GROUP MEMBERS:
2001EE19, 2001EE23, 2001EE79, 2001EE27

EE305

Control Panel, under Dr. SHOVAN BHAUMIK

Design Lab Project

A Scientific Approach of Violin Manufacturing and Plate Testing

CONTENTS

Introduction

We know violin is an important musical instrument, we know what a soft music it produces, we know what an attractive and dynamic design it gets by the human hands with the smoothen surface finish on just a simple piece of wood but we also know about the pricing of this, it becomes costlier cause of two reasons, first is the wood are used to make it and second one is the cost of the rare human talent. Hence, its pricing become over the lakhs which is sometimes unpayable for lot of interested ones.

So, Our Objective is to -

- Reduce the cost to minimum and make it affordable to all.
- Minimize the human effort in the world of machines.
- Manufacture violin with the help of Computer Numeric Control (CNC) Machines after being modelled on the designing software.

OVERVIEW

We developed an engineering method for violin manufacturing that differs from the traditional Luthiers approach.

Our focus was on designing and manufacturing wooden parts. We shall the wooden parts with a rubber hammer to capture sound and used software (e.g. Audacity) to analyze the sound using MFCCs .

We will compare the analysis with the original spruce used and draw conclusions on whether violin manufacturing using an engineering approach is feasible.

DESIGNING
(CREO)

