulath from MES College of Engineering Kuttippuram, India. They designed a system that uses a voice recognition kit with fifteen pieces in a module. Those fifteen pieces are divided into three groups with five pieces in each group. They first train the their module with voice instructions group by group and after that they import 1 group before recognition within that group. They do so for other groups using this technique. They train the module using this. They use a microphone for this input and train the voice recognition kit by recording voice instructions. In addition, to this they also use a joystick with the voice recognitions.[4]

Concept of operations

Product mission statement
As previously stated, our possible stakeholders for our system include people who are handicapped, injured or suffering from the effects of old age. Due to these circumstances the people affected by this are are not able to participate in simple activities like getting around cities or travelling to and from their homes on their own. Owing to this they need to rely on aid from others in order for them to get around. Using this system will help these people to regain their sense of independence as they will not need to rely on others as much. In addition, it will provide an alternative for them to use instead of a more strenuous wheelchair that requires muscle movements**.** So, it provides a wider impact across different types of conditions.
Not only people who are using a speech-controlled wheelchair are stakeholders, but also doctors as they can advise their patients to get such a wheelchair. And the company who is designing these wheelchairs is also a stakeholder.

System overview
In our system we will proceed with the following steps in order to execute a command:
1. Process the signal;
2. Remove silence and use frame blocking;
3. Apply FFT to signal;
4. Create a vector with MCC of the signal;
5. Gain the features of the signal and compare with the database of samples;
6. If the word has been incorrectly identified we get a new signal;

7. If the word has been correctly identified the wheelchair will perform the command;

Description of how we implemented our system
First, the user has to input data in the system. Therefore, we made the `setup` function. We chose a sampling frequency of 44100 Hz and a bit rate of 16 bits. Due to this, we have an excellent quality of the recorded data, but also not too much storage (memory) is needed.

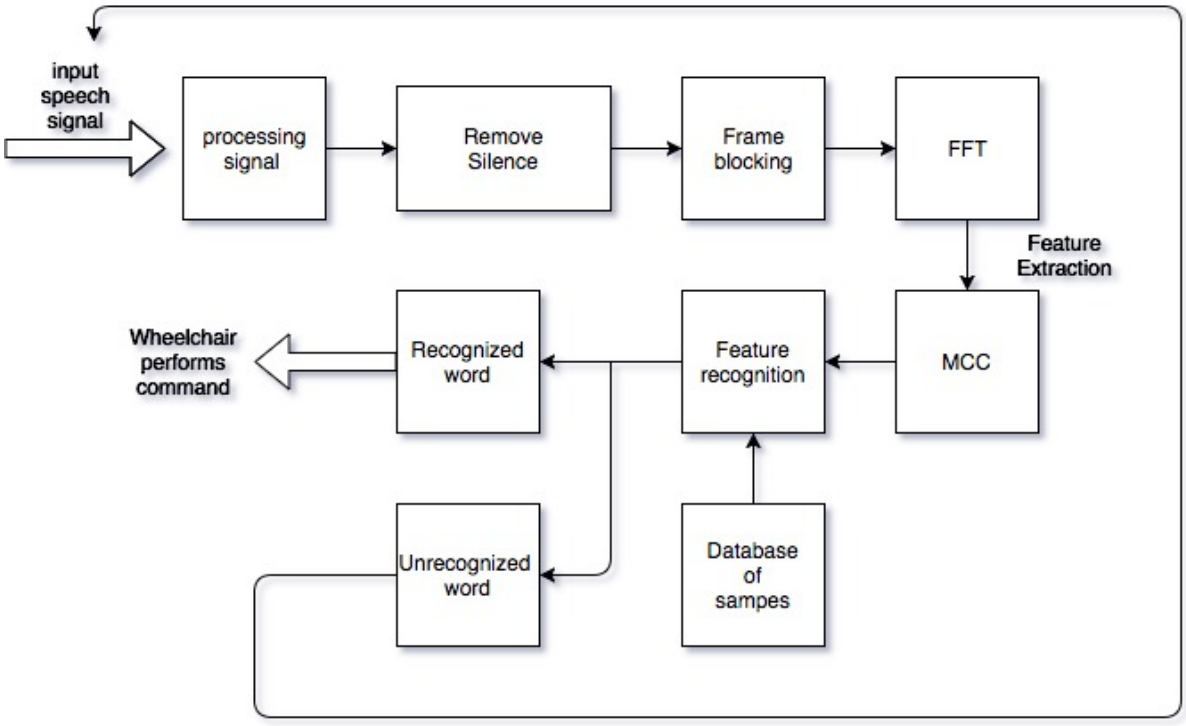
format	Sampling	Bit rate	quality	Size
Wave/Aiff	8,000hz-16,000hz	8	Very Low	Very samll
	16,000-32,000 hz	16	decent	medium
	44,100 Hz	16	excellent	big
	48,000Hz and Above	16 bit -32 bit	pristine	Extremely big

