# B.Tech. Project Final Report
# 2019-2020

## Mechanical and Control System Design with fabrication of Human Assisting Droid

Mechanical & Industrial Engineering Department,
IIT Roorkee, 247667

**Submitted by:**
Rohan Saraswat (16117073)
Jayant Teotia (16117037)
Morvi Bhojwani (16117053)

**Submitted to:**
Professor Dr. S.P. Harsha
MIED, IIT Roorkee

# CERTIFICATE

We hereby certify that the work, presented in this report, entitled, **Mechanical and Control System Design of Human Assisting Droid**, towards the fulfilment of the requirements for the award of the degree of Bachelor of Technology in Mechanical Engineering, submitted to the Department of Mechanical and Industrial Engineering, Indian Institute of Technology Roorkee, India. This is an authentic record of our work carried out during the period from July 2019 to June 2020 under the guidance of Dr. S.P. Harsha, Professor, Department of Mechanical & Industrial Engineering, IIT Roorkee.

The matter embodied in this report, to the best of our knowledge, has never been presented for the award of any other degree elsewhere.

Date: 8th June, 2020
Place: Roorkee

…………………………
(Supervisor’s Signature)

# ACKNOWLEDGMENT

We would never have been able to complete our work without the guidance of our supervisors, help from friends, and support from our families and loved ones.
We would like to thank our supervisor Dr. S.P. Harsha for providing us an opportunity to work on this challenging project. His support and continued motivation helped us to gain insight into various aspects of research and development. We would also like to express our gratitude to the department for providing the requisite resources.

On a personal note, we would like to thank all the people especially our parents and teachers for providing guidance in various aspects of life and supporting us throughout these four years.

At last but not the least, we thank God/Nature from our souls who made all the things possible despite the ongoing pandemic. We also pray for everyone who is affected by CoronaVirus directly or indirectly.

# CONTENT

# Abstract

With ever-evolving technology, robots have become an integral part of our life. Nowadays, robots can be seen working in almost every field, from working in factories, performing surgeries in places inside the body where a human hand is difficult to reach to exploring distant planets, etc. Our team saw a need for a personal assisting robot in various general professions like a garage owner, a surgeon, etc. So we aim to design an appropriate droid for performing the basic tasks with minimum energy and cost requirements.

So to explore and learn more about robots, we decided to design and fabricate a human assisting droid optimized for a specific task. Also to power our bot, we will be using a customized control system. Thus our designing process started with looking into the current designs and we came up with an innovative design that can deliver products per user input. Currently, we have zeroed down on a design that has two segments with one movable and other non-movable. These two parts will work in sync with each other to deliver any product as per user input.

There are two segments of work - Mechanical System Design and Control System Design. We look for innovation in both segments through iterations. This report is about the first iteration and future deliverables of our project. We have made a working prototype that conforms to the design we made and hence a positive outcome for the next improved iteration.

# Objectives

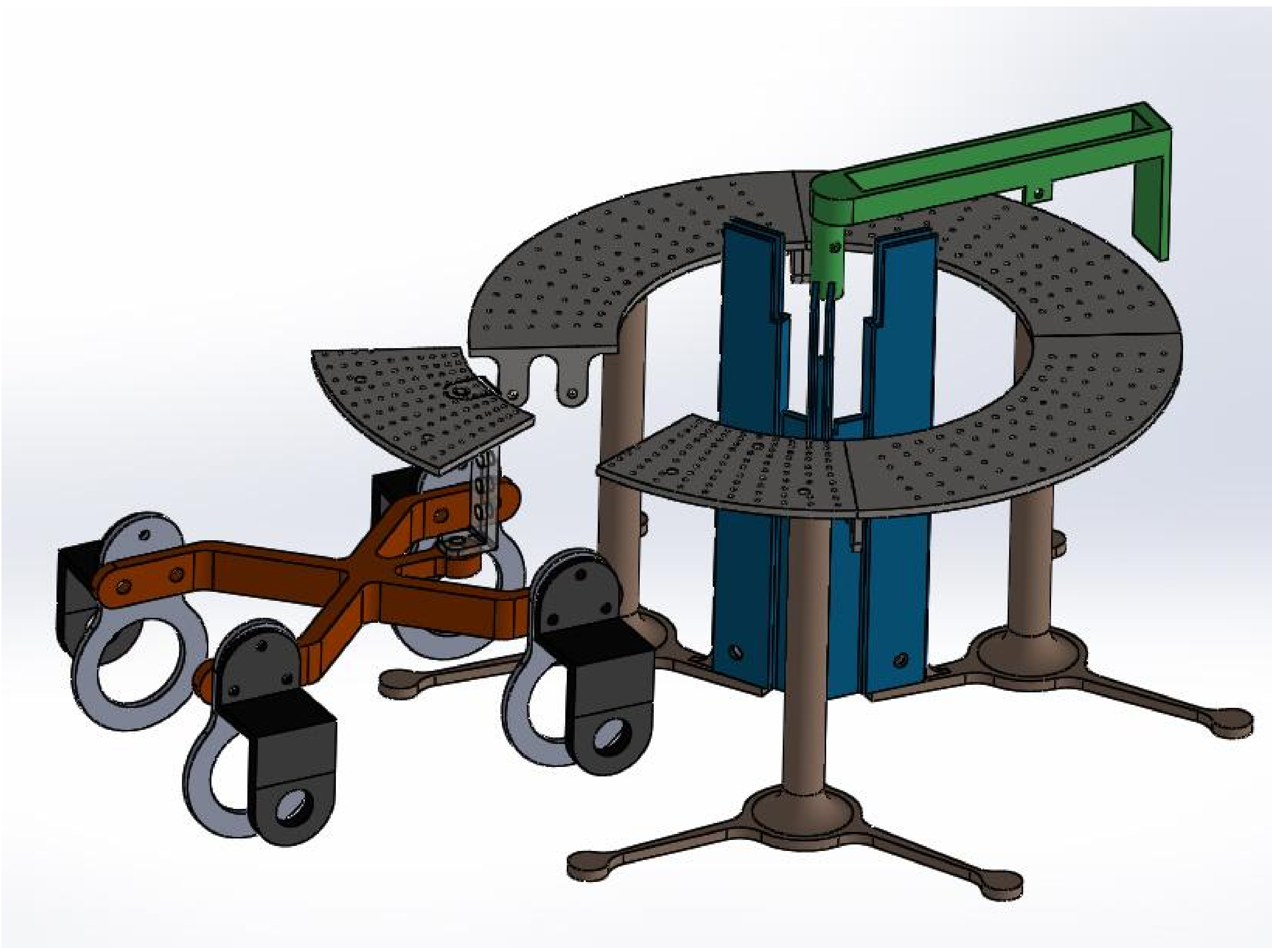
The main objective of our human assisting droid is to bridge the gap between robotics and its use in day to day life. We aim to come up with a design that is user-friendly, works in simple ways, is cost-effective, and has less weight. These objectives are to be met with the continuous iterations we make over the course of time available.

The objectives can be summarised as follows:-

- The bot with possible configuration changes as per requirements.
- Delivers an article with precision at the user's final location.
- Incorporate portability for household and professional uses combined.
- Uses a Bluetooth application for user input.
- Incorporate semi-autonomy for path tracing.
- Uses sensors to locate the position of the user.