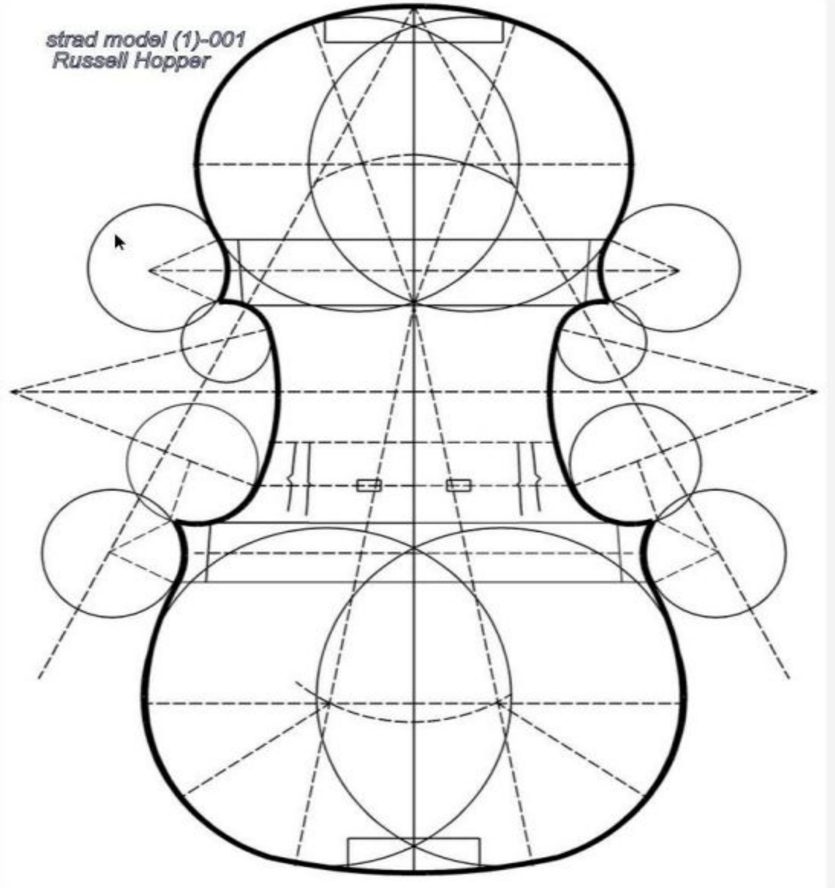
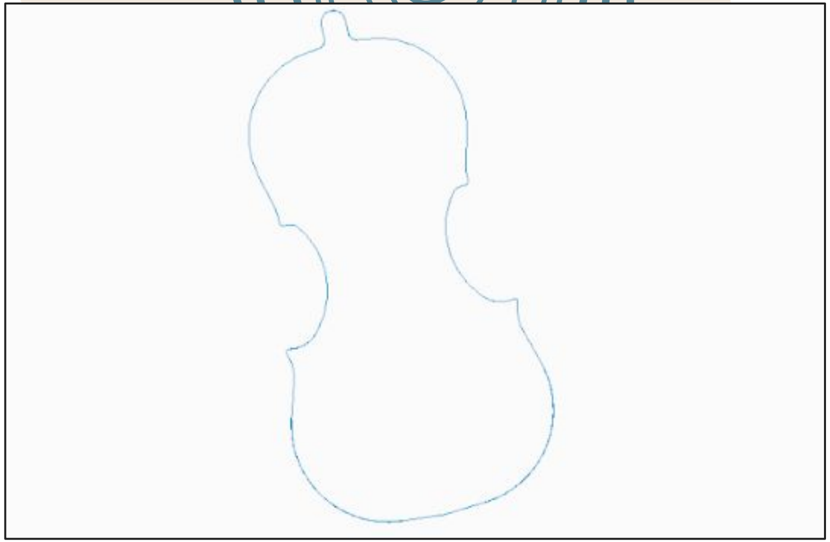
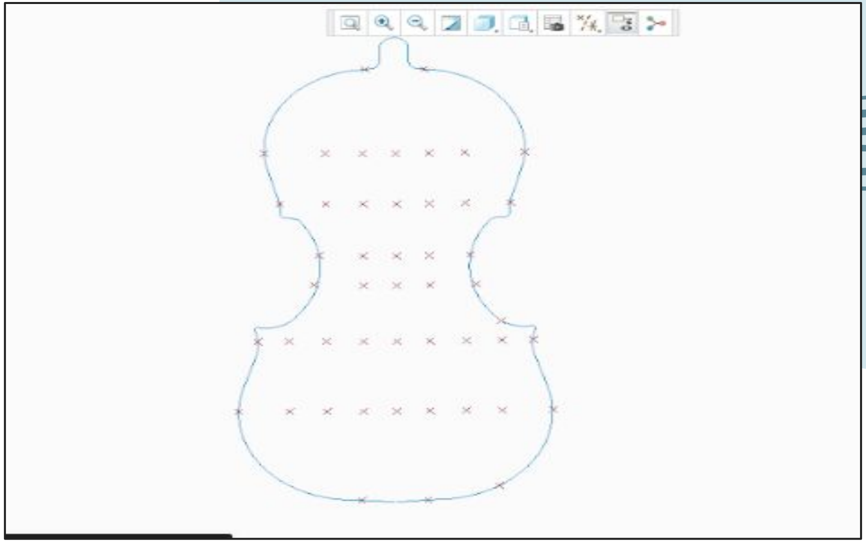
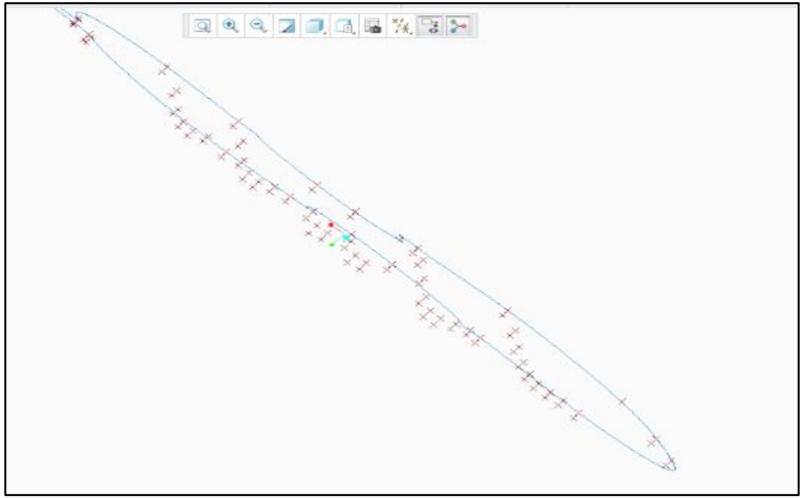
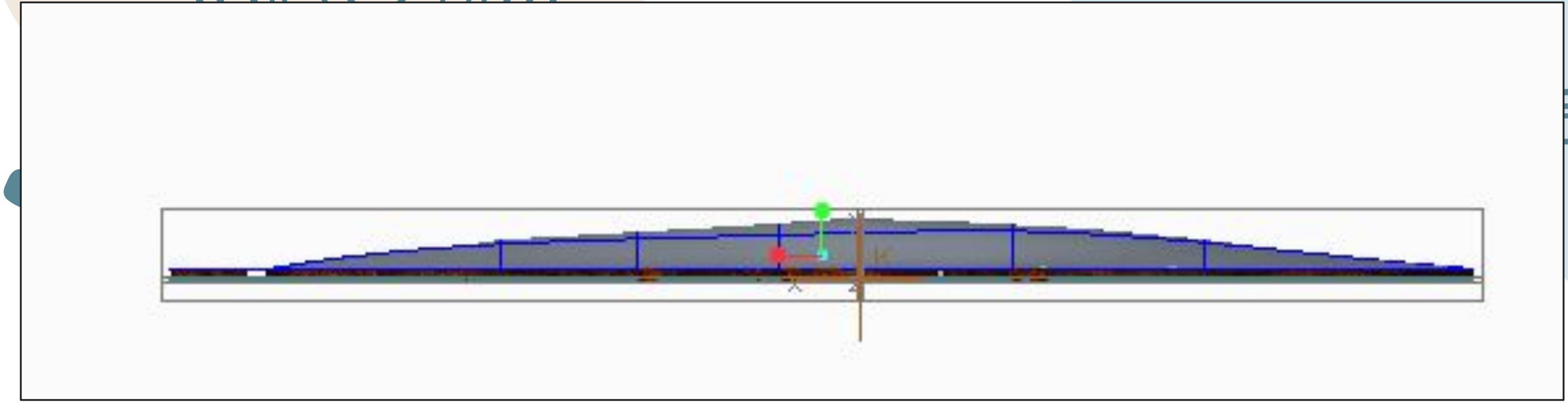