We set the number of samples to four as we have four words we want our system to recognize.

We apply a filter to decrease the background noise and then we divide the signal into frames. We use an threshold of 0.05, every frame with an amplitude higher than this is kept. This way we remove the silent frames an only keep the ones with the voice. Then we apply MFCC to the whole array. Then each sample that is saved. And the configuration of the user was successful.

Next the user can start the program. A loop iterates and checks every few seconds for a signal with similar values. If so the function outputs the recognised word and sends the corresponding instruction to the motor.

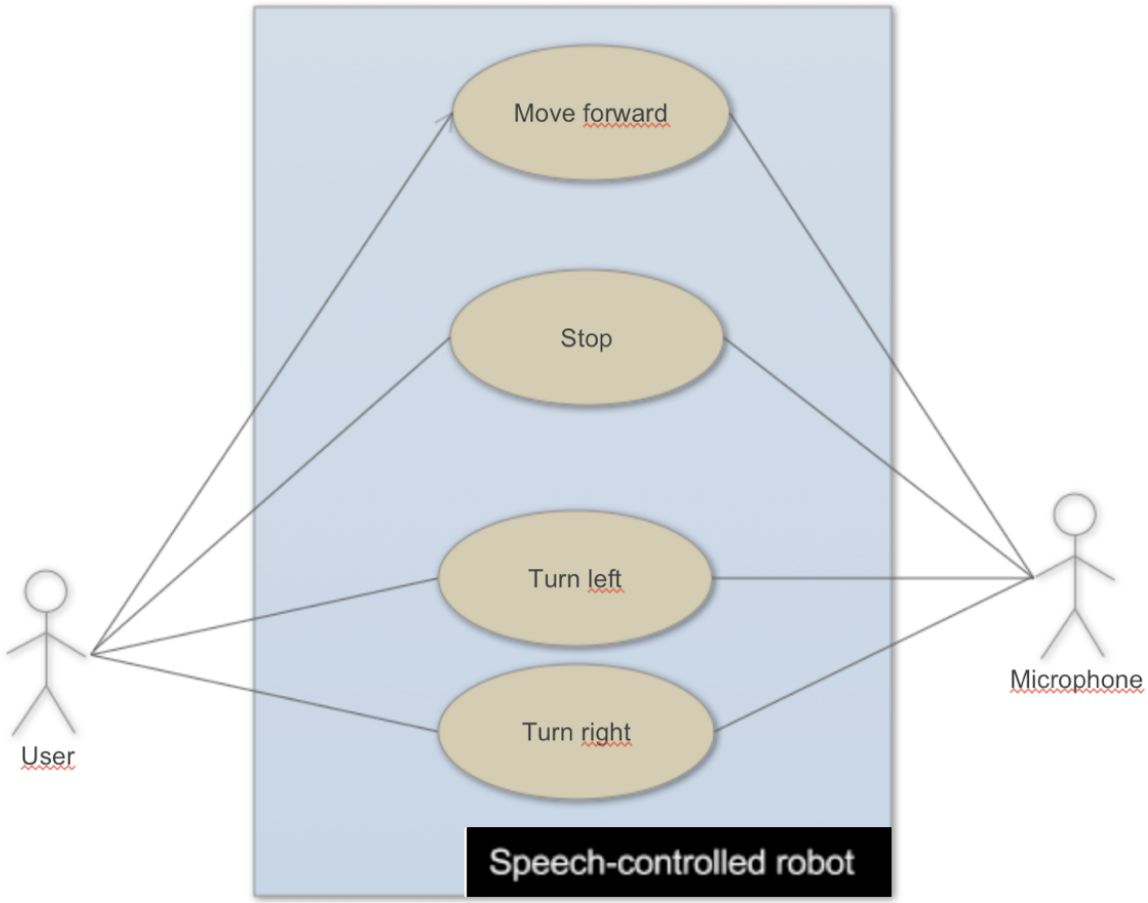
Block diagram



Scenarios

A Quadriplegic man would like to go to the metro or tram stop in order to go and see one of his friends. The wheelchair is active and he may issue different commands to it. The metro or tram station is a few metres away from his home. The man says forward and the wheelchair will move accordingly. The metro or tram station is now to the right of the man, across the street. The man says right and the wheelchair turns right and stops. The man says forward and the wheelchair moves forwards. As the man is now approaching traffic lights, he commands the wheelchair to stop. The wheelchair stops. Once it is clear for the man to proceed, the man says forward and he crosses the street to the metro or tram stop.

Use-case diagram



System requirements

Our system shall recognize spoken commands and will perform the right movements according to the spoken command.

We used the MoSCoW method to distinguish between should haves, must haves, could haves and won't haves:

Requirements	Should have	Must have	Could have	Won't have
Functional	Recognize the following commands: forward, left, right and stop.	Commands that can make the turn instead of going to the left or right the whole time, like: turn left and turn right.	The robot could respond to a command that can make the robot go backwards, like: back. This is very useful in real-life. Or, it could verify the spoken command before executing it in order to avoid accidents. This could also be usefull, because this can avoid accidents from happening.	The system will not have a correction. So, if someone said a command and the system is not sure whtether it is 'valid' to execute it, it can ask if the person using the system actually meant to say a certain word. This is a feature that the search engine Google has when it suggests something if a person made a typing mistake, for instance.

Word	Forward
Description	The robot shall move forward.
Precondition	The person in the wheelchair shall already know the command beforehand.
Post condition	The wheelchair will the move forward and until the next command is received.
Error situations	• The robot receives the classification of the command "forward" wrong and performs another command. • The command cannot be interpreted and the robot does not do anything.
System state in the event of an error	The robot will receive the wrong command and performs the wrong action. It will then wait for a new command.
Actors	User of the robot
Trigger	Command == "forward"