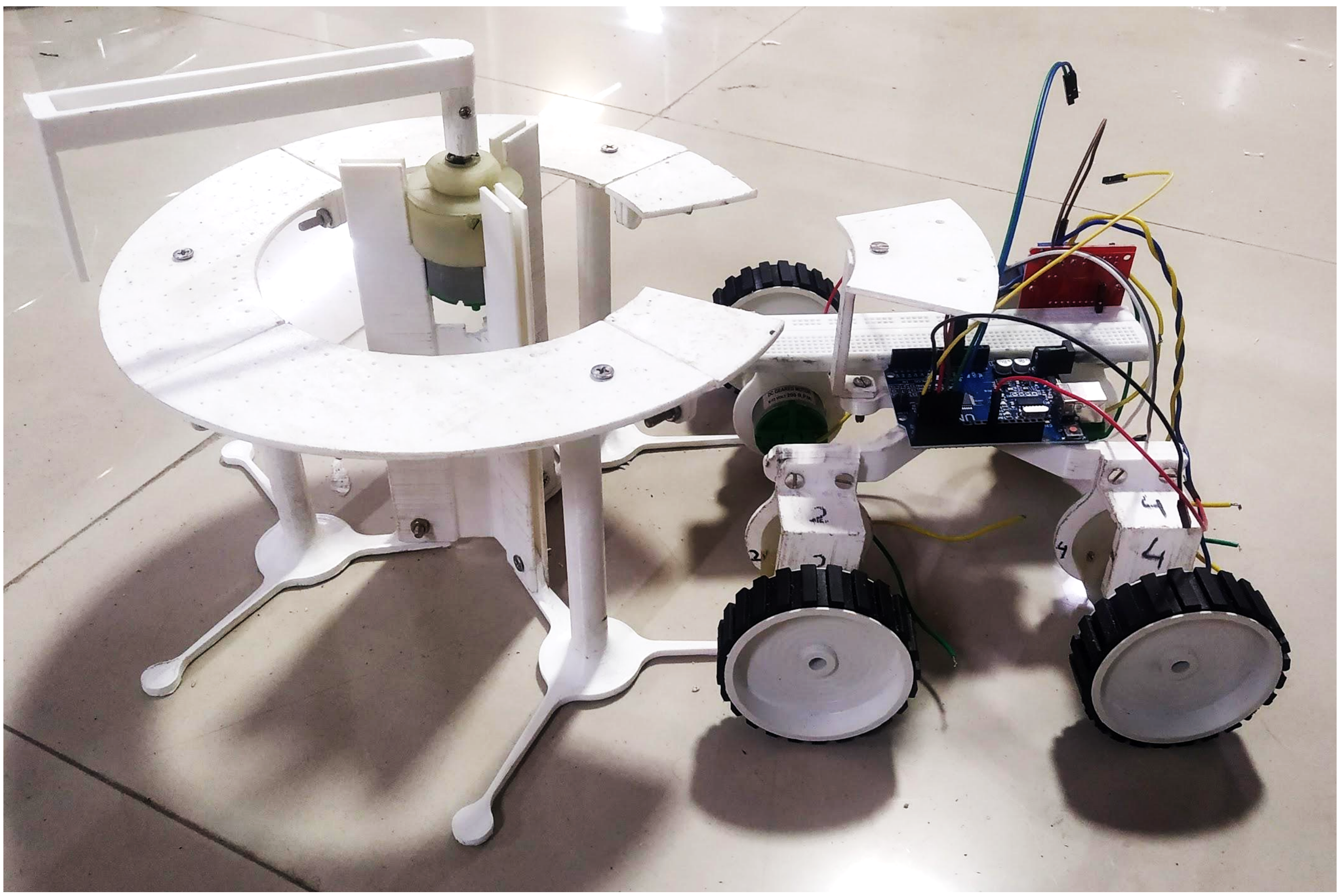
The objectives have to be met with accuracy so as to deliver a useful robot.

# Design



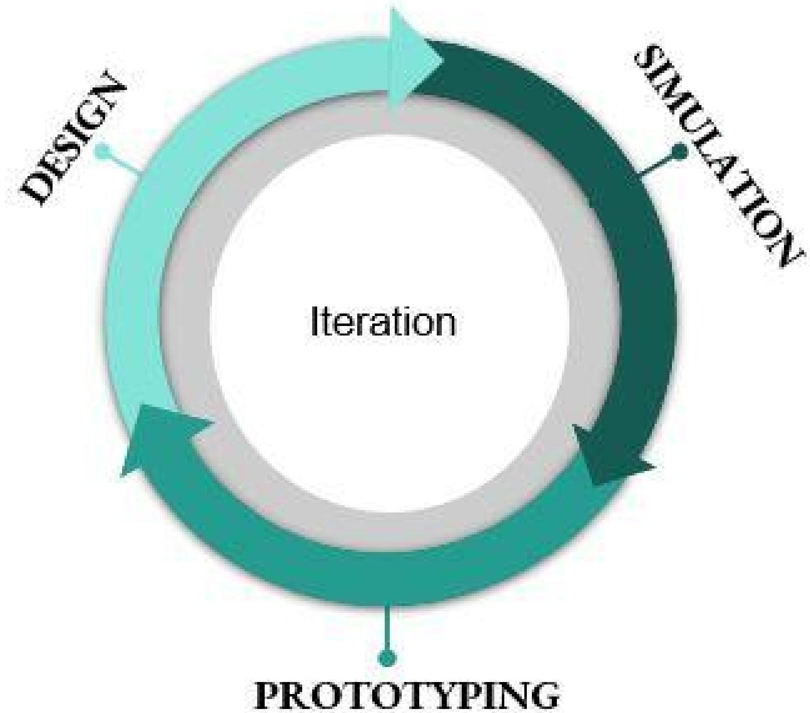
Simulated design of the droid

Using the above design and 3D printing the parts with high accuracy and design considerations, the model was created and stands as shown in the figure.



The two visible movable and non-movable parts will fit at the junction in the circular plate.

# Methodology - Mechanical System Design




The basic design of the droid includes a circular disc-shaped tray with slots to keep the required instruments or objects in it, an arm to pick up a required object from the particular slot, a movable part that will deliver the object to the destination, customized control system that will receive the instructions from the operator and control the mechanical motion of the droid.

To explain it's working let's take an example of assisting a mechanic in the garage. There are a variety of tools like wrenches, spanner, hammer, etc to be used while repairing a machine. All these will be kept in respective slots on a circular disc. The operator will send the command/ instructions through a Bluetooth device to the droid.

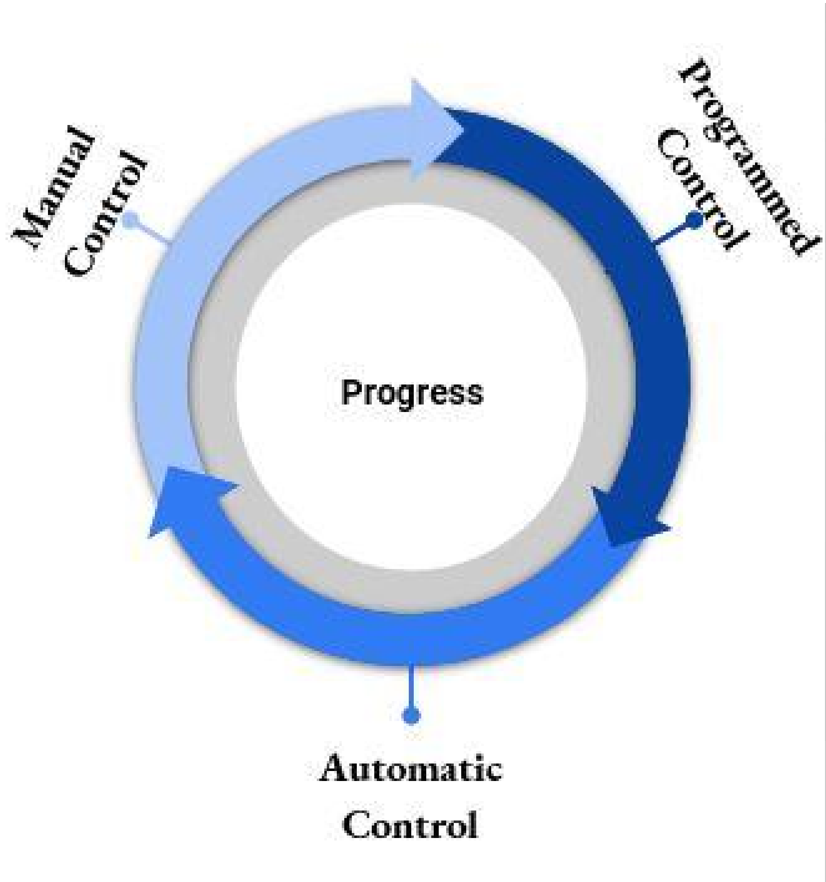
The control system will receive the signals, process it, and guide the mechanical motion to complete the given command. The droid can pick the required tool from the slot with the help of an arm which can perform the both circular and linear motion. For the picking mechanism, we will be using a couple of servo motors to hold the object. It will pick the tool and place it on the master slot. All slots except the master slot will be fixed.

The master slot will have wheels attached to it so that it can move to the operator. It would be detachable from the disc as a sector of it. First, the arm will keep the required tool from its original slot to the master slot as per user input. Then the slot will detach itself from the structure and travel to the operator to deliver the tool. After completing the task, the master slot will move back to its initial position.

We will be facing several challenges in performing the above set of tasks. One of them would be the accuracy of the arm to pick the correct object from the slot, and placing it on the master slot. Another is keeping back the object to its allotted slot after use correctly. Also controlling the movement of master slot especially it's detaching from the main structure and coming back to its initial position etc.

Our first task is to come up with a working basic design implementable in a variety of professions. The initial target is the management and delivery of tools in a garage. Once ready with the design, we will prototype it combining it with the control system designed by our partner team on the same project. We are planning to expand the usage of this concept to certain high skilled and precision fields like assisting surgeons during surgeries.

# Methodology - Control System Design




The basic design of the droid includes a circular disc-shaped tray with eleven out of twelve slots to keep the required instruments or objects in it, an arm to pick up a required object from the particular slot, a movable one-twelfth part of the central disk that will deliver the object to the destination.

In the sense of controlling the bot, the major moving parts are the central rod which supports the arm, the arm, and the one-twelfth of the circular disc that will be used to deliver objects. The central rod and the arm will be controlled by the motors, while the segment of the annular disk will be moved by a level 1 autonomy system incorporated with PID or MPC. To reach the target positions of the objects placed at the disk we will manually give direction through a Bluetooth mobile design application.

There are several challenges that the control team may face include the very basic problem of precision control. While most of the motion will be controlled manually initially, we will try to make the system semi-autonomous to do that high-level accuracy will be the huddle in our path. Further to make the bot fully autonomous image processing techniques will be used to identify the obstacles in the path of motion of bot.

