Task 1 – write class with the following functions for a fft data analysis and execute the functions at the end of the python file

1. read data from Test\_data.txt file line by line with open() command

input: path to file, filename, sensor names list

output: data array

additional information:
read all sensors if sensor names list is empty

in datafile tab-delimiter is used

format of data array:

data array [sensor number X][0] - times

data array [sensor number X][1] - values

data array [sensor number X][2][0] – sensor name

1. make fft of data array with numpy fft.rfft/fft.rfftfreq

input: data array, start time ,frequency resolution, time interval for averaging
output: output data array

format of output data array:

output data array [sensor number X][0] - frequencies

output data array [sensor number X][1] - values

output data array [sensor number X][2] - angles

output data array [sensor number X][3][0] – sensor name

1. write plot with matplotlib.pyplot
input: output data array, output folder, output file name
optional input: xlabel, ylabel, xticks, yticks, xlim, ylim

output: lineplot in png format for each sensor in output data array saved in output folder with name “output file name.png”

Task 2 – write a function within the python file Machine\_modelling\_Tool.py within the class Coil(Main\_collection) based on the existing old classes insulation\_calc() and Coil\_calc\_2D()

1. function geometry\_2d(self)
**input:**

*machine data - lamination*
self. l\_Z2\_bn slot width

self. l\_Z2\_hn slot height

*machine data - coil*

n\_parallels\_v number of parallel strands in vertical direction (currently 1)

n\_parallels\_h number of parallel strands in horizontal direction

n\_W\_turns number or turns of the strands

l\_slot\_clearance\_v not usable slot height

l\_slot\_clearance\_h not usable slot width

l\_copper\_wX copper width of strand number X (relative values, sum(l\_copper\_wX) is 100% of available copper width)

*insulation data matrix*

self.coil\_insulation\_system

**output:**

the following output is calculated for pressed and unpressed condition

*for each strand:*
variable:
copper height, insulated strand height, copper rounding radius

array[horizontal strand number]:
copper area, copper width, insulated strand width

*for whole coil:*

variable:
consolidated/insulated/protected coil height/width

consolidated/insulated/protected coil area

slot area

filling factor copper/insulated coil area, filling factor copper/slot area



Task 3 – write a function within the python file Machine\_modelling\_Tool.py within the class Coil(Main\_collection) based on parts of the existing old class Coil\_calc\_3D()

1. function geometry\_3d(self)
**input:**

required machine data can be retrieved as described in Task2
self. l\_fe active part length

self. l\_str\_ew straight length outside the active part region before bending

self. l\_terminal winding terminal (straight part outside active region+ending)

insulation data matrix can be retrieved as described in Task2

material data properties

self.mat\_X\_Y X material name, Y material property

e.g. self.mat\_Copper\_gamma20

variables created in the function geometry 2d(self) as self.new\_variable can be retrieved the same way in the new function

**output:**

*for each strand:*
array[horizontal strand number]:

copper length, resistance

*for whole coil:*
variable:

resistance