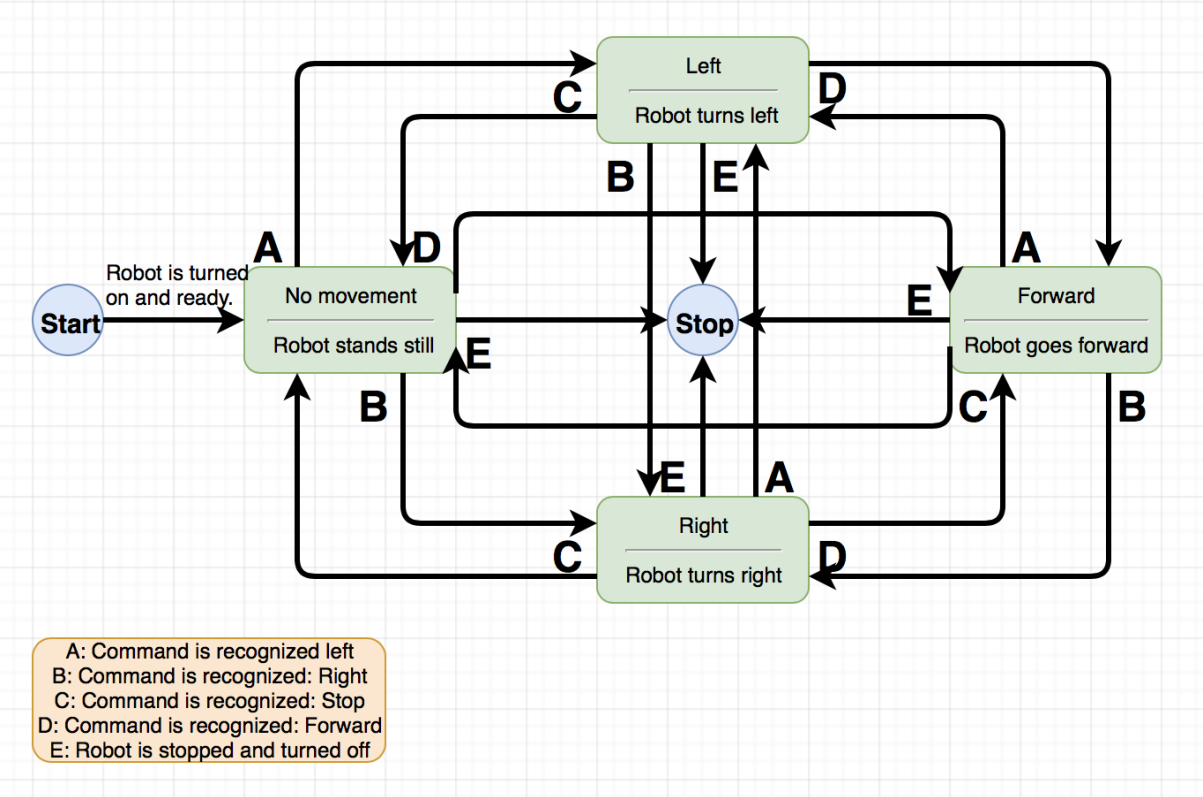
Standard process	• Robot is already active. • User issues command to microphone. • The signal from the microphone is then processed. • Features are extracted from the signal. • Classifier recognizes the word. • Command is sent to the robot. • Robot performs the command
Alternative Process	The robot does not recognize the word and waits for the next instruction.

Word	Stop
Description	The robot shall stop moving.
Precondition	The person in the wheelchair shall already know the command beforehand.
Post condition	The wheelchair will the move forward and until the next command is received.
Error situations	• The robot receives the classification of the command “stop” wrong and performs another command. • The command cannot be interpreted and the robot does not do anything.
System state in the event of an error	The robot will receive the wrong command and performs the wrong action. It will then wait for a new command.
Actors	User of the robot
Trigger	Command == ”stop”
Standard process	• Robot is already active. • User issues command to microphone. • The signal from the microphone is then processed. • Features are extracted from the signal. • Classifier recognizes the word. • Command is sent to the robot. • Robot performs the command
Alternative Process	The robot does not recognize the word and waits for the next instruction.

Word	Right
Description	The wheelchair will move right.
Precondition	The person in the wheelchair shall already know the command beforehand.
Post condition	The wheelchair will the move forward and until the next command is received.
Error situations	• The robot receives the classification of the command “right” wrong and performs another command. • The command cannot be interpreted and the robot does not do anything.
System state in the event of an error	The robot will receive the wrong command and performs the wrong action. It will then wait for a new command.
Actors	User of the robot
Trigger	Command == ”right”

Standard process	• Robot is already active. • User issues command to microphone. • The signal from the microphone is then processed. • Features are extracted from the signal. • Classifier recognizes the word. • Command is sent to the robot. • Robot performs the command
Alternative Process	The robot does not recognize the word and waits for the next instruction.