The segments of work include -

1. To identify primary dynamic parts of the bot.
2. To design a basic control system to provide the desired motion to the moving part in bot.
3. To study the motion of the arm required to pick and place the objects from the tray.
4. To incorporate the level 1 autonomy system with PID or MPC to achieve the motion of the moving circular portion of the disk

# Mechanical System Design

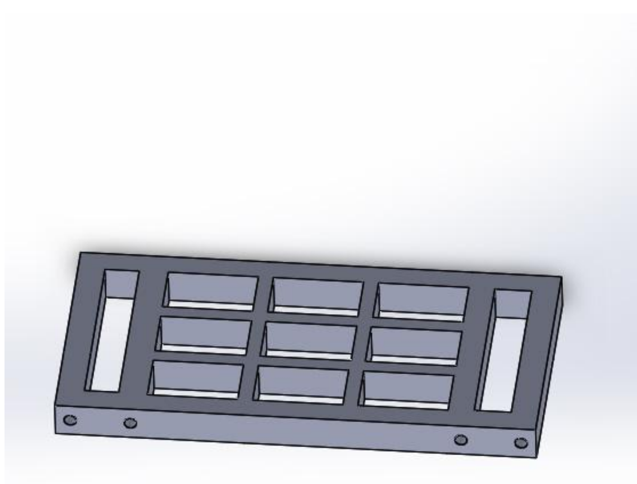
As per the methodology given above, the work was divided into three segments:-

**Design**

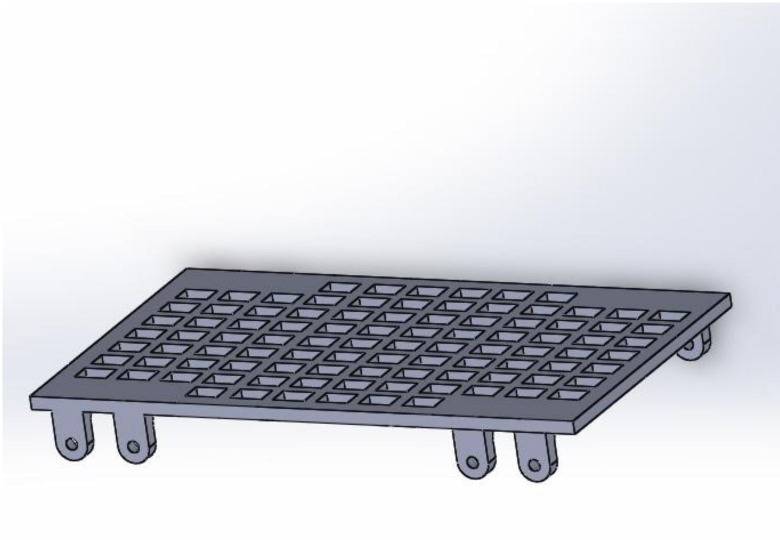
The designing phase starts with defining the constraints of the problem and then coming up with a design to suit the needs. Hence first the dimensions were decided for the whole structure. The circular disc has an inner radius of 7cm and an outer radius of 12cm. The whole plate is raised from the ground with an offset of 15cm. The wheels used are of 6cm diameter and the wheel separation is kept 15cm.

Using these constraints, the design was made in Solidworks. The circular cross-section was finalized for the base on which objects will be placed because it is easily motorable through a single motor and hence reduces weight and power requirements. Also, the movable segment has an X as the base to reduce weight as checked with various design iterations.

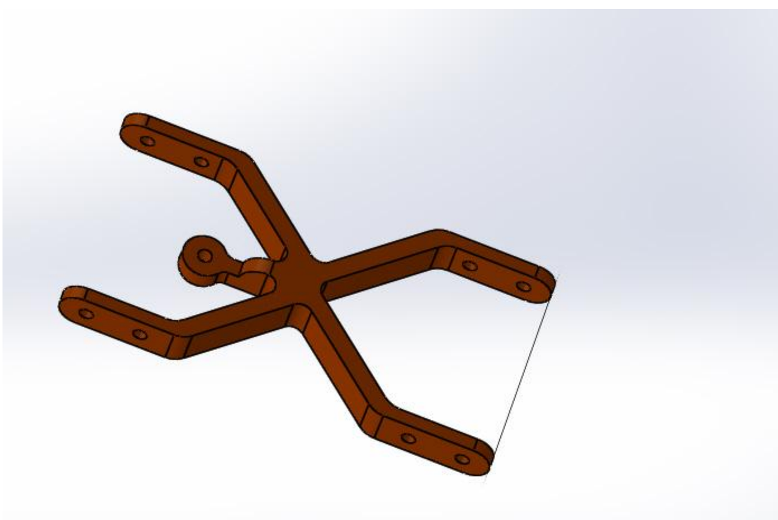
One major iteration was done in the base selection -