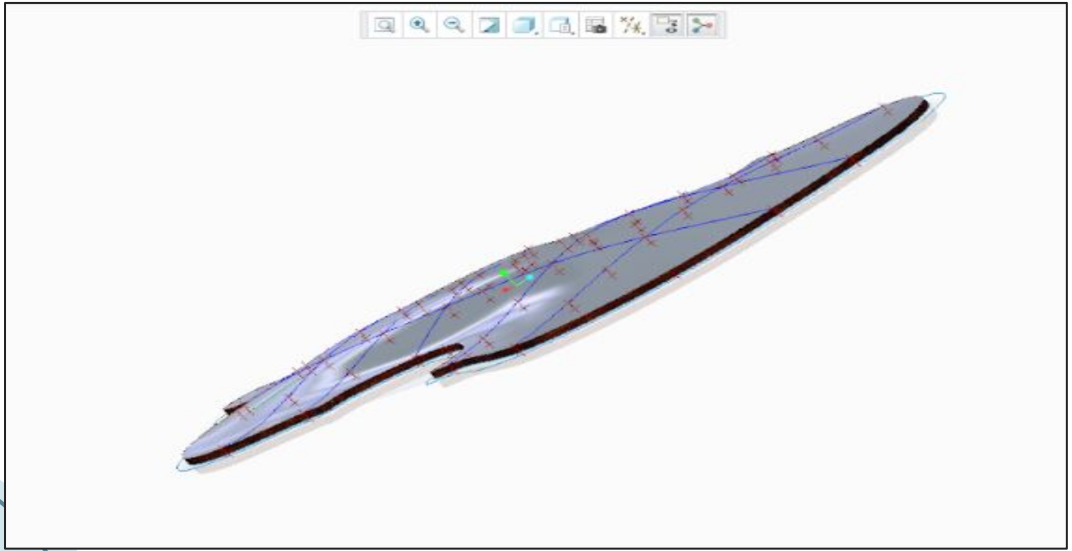
Figure 2. 2D drawing of violin by making circles