Word	Left
Description	The wheelchair will move left.
Precondition	The person in the wheelchair shall already know the command beforehand.
Post condition	The wheelchair will the move forward and until the next command is received.
Error situations	• The robot receives the classification of the command “left” wrong and performs another command. • The command cannot be interpreted and the robot does not do anything.
System state in the event of an error	The robot will receive the wrong command and performs the wrong action. It will then wait for a new command.
Actors	User of the robot
Trigger	Command == ”left”
Standard process	• Robot is already active. • User issues command to microphone. • The signal from the microphone is then processed. • Features are extracted from the signal. • Classifier recognizes the word. • Command is sent to the robot. • Robot performs the command
Alternative Process	The robot does not recognize the word and waits for the next instruction.



The letter belongs to the arrow leaving the block.

System design

Components

This is what our system needs:

- A robot;
- A microphone to record commands.

Algorithms

Our algorithm works as follows. We ask the user to take samples of their voice. This is done user-friendly as the software tells the user what to do and when to do it. Then, when the samples have been taken, the user can use the system by using the recognizing software. The software will record the user's voice and try to recognize the commands if the user said on of the commands ("Forward", "Left", "Right" and "Stop"). If the users says something that is not one of the commands or if the user does not say one of the commands properly, the system will ask the user to speak again.
When the user takes samples, the silence is removed between the spoken parts. This makes the recognition of the commands easier. Then, when the user tries out the recognizing program, the silence is also removed. Both samples and voice recording (when using the software) are modified to make recognition of the commands easier.

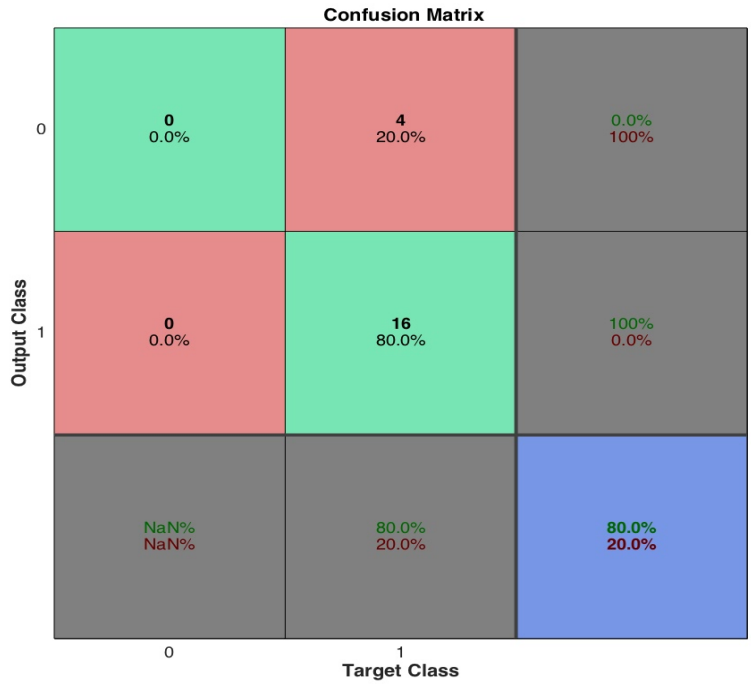
Features

In order to distinguish the differences between the four commands we decided to include as features a row of the mel frequency cepstrum coefficients (MFCC). We put the MFCC values into a vector and compared the MFCC values obtained from the input to the same row with values obtained by running the `setup` function.
The `setup` function is a function which will calibrate on the user's voice. This might turn out really useful and helpful as only the owner and user of, for instance, a speech-controlled wheelchair will be able to control the system. Hence, other people cannot not control the system, or wheelchair.