Iteration 1: Weight - 72 grams



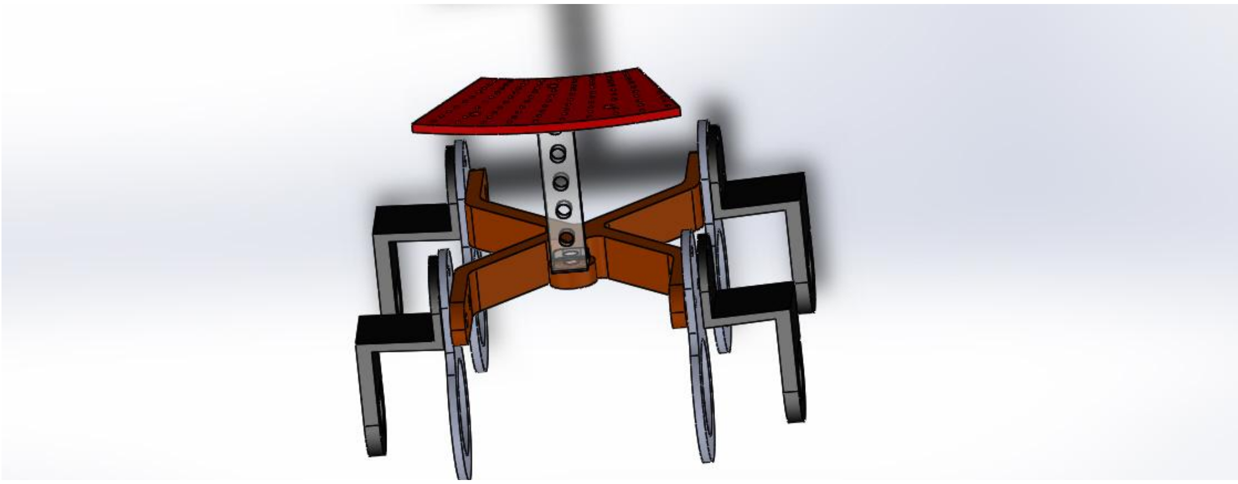
Iteration 2: Weight - 42 grams



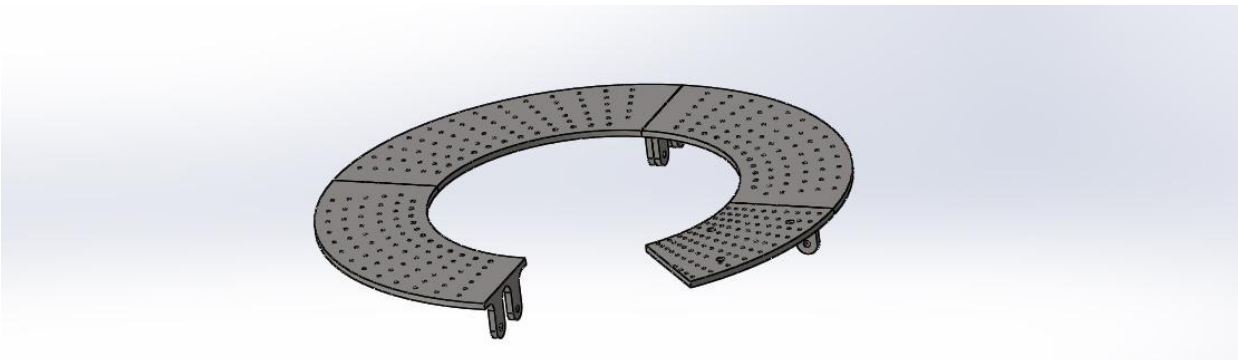
Iteration 3: Weight - 32 grams

**Modeling and Simulation**

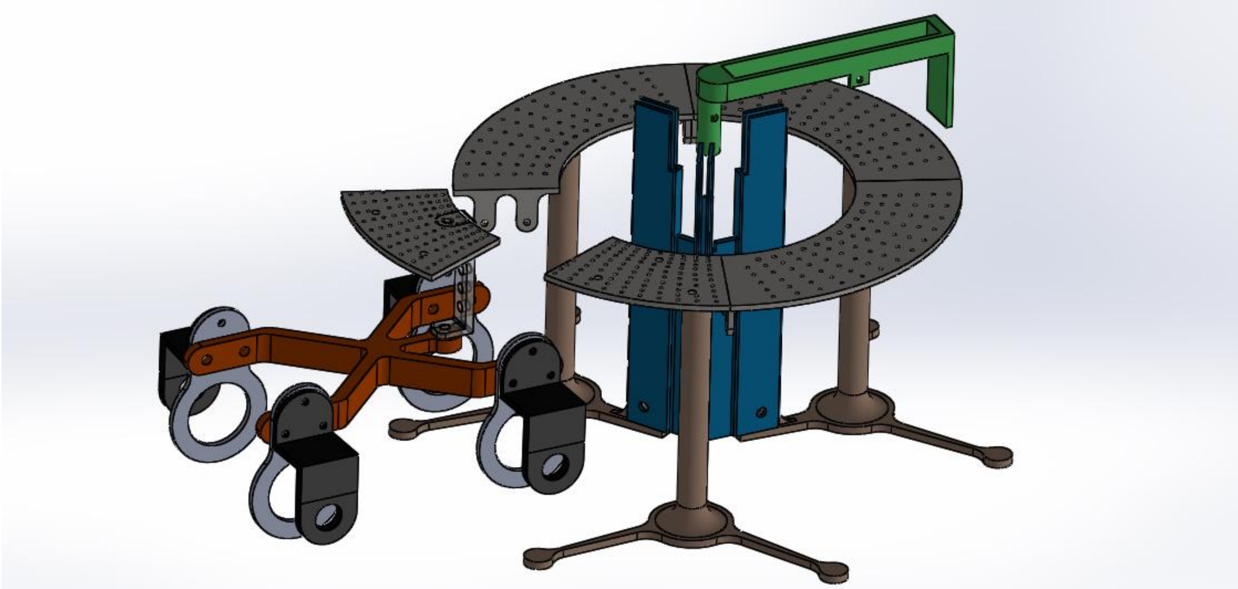
The full subparts were designed and then assembled in SolidWorks for checks in design alterations. The places where design faltered and didn't mate with other parts, the parts were redesigned to come up with a full-scale model.



The complete model (Without motors) of the mover robot of the droid



⅞ segment of the circular plate on which objects will be placed



The complete model and assembly of the first prototype

## Prototyping

3D Printing

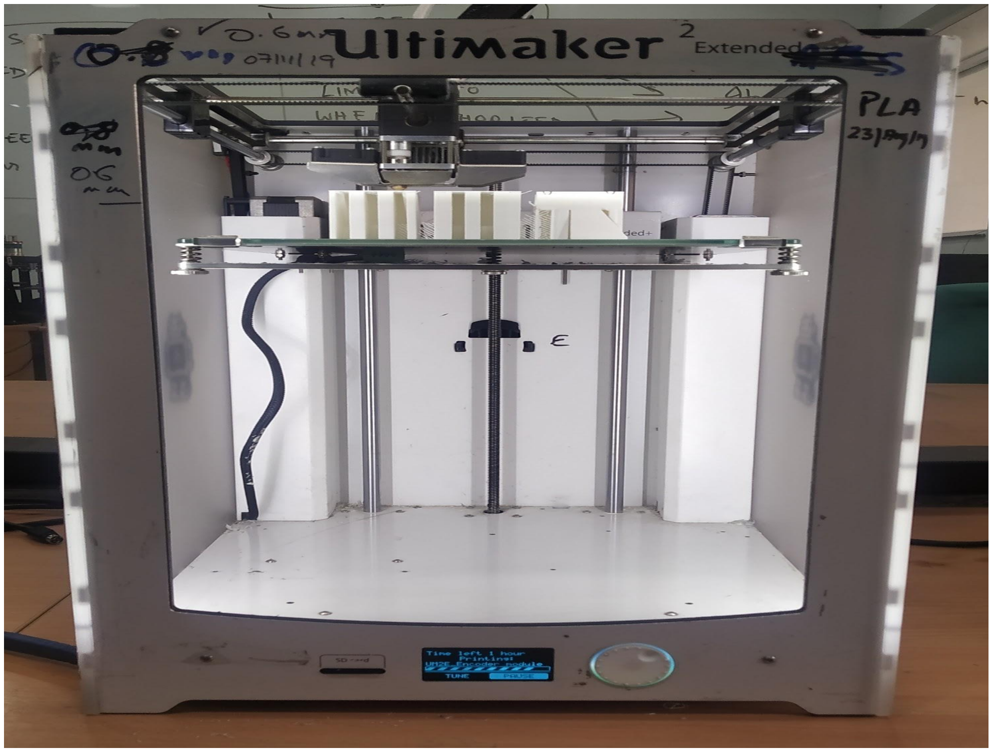
The **3D printing** process builds a three-dimensional object from a computer-aided design (CAD) model, usually by successively adding material layer by layer, which is why it is also called **additive manufacturing**, unlike conventional machining, casting and forging processes, where the material is removed from a stock item (subtractive manufacturing) or poured into a mold and shaped by means of dies, presses, and hammers.

The term "3D printing" covers a variety of processes in which material is joined or solidified under computer control to create a three-dimensional object, with the material being added together (such as liquid molecules or powder grains being fused together), typically layer by layer. In the 1990s, 3D-printing techniques were considered suitable only for the production of functional or aesthetic prototypes and a more appropriate term for it was rapid prototyping.

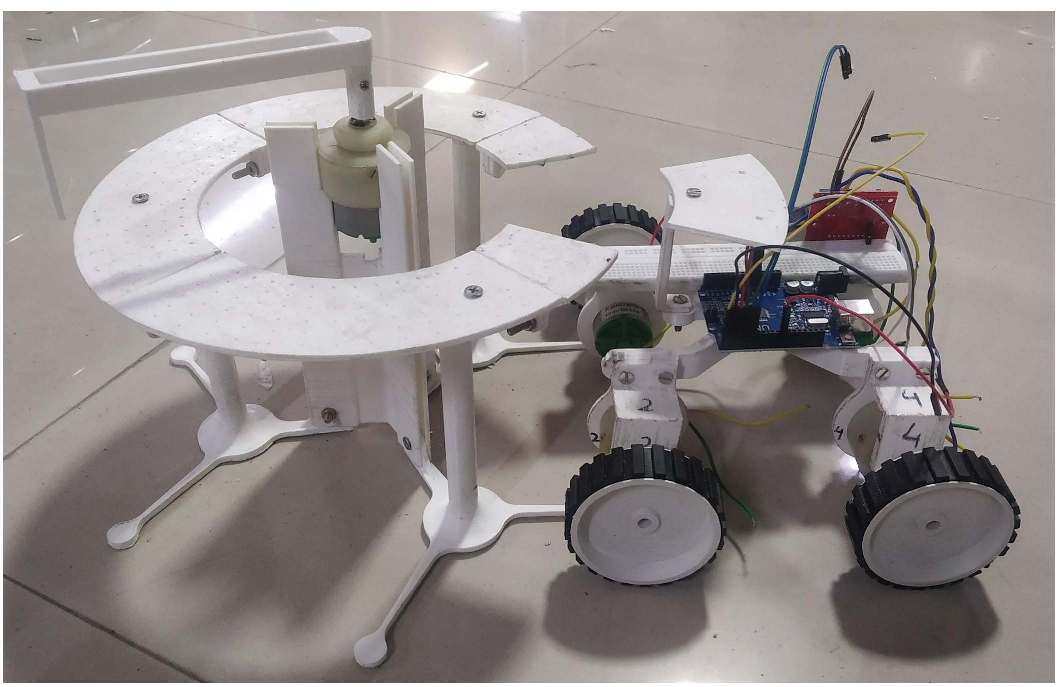
As of 2019, the precision, repeatability, and material range has increased to the point that some 3D-printing processes are considered viable as an industrial-production technology, whereby the term additive manufacturing can be used synonymously with "3D printing". One of the key advantages of 3D printing is the ability to produce very complex shapes or geometries, and a prerequisite for producing any 3D printed part is a digital 3D model or a CAD file.

Using this technology at Tinkering Lab in our institute, we were able to materialize our design. The 3D printer used the material ABS with a 0.6mm nozzle which operated at 230 degrees Celcius to melt the material.

The machine - Ultimaker was used for the operations.




The achieved prototype is as shown in the figure:



# Electrical Design

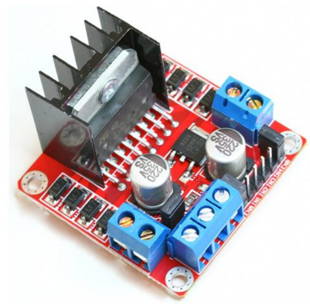
As the purpose of our droid was to grip an object, move it on the master slot and deliver the item to a specified location, hence an electrical circuitry was designed. It is composed of the following components -

- 3 LiPo Batteries(12 volts)

  Lithium Polymer batteries (henceforth referred to as “LiPo” batteries) are a type of battery now used in many consumer electronics devices. They have been gaining in popularity in the radio control industry over the last few years, and are now the most popular choice for anyone looking for long run times and high power. LiPo batteries offer a wide array of benefits. They are light in weight, have high power concentrations, and also can be made of any shape or sizes.

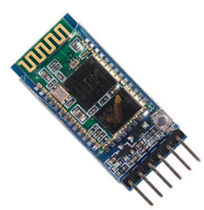

- 3 L298N Motor drivers

  The L298N is a dual-channel H-Bridge motor driver capable of driving a pair of DC motors. That means it can individually drive up to two motors making it ideal for building two-wheel robot platforms. It can control both the speed and spinning direction of two DC motors.