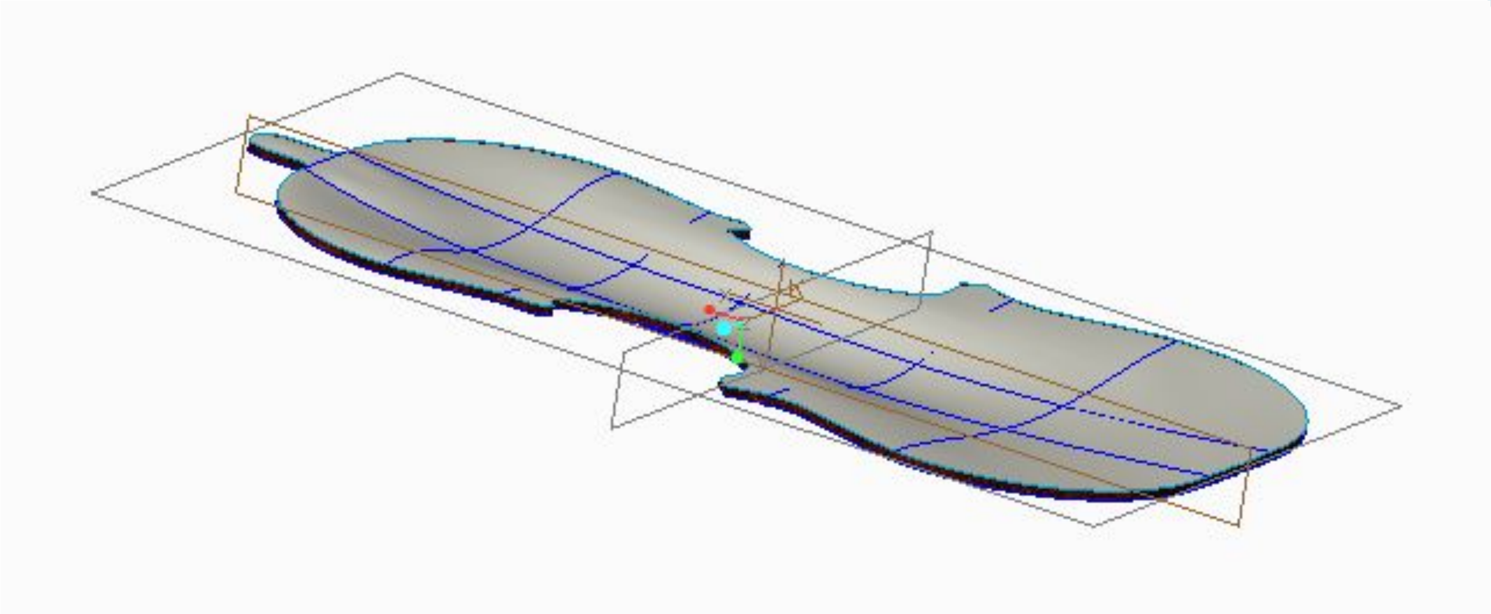


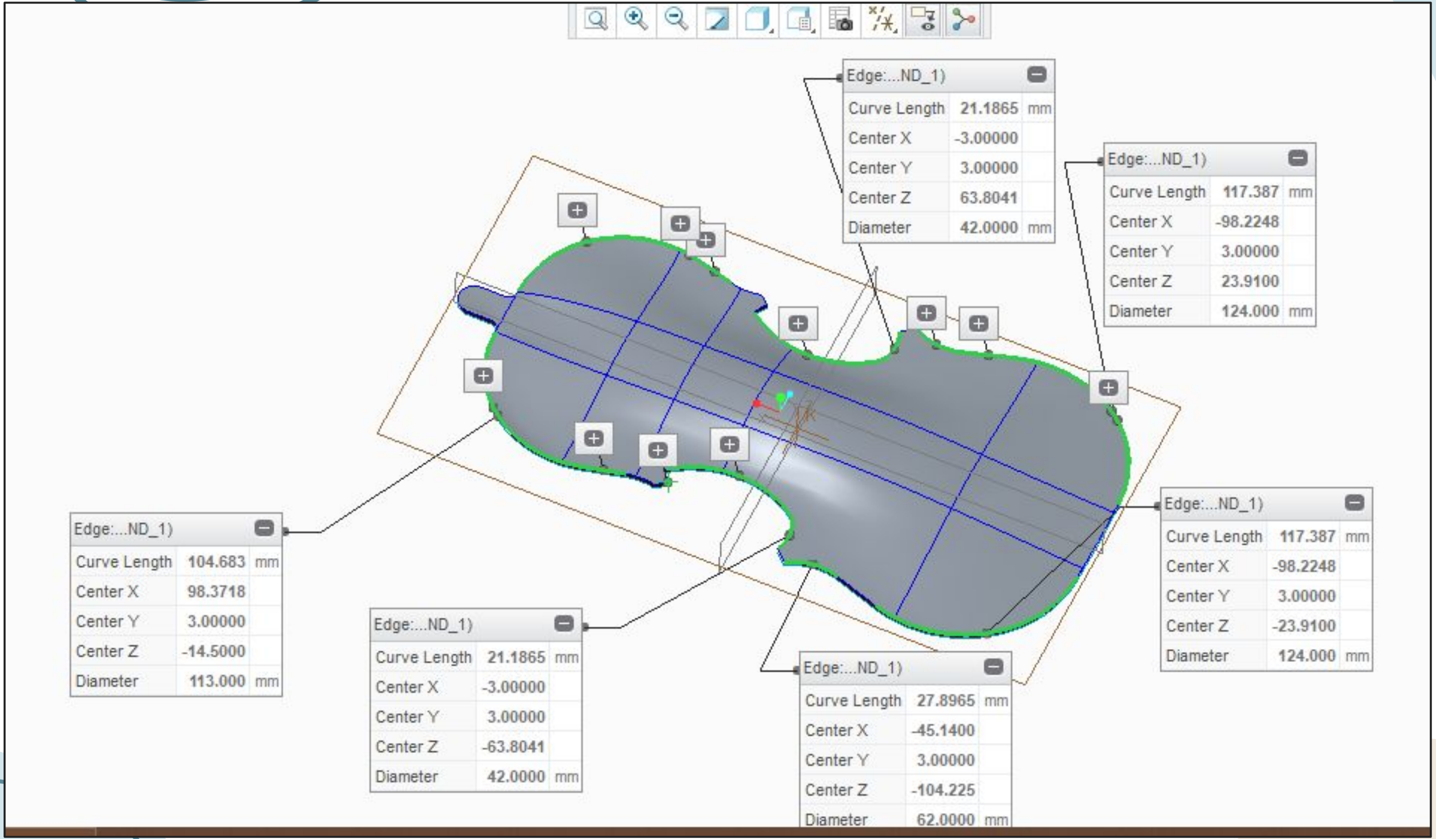











Edge:...ND_1)
Curve Length 21.1865 mm
Center X -3.00000
Center Y 3.00000
Center Z 63.8041
Diameter 42.0000 mm
Edge:...ND_1)
Curve Length 117.387 mm
Center X -98.2248
Center Y 3.00000
Center Z 23.9100
Diameter 124.000 mm
Edge:...ND_1)
Curve Length 104.683 mm
Center X 98.3718
Center Y 3.00000
Center Z -14.5000
Diameter 113.000 mm
Edge:...ND_1)
Curve Length 21.1865 mm
Center X -3.00000
Center Y 3.00000
Center Z -63.8041
Diameter 42.0000 mm
Edge:...ND_1)
Curve Length 27.8965 mm
Center X -45.1400
Center Y 3.00000
Center Z -104.225
Diameter 62.0000 mm
Edge:...ND_1)
Curve Length 117.387 mm
Center X -98.2248
Center Y 3.00000
Center Z -23.9100
Diameter 124.000 mm

MANUFACTURING