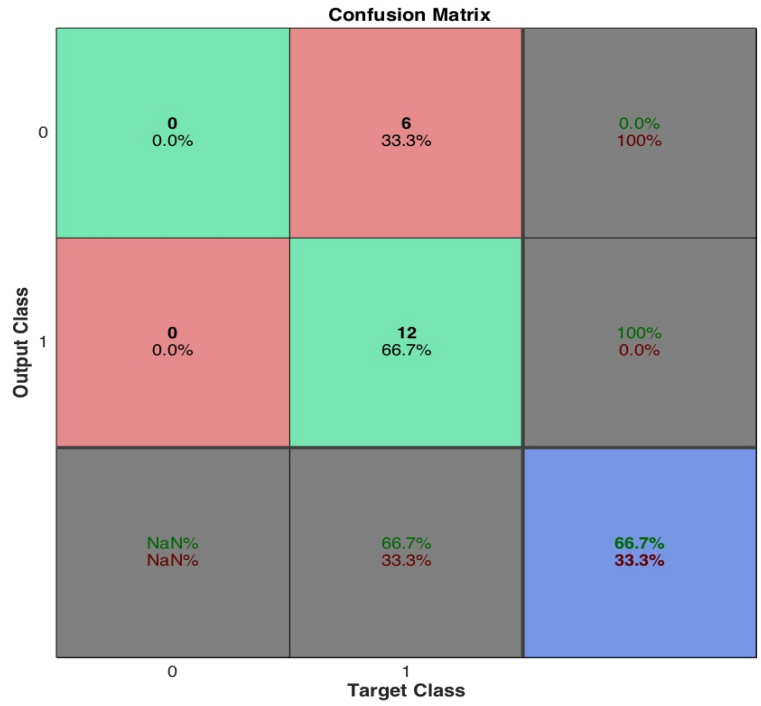
Classifier

Testing

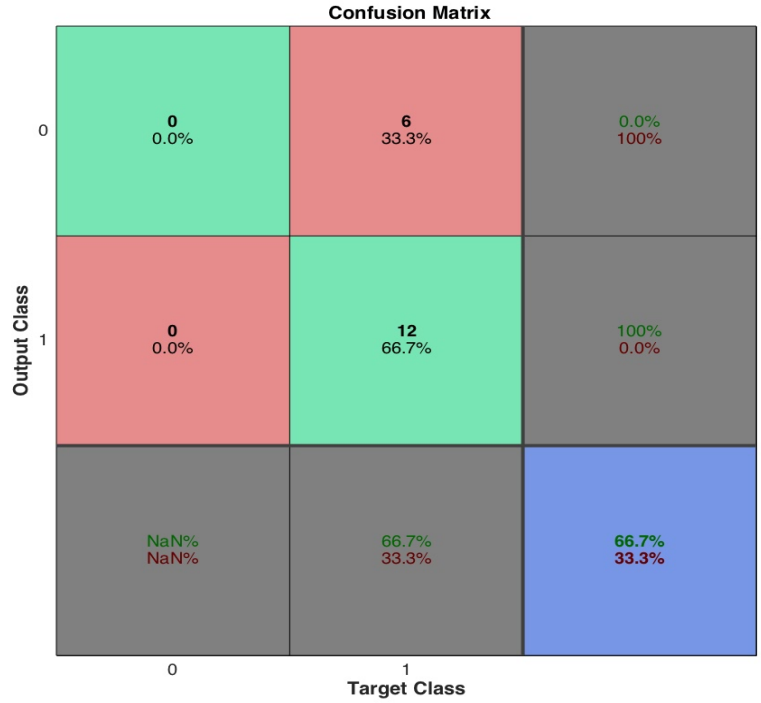
We tested our system in a quiet area in order to reduce background noise . We did this as some of words became unrecognisable to our program because of this interference. We used a confusion matrix in order to illustrate the result of our testing with our input. See MATLAB function: `confusion.m`.[5] These are the result for each command:



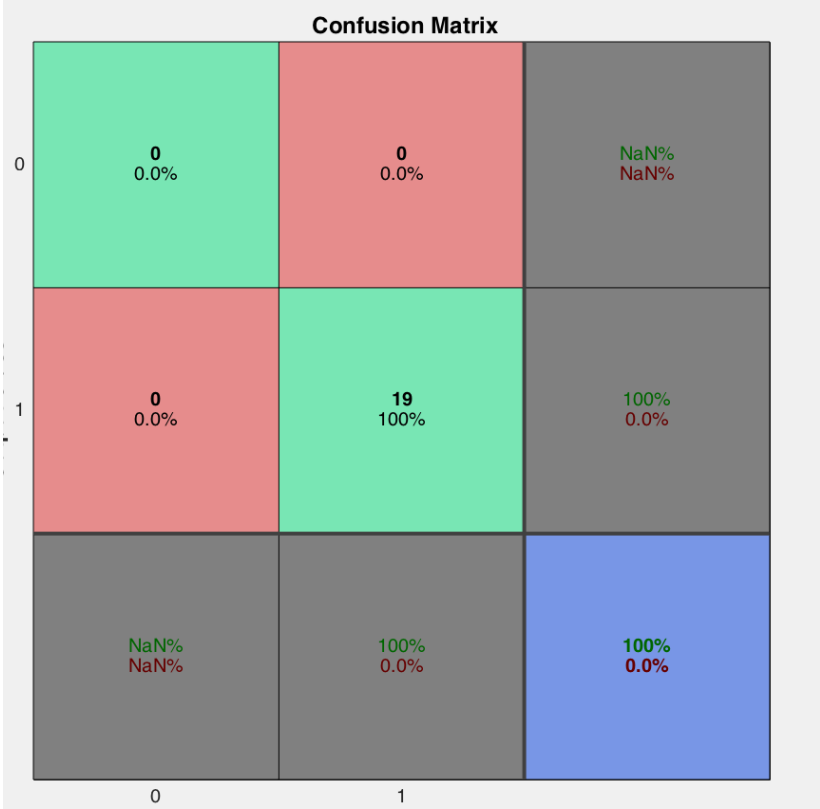
Confusion matrix of the command “Left”



Confusion matrix of the command “Stop”



Confusion matrix of the command “Right”



Confusion matrix of the command "Forward"

This may have been because the of the noise in the background. Alternatively, another cause could have been pronouncing the word in a different manner.

Ethics

Since our system first asks the user to take samples of his or her voice which are used to recognize the commands spoken by the user, the system uses and saves data from its user. If one thinks about this, one can come up with ethical questions such as "What happens with the data?" It is good to think about this, because the user might not think about ethical questions such as this one, but it is very important for the designers of system to think about it. For instance, the data can be sent to a big data center where all the data of every user of the system is stored. This data center needs to be well-secured, because there might be someone wanting to gain access into the information that is stored in the data center. This might not seem to be a "real" problem, but what if the user can enter the system with his or her password. Maybe, the system saves information about where the user lives and where the user goes, and travels on a chip. This chip can, for instance, be used to find a lost wheelchair (using this system). Actually it is big problem if third parties try to enter the data center. The designers of system should try to not let this happen.
Moreover, "What if other people try to enter the system without the user knowing or wanting this?" This issue is not as big as it looks like, because the system is calibrated on the user's voice. However, it might still be possible for others to control the wheelchair (using this system) as their voice resembles the user's voice. Or, maybe they hacked into the system.

In addition, another ethical issue might be caused by people who get attached to the wheelchair and cannot live without it anymore. For example, people who are going to use the speech-controlled wheelchair temporarily, but they get attached to it, because, for instance, they like the wheelchair a lot or it is really easy to control as it takes less effort to

control a speech-controlled wheelchair than a wheelchair that has to be controlled or moved manually. The speech-controlled wheelchair (using this system) can be used by anyone. However, this ethical “problem” can be solved by giving this wheelchair to a small group of people. So, it would be helpful if only people who are in need of such a device get one.