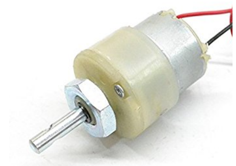
- 2 HC05 Bluetooth modules

  The HC-05 is a very cool module that can add two-way (full-duplex) wireless functionality to your projects. You can use this module to communicate between two microcontrollers like Arduino or communicate with any device with Bluetooth functionality like a Phone or Laptop. There are many android applications that are already available which makes this process a lot easier. The module communicates with the help of USART at 9600 baud rate hence it is easy to interface with any microcontroller that supports USART.



- 5 DC geared motors (300 rpm)

  DC geared motors can be brushed or brushless. Both of these types of motors use magnets to drive the spinning rotary motion of the motor but have different applications. Brushed DC geared motors have a lower cost and are often used for applications with simpler control systems.

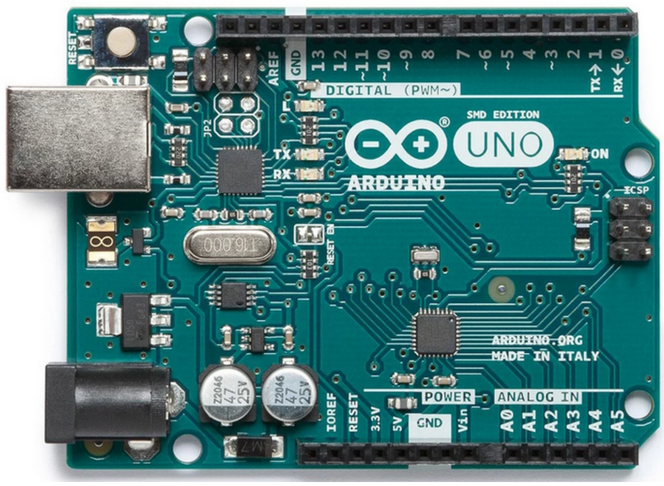


- 2 Servo motors with 180 degrees rotation (MG90S)

MG90S is a micro servo motor with metal gear. This small and lightweight servo comes with high output power, thus ideal for  RC Airplane, Quadcopter, or Robotic Arms. Robots where position control is required without feedback. Less weight hence used in multi DOF robots like humanoid robots.


- **2 Arduino UNO**

Arduino Uno is a microcontroller board based on an 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists of other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header, and a reset button.




**Central Structure - Robotic Arm**

The robotic arm was conceptualized to grip a specified object and move it to the master bot. To accomplish this task, a DC geared motor was used to rotate the arm of the central structure and it was connected to a motor driver.

For the reciprocatory motion of the arm and for the functioning of the gripper, two servo motors were used independently. The servo motors and the motor driver for the DC motor were connected to one Arduino.

A Bluetooth module was also connected for operating the central structure through a mobile app.

All the components were powered by a 12-volt rechargeable battery.

**The Moving Bot**

This bot was designed for delivering the object already placed on the master slot to a specified location. Four DC geared motors were connected to the wheels of the bot for providing motion to it.

Two motor drivers were connected to the four motors which were connected to an Arduino board. A Bluetooth module was connected to the board for operating it through a mobile app.

This unit was powered by two 12 volt batteries.

# Old approach for user input - Mobile App Design

Two separate mobile apps are designed for controlling the central structure and the moving bot independently. The code for the app was built on the MIT app inventor.

The app for the central structure allowed it to rotate the shoulder to any desired position (360-degree rotation). The reciprocatory motion of the arm and the gripping mechanism of the gripper is also independently controlled through the app.

The app for the moving bot allows it to move in the forward and backward direction and also rotate about the center, clockwise and anticlockwise.

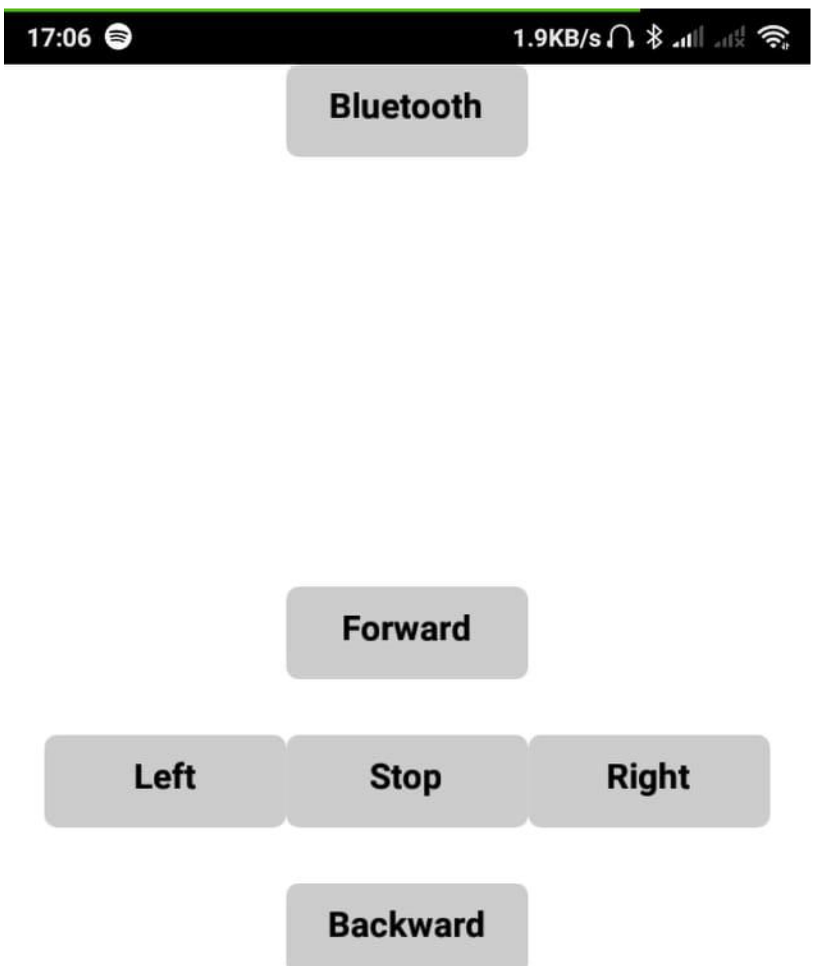



**Figure**: Bluetooth Application

# New User Input approach - Voice-based ML model

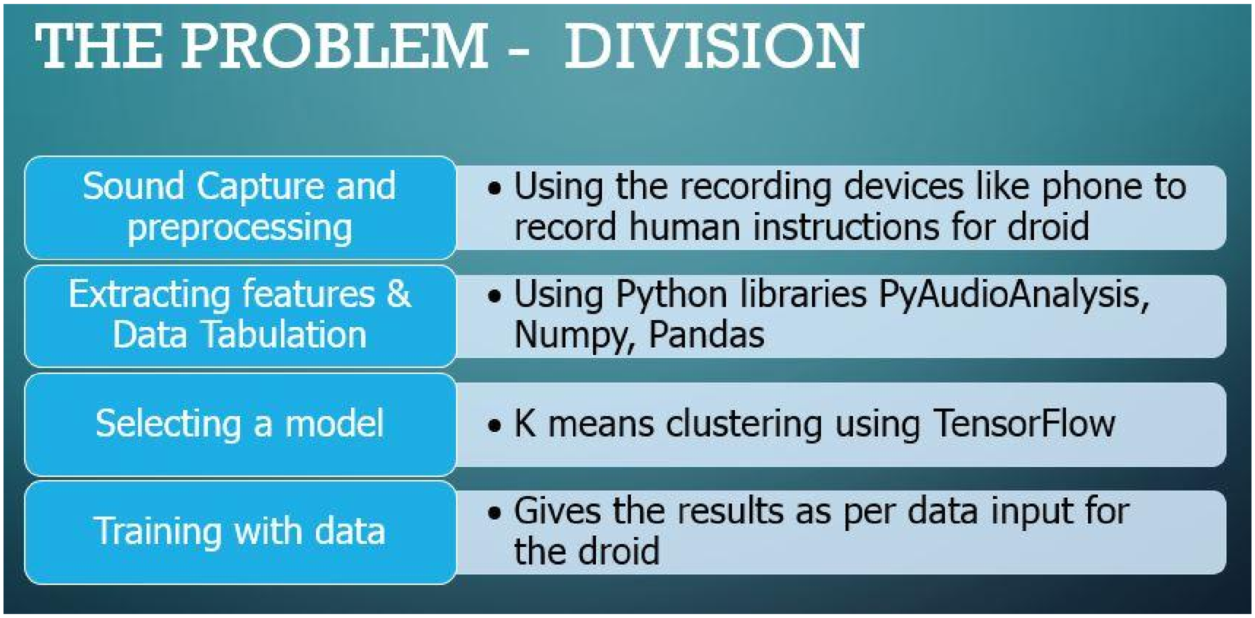
The motive of advancement in technology is largely to make the life of humans comfortable. Simplifying and humanizing the machine-human interaction for receiving instructions is an important aspect of it. In our droid, we decided to implement a user voice-based input for carrying out different tasks. This will make it easy to handle and could be devoid of any excessive hardware by using the aspects of unsupervised Machine Learning.