Manufacturing engineering is the practical application of engineering principles to turn digital designs into physical products. It involves various processes such as traditional and advanced machining, which utilize tools like lathes, milling machines, and CNC machines. In CNC manufacturing, a code written in the CAM programming language is required to operate the machine and create the final product.

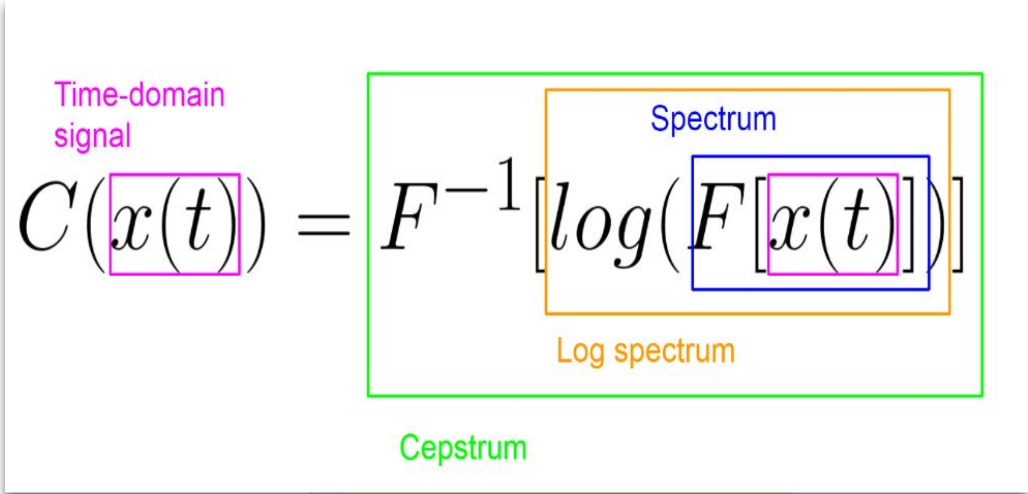
CODE GENERATION

The Creo software includes a distinct interface called "manufacturing" which is used to convert design models into codes. This interface includes various operations such as roughing, finishing, profile milling, drilling, and trajectory. Once an operation is selected, specific parameters such as feed rate, depth of cut, step over, spindle speed, clearance, surface, tolerance, and tool path need to be entered.