Furthermore, the wheelchair might not be hundred percent safe in terms of usage. The wheelchair can be controlled with speech. To control the wheelchair, the user needs to say commands, which the system then tries to recognize. What if the user says “Forward” and the wheelchair goes straight forward. This may not seem as a big problem. but this might be dangerous for both the user and his or her surrounding as the wheelchair may collide into something or someone. However, this ethical problem is by far the “easiest” to solve. The solution to this problem comes in two parts: the first part is detecting and avoiding obstacles and the second part is using verification commands (verifying if the user really wanted the recognized command to be executed).

In conclusion, our system would work perfectly and fine in the “real world” as long as the system is really well-secured and as long as the information of the users is not unwillingly gained by third parties. Also, it should be only available for people who are really in need of a speech-controlled wheelchair. The system could be improved by implementing obstacle detecting and stopping when obstacles are too close and by using verifying commands.

Evaluation and conclusion

When we came up with the idea to build a system that can recognize speech, we actually thought that it would be difficult. But, we underestimated the level of difficulty of this project. Especially, because it is a speech-recognizing system, it is hard to come up with an idea to make the system.

What we first did was taking twenty samples of each word. At that time we wanted to distinguish between five commands (“Fast”, “Left”, “Right”, “Stop” and “Quit”). The next thing we did was that we tried to use mel cepstrum coefficients to distinguish between the different words. However, this approach did not work out, because each sample would give an array containing thousands of values. Also, it was very hard to decide which value to choose in order to distinguish between the words. Still, we tried this and we tried it with a while-loop which would continuously record sound and the code would look at a specific mel cepstrum coefficient. This worked somewhat, but we thought that this was not a good approach to our problem. Therefore, we decided to come up with something else. Also, we decided to try to distinguish between only four commands (“Forward”, “Left”, “Right” and “Stop”). The new approach was to record new samples and save them to our workspace. We then stored the values from the result of MFCC into a vector and then proceeded to compare these values with our input signal . The MATLAB functions would save the recorded samples. Then, we had to come up with an idea to process the new spoken commands in such a way that the system would recognize the commands properly. This was the hardest part.

We did not use neural networks, which could have been in a way easier, since the Neural Network toolbox for MATLAB was provided. Still, we would have to do research on how to use neural networks and figure out a lot about using neural networks to recognize speech, but it definitely was an alternative.

To conclude, our systems works pretty good. However, it needs improvement, such as using ultrasound sensors to detects objects close to the robot. Also, the result of the testing is not hundred percent, thus the system can definitely use improving techniques.

Reference

Image on the front page
http://www.skomaasdal.nl/nieuws/module-robotica/

Literature study [1]
https://www.researchgate.net/publication/4307102_Voice_controlled_intelligent_wheelchair

Literature study [2]
http://www.iraj.in/journal/journal_file/journal_pdf/1-290-147573788733-37.pdf

Literature study [3]
https://www.researchgate.net/publication/4307102_Voice_controlled_intelligent_wheelchair

Literature study [4]
https://www.ijirset.com/upload/2016/incets/17_incets74.pdf

Testing [5]
(Matlab help)
https://nl.mathworks.com/help/matlab/
https://nl.mathworks.com/help/nnet/ref/plotconfusion.html

Frequency, sampling and bit rate
http://www.audioshapers.com/blog/bestaudio-recording-format.html

Remove Silence from signal
https://www.youtube.com/watch?v=lyF_d4QYCc8

Tutorial for mfcc

Voicebox
http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

To process the sound, we have used functions which were provided by the website (indicated by the link above). We used the functions in the code so that we could process the sound properly. This is the section where we refer to the website from which we downloaded and used the sound-processing functions we used to process the sound in our project.