Machine Learning, popularly known as ML is a tool that can make machines interact with and behave like humans. It has various subdivisions depending on the way data is processed. The various parts are Supervised Learning, Unsupervised Learning, and Reinforcement Learning.

The various tools and resources used to implement ML are:- Data, Infrastructure/Platform, Algorithms, Feature selection, Model Training, and visualization. They are together used to teach a machine of finding patterns in the data and then extending the concept to new incoming data. This is what we aim to achieve in our project. The computer will be trained once with a lot of voices from users as training data. It finds various patterns in them and then creates a model that can recognize new incoming data.

The type of machine learning we have used in our project is Unsupervised Learning. In the case of unsupervised learning, not all variables and data patterns are classified. Instead, the machine must uncover hidden patterns and create labels through the use of unsupervised learning algorithms. The k-means clustering algorithm is a popular example of unsupervised learning. The advantage of unsupervised learning is it enables you to discover patterns in the data that you were unaware existed—such as the presence of two major customer types. Clustering techniques such as k-means clustering can also provide the springboard for conducting further analysis after discrete groups have been discovered.

The model we made can be divided into 4 subparts as per the approach to the problem. They are as per the figure below:-




**Sound Capture and Preprocessing**

The sound is captured through the microphone of an audio device. It records the audio using the time-amplitude plot. Using a spectrogram, this time-amplitude plot is converted to a frequency-amplitude plot. This is achieved through Fast Fourier Transform (FFT) which mathematically changes the time dependence to frequency dependence.

**Extracting Features and Data Tabulation**

Audio Features - Every music file is basically made up of two important things, the sample rate, and the bit rate. The rate of capture and playback is called the sample rate. The sample size—more accurately, the number of bits used to describe each sample—is called the bit depth or word length. The number of bits transmitted per second is the bit rate. Using the two, the library - PyAudioAnalysis can create 11 features of the sound data that can be used for clustering. They are as follows and could be used for data tabulation :

1. Energy
2. Entropy of Energy
3. Spectral Centroid
4. Spectral Spread
5. Spectral Entropy
6. Spectral Flux
7. Spectral Roll off
8. MFCC
9. Chroma Vector
10. Chroma Deviation
11. Zero Cross Rate

All of the above features have some or the other use but the one that we will be using is the *Chroma Vector*. It is a representation of how humans relate colors to notes. In other words, we think of the same notes but from two different octaves to be of the same color. Thus we have 12 possible values at each window.

This is done through following sample code :-

```
def preProcess( fileName ):
[Fs, x] = audioBasicIO.readAudioFile(fileName)
```

```
if( len( x.shape ) > 1 and  x.shape[1] == 2 ):
x = np.mean( x, axis = 1, keepdims = True )
else:      x = x.reshape( x.shape[0], 1 )
F, f_names = audioFeatureExtraction.stFeatureExtraction( x[ :, 0
], Fs, 0.050*Fs, 0.025*Fs)    return (f_names, F)
```

Once the features are extracted, Chroma Vector features can be returned to the code.

```
def getChromagram( audioData ):

temp_data =  audioData[ 21 ].reshape(1, audioData[ 21 ].shape[0]
)

chronograph = temp_data

for i in range( 22, 33 ):

temp_data =  audioData[ i ].reshape(1, audioData[ i ].shape[0])

hello  chronograph = np.vstack( [ chronograph,  temp_data ] )

return chronograph hello
```

Using the data extracted from the above function, it is tabulated using Numpy Matrix for later training the model.

**Selecting a model**

We start by defining the hyper-parameters for the K-means clustering algorithm. k is the number of clusters we wish to segregate the data into. epochs are the number of iterations the algorithm will run for. Both of them can be changed as and when necessary.

```
def initilizeCentroids( data, k ):

centroids = data[ 0: k ]

return centroids

X = tf.placeholder( dtype = tf.float32 )

C = tf.placeholder( dtype = tf.float32 )

C_labels = tf.placeholder( dtype = tf.int32 )
```

So these are the tensors that will act as placeholders for our data. Here X will represent the data, C is the list of k centroids and C_labels is the centroid index that has been assigned to each data point.

Now we need to do an element-wise square for the whole matrix and sum up the values along the 3rd dimension. And we will be left with the distance of each point with all the K vectors. All we have to do is find the least one for each of the data points. As the model is ready, we can train it using the dataset earlier created.

**Training the model**

The final code has to be written which implements K-means clustering and iterates the code for new centroids after each computation. This is done with this final code to get the results.

```
data_labels = []

centroids = []

with tf.Session() as sess:

sess.run( tf.global_variables_initializer() )

centroids = initilizeCentroids( data, k )

   for epoch in range( epochs ):
```

```
       data_labels =  sess.run( getCentroidsOp, feed_dict = { X: data, C: centroids } )

       centroids = sess.run( reCalculateCentroidsOp, feed_dict = { X: data, C_labels: data_labels } )

print( data_labels )

print( centroids )
```

This is the driver function of the training algorithm. We feed the graph all the necessary data from here. The training loop is also defined here. This clusters the sound data with labels.

In our droid, once this algorithm is ready, we can train the model for specific verbal instructions. When a user says a word, it will check its label and accordingly will send the value to the BlueTooth which will transmit it to the robot for application.

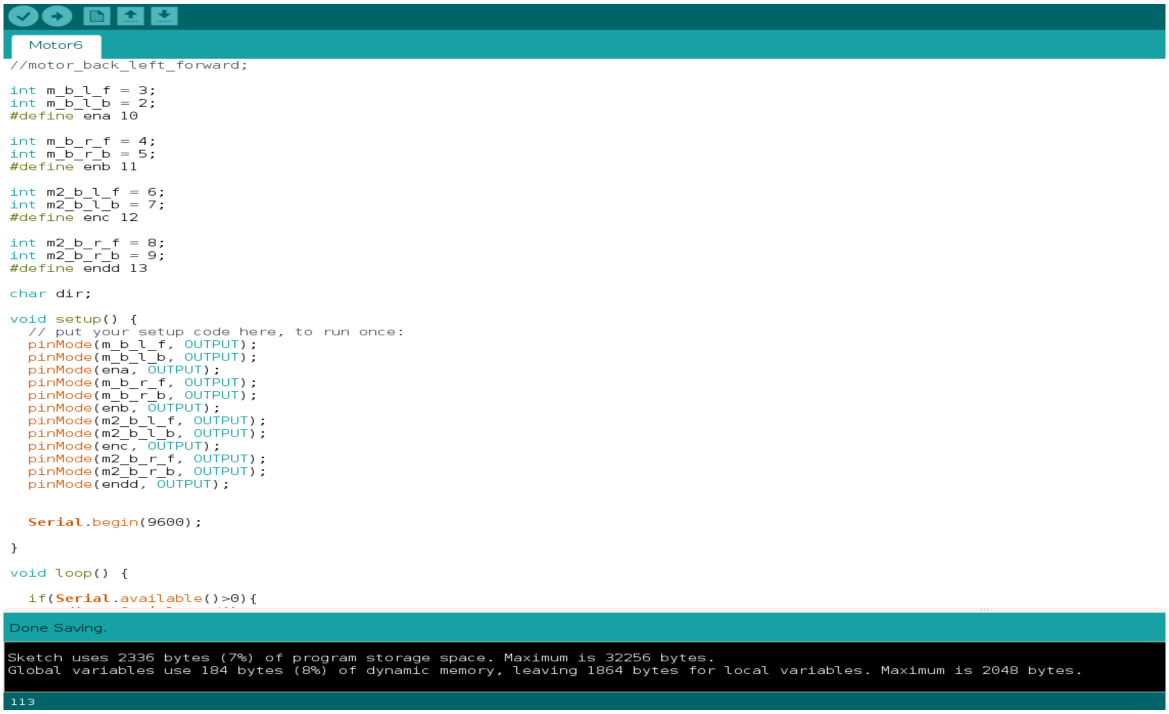
# Control System Design - Manual Control

For the smooth functioning of the droid, a code was written on the Arduino IDE.

- DC motor control – Two digital pins from the Arduino were connected to the motor driver for operating a single motor. By providing 'HIGH' signal (5 volts) to one pin and 'LOW' signal (0 volts) to another pin, the motor can be made to rotate clockwise or anticlockwise. A PWM digital pin from the Arduino was connected to the motor driver to control the speed of the motor.

- Servo Motor Control – The servo motor was directly connected to the Arduino board and was driven by the 5-volt power input. A PWM digital pin from the Arduino was connected to the motor which was used to specify and control the degrees of rotation.

- Bluetooth Module control – The Bluetooth module is powered 5-volt power input. The transmission (TX) and receiver(RX) pin of the BlueTooth module were connected to the receiver (RX) and the transmission(TX) pin of the Arduino board.




# Control System Design - Autonomous (I)

The final aim of the project was to give a voice input and accordingly the robotic arm will pick the object from the circular platform and place it on the four-wheeled differential drive mobile robot. The mobile robot will then navigate autonomously and reach the final point.