The software generates three types of codes for the selected operations, which are:

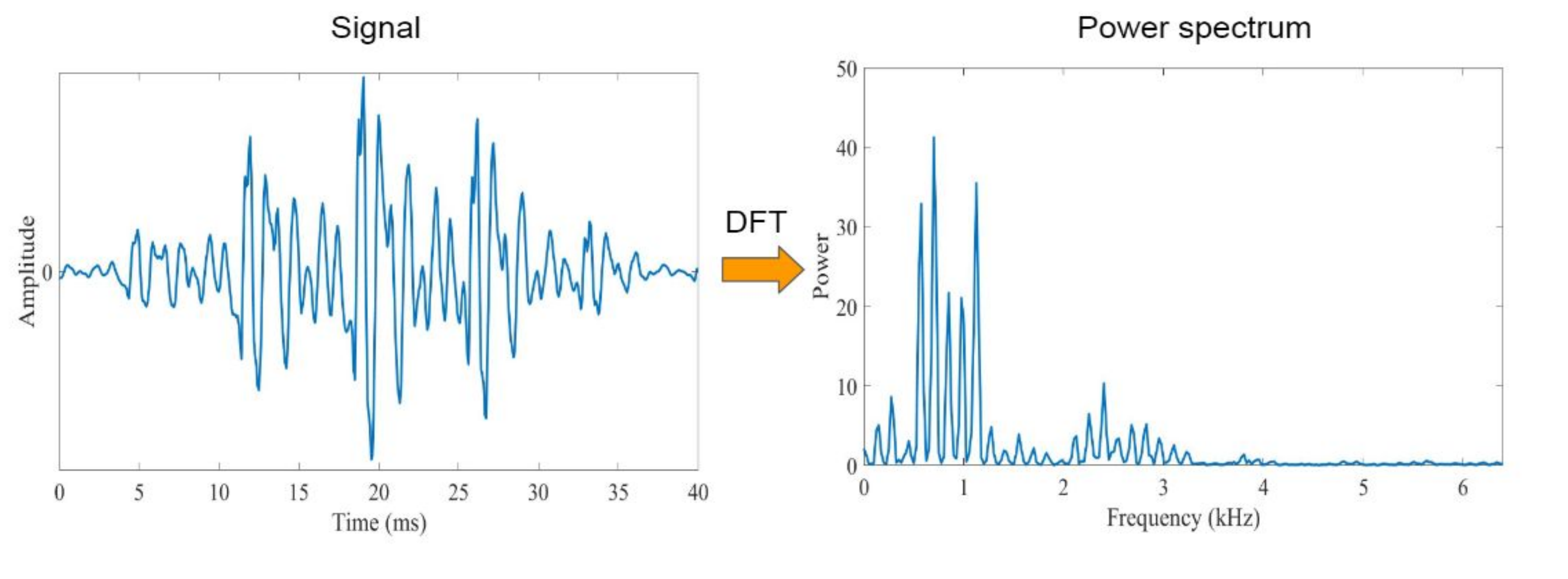
- roughing
- surface finishing
- profile milling

SOUND ANALYSIS

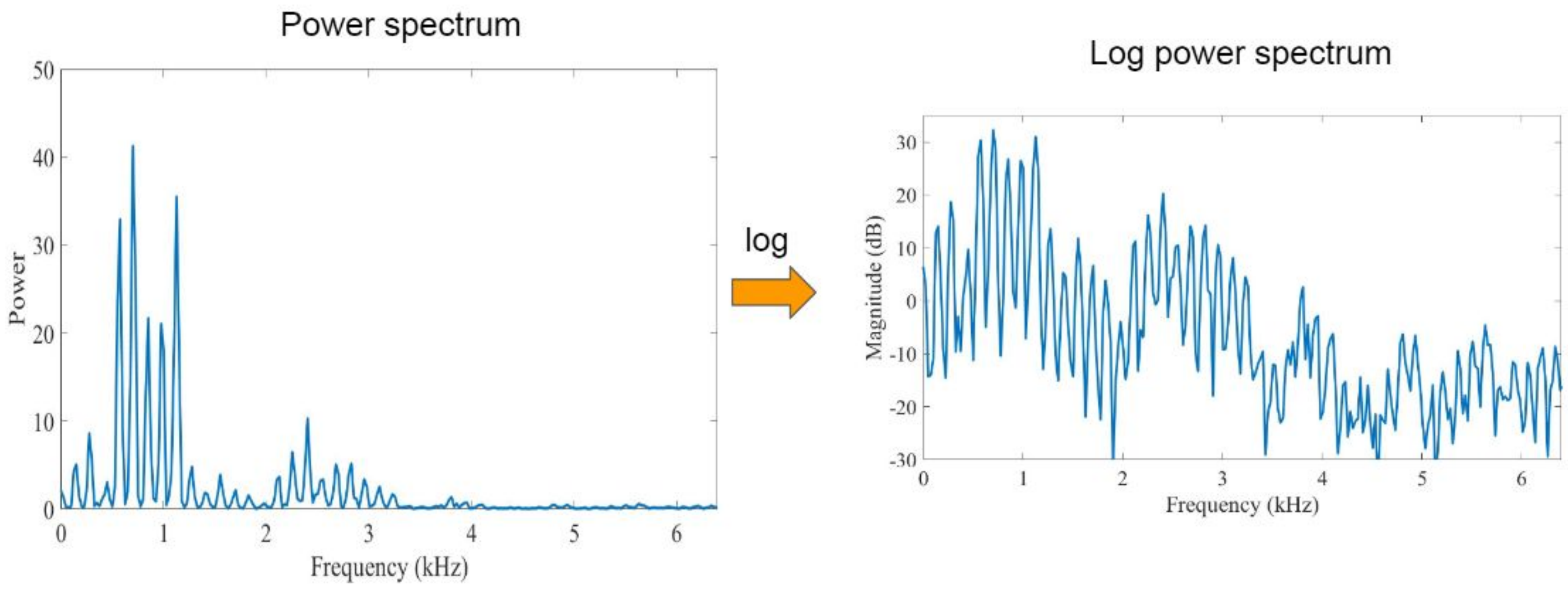


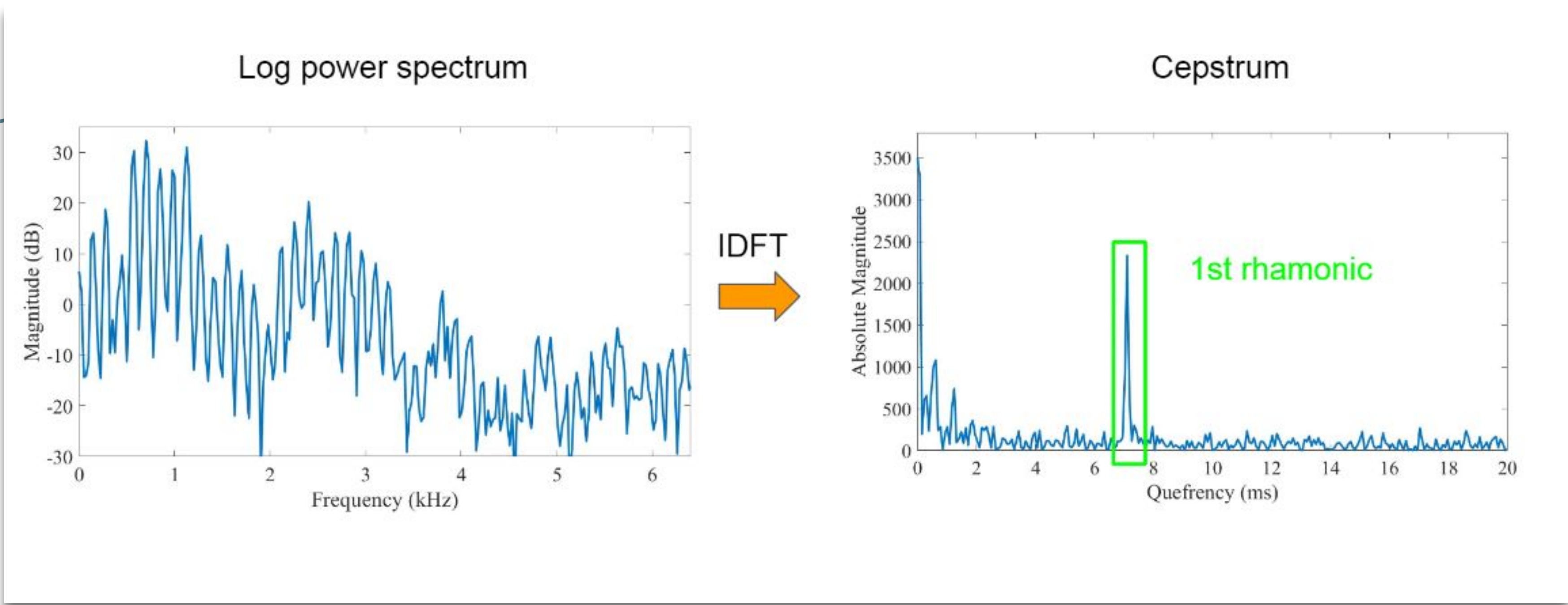
Computing the cepstrum

From the formula we shall define what is a cepstrum.