The algorithms were developed and tested on Robot Operating System (ROS) and were tested on a Gazebo based F1-tenth simulator which is developed by the University of Virginia for testing similar robots.

**Robot Operating System**

The Robot Operating System (ROS) is a flexible framework for developing Models and Algorithms for Robots which can be easily deployed on a physical robot. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.

The Gazebo is a platform on ROS where a physical model is developed on which the algorithms are tested. It has 'topics' to which messages can be published and a control command is given for the working of the bot.

'Nodes' are an important part of the ROS framework as these contain the algorithms for the bot. The advantage of using the ROS framework is that nodes can be created independently of each other and can be executed at any time in a simulation. The conversation between nodes and the robot model, which happens through messages published on topics, is completely independent of the other nodes being executed at the same time.

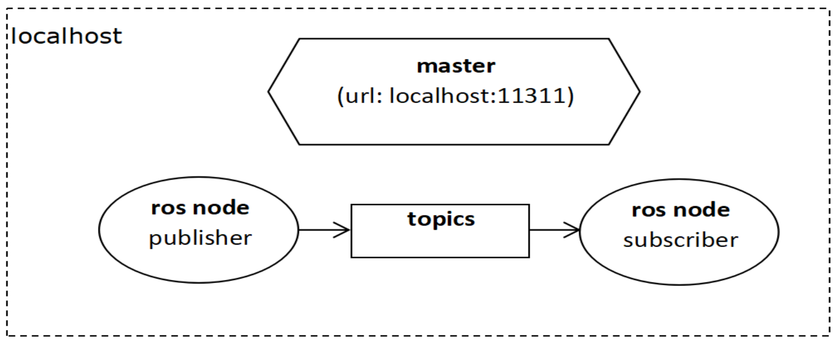



Figure: ROS Framework

'Topics' are named buses over which nodes exchange messages. Topics have anonymous publish/subscribe semantics, which decouples the production of information from its consumption. In general, nodes are not aware of who they are communicating with. Instead, nodes that are interested in data subscribe to the relevant topic; nodes that generate data publish to the relevant topic. There can be multiple publishers and subscribers to a topic.

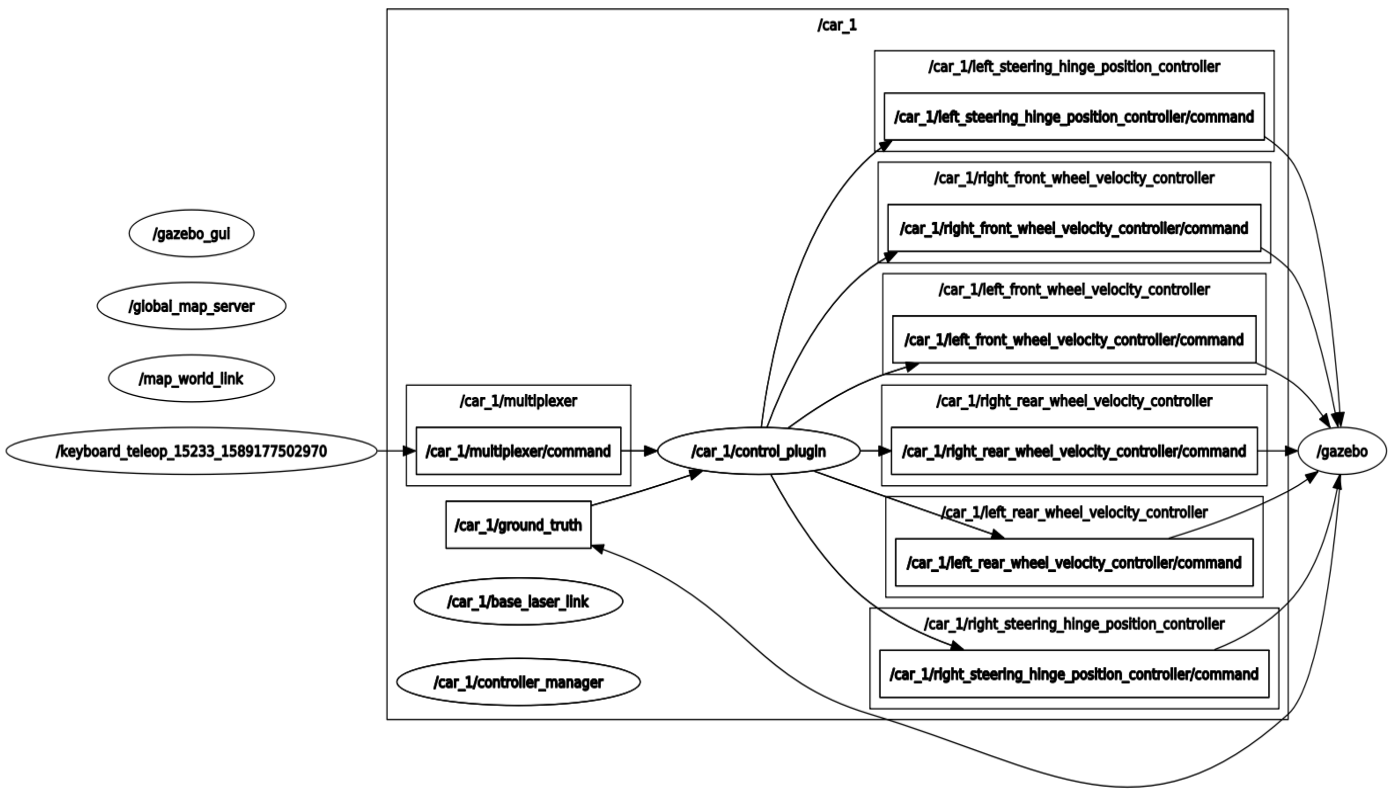



**Figure**: Topics for our project in ROS

We developed the algorithms for autonomous navigation in the ROS framework.

The Robot has a 2-dimensional LIDAR mounted on it which gives the angle and distance of the nearest Obstacle. For the autonomous control of a mobile robot, there are two components which need to be addressed -

- Mapping of the environment and localization of the bot in that environment.
- Navigation Strategy.

Mapping and Localization helps map the environment of the robot and Localizes the robot in the environment. While Navigation Navigates the robot in the environment, avoiding any obstacle.

**Mapping and Localization**

Mapping and Localization is done simultaneously using **Hector Slam technique**. This technique matches the point clouds for subsequent scans and estimates the position of the obstacles as well as the state of the robot simultaneously in the world frame. It  is an open source algorithm used for building a 2D grid map for the surrounding environment based on a laser scan sensor (LIDAR). This algorithm locates the position of the robot based on scan matching and doesn't use the odometry of the wheel which is the common method in other SLAM algorithms. The LIDAR manages it to perform the scan matching task to locate the robot fast and accurately due to its high update rate. Hector algorithm uses Gaussian Newton minimization method which is considered as an update for Newton method, it has the advantages that second derivatives needn't to be computed.

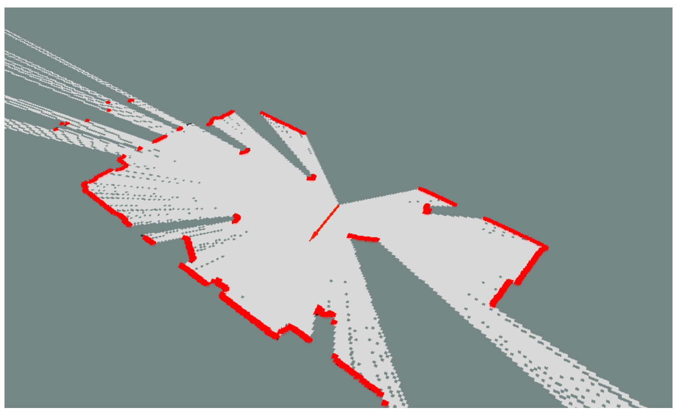


Figure: LIDAR point cloud for single scan

**Navigation - Obstacle Avoidance**

In this project, Obstacle avoidance is achieved by a reactive method. Reactive methods are preferred over predictive methods due to their less computational time and easy implementation. The algorithm our robot uses is developed using Gap following technique, where the robot moves in the direction of the farthest obstacle in the (current) laser scan. The novelty of this approach lies in the creation of a new method for analyzing openings in front of the robot that highly reduces their number when compared with the Nearness-Diagram Navigation (ND) technique, particularly in complex scenarios.