Signal
Amplitude
0
0
5
10
15
20
25
30
35
40
Time (ms)
DFT
Power spectrum
Power
50
40
30
20
10
0
0
1
2
3
4
5
6
Frequency (kHz)


Power spectrum
Power
Frequency (kHz)
log
Log power spectrum
Magnitude (dB)
Frequency (kHz)


Log power spectrum
Cepstrum
Magnitude (dB)
30
20
10
0
-10
-20
-30
0
1
2
3
4
5
6
Frequency (kHz)
IDFT
Absolute Magnitude
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
16
18
20
Quefrency (ms)
1st rhamonic


Spectrum of a spectrum
Cepstrum
imgflip.com

Implementation Of MFCC analysis

```python
import os
import librosa
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras import models, layers
```

```python
good_sounds_dir = '/content/drive/MyDrive/GoodSounds'
bad_sounds_dir = '/content/drive/MyDrive/Bad_sounds'


sr = 44100   #22.05 Khz
duration = 5


good_sounds = []
bad_sounds = []
```

Implemented the algorithm in python with help of Librosa library.

Deep learning neural network, CNN.

To create stack of neural network layers we will use the Keras API.

Sampling rate is kept at 44.10 KHz.

```python
for filename in os.listdir(good_sounds_dir):
    if filename.endswith('.wav'):
        filepath = os.path.join(good_sounds_dir, filename)
        sound, _ = librosa.load(filepath, sr=sr, duration=duration)
        mfcc = librosa.feature.mfcc(y=sound, sr=sr, n_mfcc=20)
        good_sounds.append(mfcc)


for filename in os.listdir(bad_sounds_dir):
    if filename.endswith('.wav'):
        filepath = os.path.join(bad_sounds_dir, filename)
        sound, _ = librosa.load(filepath, sr=sr, duration=duration)
        mfcc = librosa.feature.mfcc(y=sound, sr=sr, n_mfcc=20)
        bad_sounds.append(mfcc)


label_encoder = LabelEncoder()
labels = ['good'] * len(good_sounds) + ['bad'] * len(bad_sounds)
labels = label_encoder.fit_transform(labels)
```

From all the GoodSounds we extract the MFCC features.

We recognised that no. of features that are useful lie in the range 13-20.

A similar process is done for the Bad_sounds.

Moving towards training our model we encode each of our feature as Good or bad.

```python
X_train, X_test, y_train, y_test = train_test_split(good_sounds + bad_sounds, labels, test_size=0.2, random_state=42)


X_train = np.array(X_train)
X_test = np.array(X_test)


X_train = X_train.reshape((*X_train.shape, 1))
X_test = X_test.reshape((*X_test.shape, 1))
```

The obtained Data is now split into training and testing sets.

Test size is fixed at 0.2 which means that 80% of data is used for training and remaining 20% for testing.

The training and testing can be modified according to the increasing complexities of sounds.

```python
model = models.Sequential([

    layers.Conv2D(32, (3, 3), activation='relu', input_shape=X_train.shape[1:]),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(1, activation='sigmoid')
])
```

Our model is build with the help of the Keras sequential API.

This model consist of two main convolution layers, each with 32 and 64 filters.
The output of each layer produces a output feature map.

A total of 64 feature maps are identified for each sample.
We can increase the no. of features to be extracted b

CHALLENGES

First, we had to figure out how to create two surfaces with different curvatures and thicknesses for the back plate on the software.

Secondly, we experienced issues with generating incorrect code parameters that resulted in poor surface quality during CNC manufacturing. To avoid this, we needed to have proper knowledge of the parameters for each operation.

Additionally, we struggled to achieve the desired back plate surface thickness during manufacturing.

We also had to ensure that the contour and edges were precisely aligned.

Finally, as wood is a soft material, we faced challenges with clipping during long operations due to machine vibrations. It was sometimes difficult to control the vibrations and prevent them from affecting the wood workpiece.

CONCLUSION

Currently, we have completed the manufacturing of the violin plates and have also developed a sound analysis code to evaluate various speech signals. This will enable us to test the sound signals on the newly manufactured violin.

In conclusion, this project was a unique challenge for us as it involved combining musical instruments with unconventional manufacturing methods.

THANK YOU!