# Control System Design - Autonomous (II)

**Reinforcement Learning**

Our team also tested this novel method for controlling the mobile robot as Reinforcement learning is the future of autonomous technology. It is an area of Machine Learning. This technique leverages feedback from previous iterations. It takes suitable action to maximize reward in a particular situation. It is used by various software and machines to find the best possible behavior or path it should take in a specific situation. Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

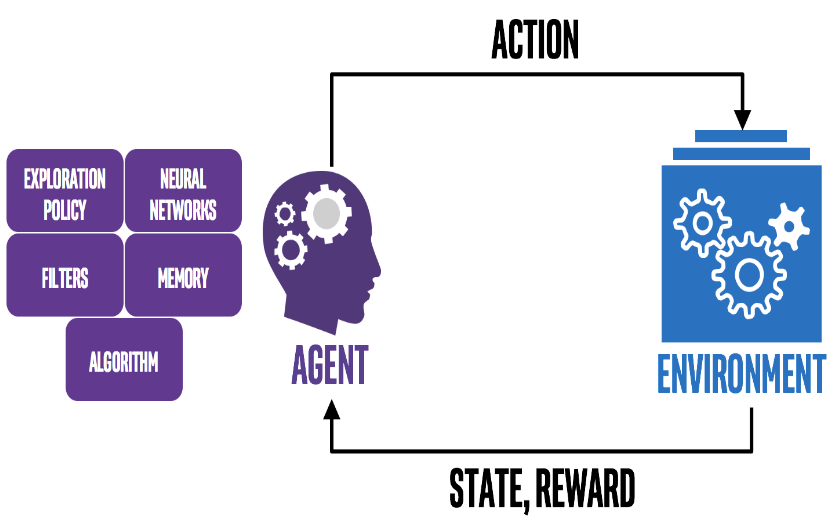



**Figure**: Reinforcement Learning Framework

Here we don't have to specify the actions taken by the robot in different scenarios, rather it learns itself by making mistakes. The robot tries out different actions in a simulation until it learns an optimized set of actions which gains maximum reward. In the basic model we developed, we were able to make the model give equal velocities to its 4 wheels and make the robot run in a straight line. Though it seems a trivial simulation, an advanced version of this will be able to navigate through obstacles all by itself without any algorithms specified for different scenarios.

**Straight Line - Reinforcement Learning**

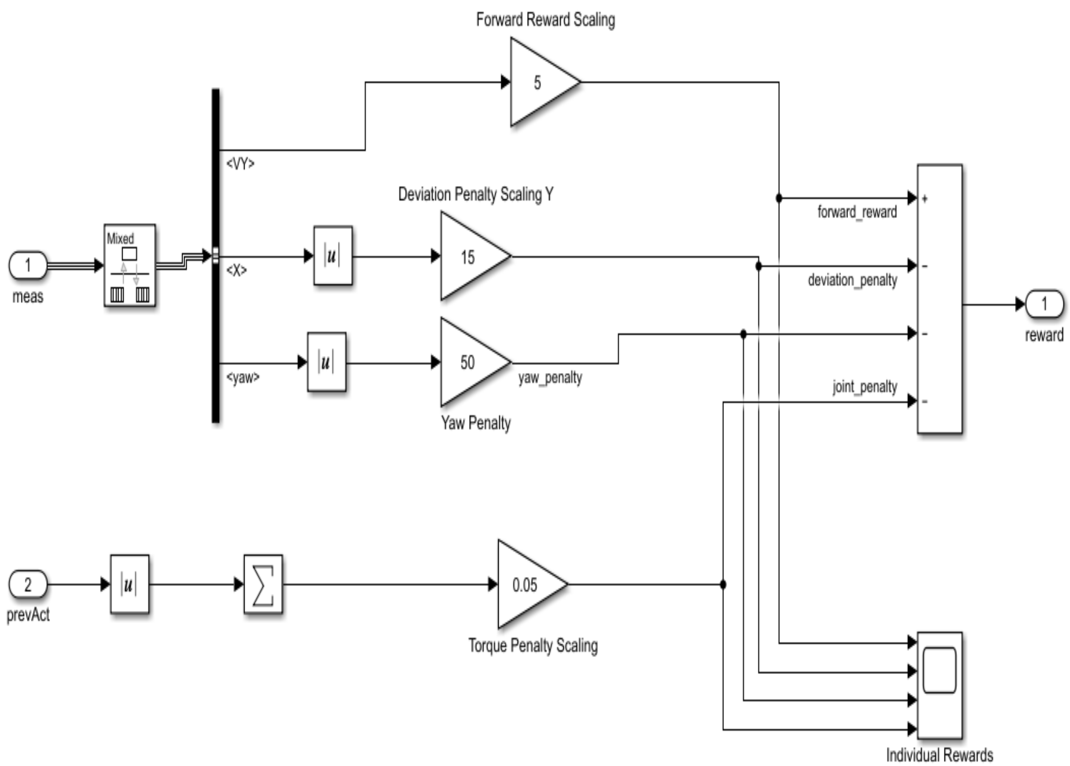



**Figure**: Reward Function

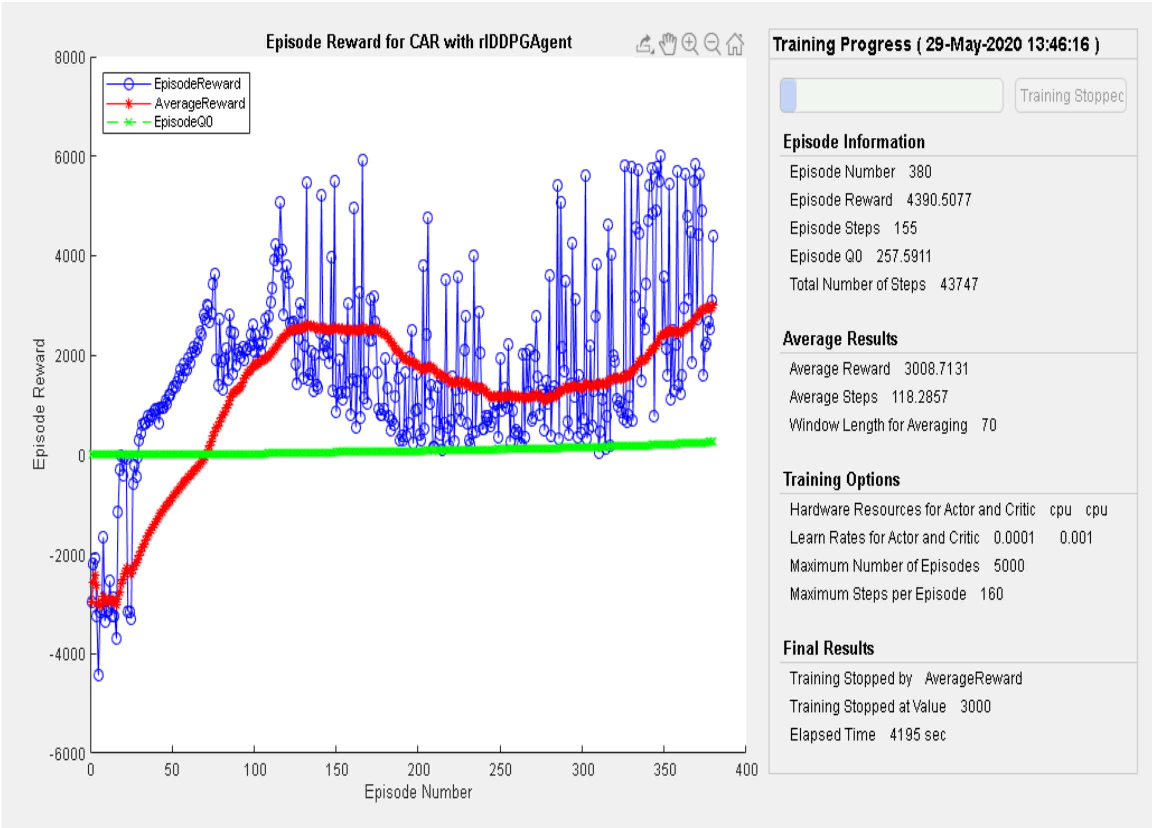



**Figure:** Final Results - Average Reward

# Conclusion

Engineering innovations are true to their spirit if they can be implemented in the real world and solve the prevailing problems using the available technology. Our team saw a gap that while there are numerous applications incorporating ease of use through path breaking concepts of Machine Learning, Neural Networks etc, there is dearth of the same in the commercial mechanical world. Hence we decided to bridge it through our project which enabled us to bring a lot of aspects of Engineering together at a place and make things work. This was an implementation based project where we were successful in achieving the desired motion in a two divide robot through user voice used as input. We made the fully functional first prototype which worked a proof of concept for our design while incorporating simple control systems. There was an abrupt stop in the progress due to the CoronaVirus pandemic as we were supposed to go home devoid of resources. Our team developed the code and approach to autonomous control systems and ML based user input but the conditions didn't allow for second prototyping.

We successfully delivered on our objectives and see a lot of scope for further improvements in the design as robotics continues to take strides towards a sustainable future.

# Future Perspectives

This project can be implemented on the following ideas:

- With a robust and tested basic design, this design could be extended to a variety of professions with a need for assistance.
- With high precision, this could be used for surgical robotics or as an assistant in surgery.
- Once implemented, the portable design of the droid can be made for further convenience.

# Bibliography


- arxiv.org/pdf/1509.02971.pdf
- www.ros.org
- linklab-uva.github.io/autonomousracing/page3.html#content10-u
- ‘Machine Learning’ by Oliver Theobald
- ‘Deep Learning’ by Ian Goodfellow, Yoshua Bengio, Aaron Courville
- www.kaggle.com
- https://github.com/audio_signal_clustering/blob/master/K_means_audio.ipynb
- ‘Machine Learning with TensorFlow’ by Nishant Shukla

**THANK YOU**