IoT BASED TOOL WEAR PREDICTION

MINOR PROJECT REPORT

Submitted by

SUBHAM AGRAWAL [Regno. RA1611018010074]

MAITREYA KULKARNI [Regno. RA1611018010078]

EMMANUEL.P.SINGSON [Regno. RA1611018010094]

Under the guidance of

G.MADHUMITHA, M.E Mechatronics
(Assistant Professor, Department of Mechatronics Engineering)

of

SIXTH SEMESTER (15MH375L)

in

MECHATRONICS ENGINEERING

of

FACULTY OF ENGINEERING & TECHNOLOGY

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

S.R.M. Nagar, Kattankulathur, Kancheepuram District
APRIL 2019

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled **"PREDICTION OF TOOL WEAR USING IoT"** is the bonafide work of "**SUBHAM AGRAWAL [Regno. RA1611018010074], MAITREYA KULKARNI [Regno. RA1611018010078], EMMANUEL.P.SINGSON [Regno. RA1611018010094]",** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
G.MADHUMITHA
GUIDE
Assistance professor (O.G)
Dept. of Mechatronics Engineering

SIGNATURE
Dr.G.MURALI Ph.D.
HEAD OF THE DEPARTMENT
Dept. of Mechatronics Engineering

Signature of the Internal Examiner

Signature of the External Examiner

ABSTRACT

Prediction of tool wear by monitoring the vibrations generally based on the concept, where the vibration created during the machining process is in correlation with the tool wear phenomenon. The prior detection of tool wear phenomenon can provide increase of performance in process of machining. Substantial amount of variations in the vibrations of the machine and tool are acquired through the MPU-6050 sensor and is uploaded to a cloud server. Relationship was established between the speed, depth of cut, feed rate and vibrations. With all the values recorded from the cloud server a machine learning mode was trained to predict the tool wear prior to the occurrence of this phenomenon. These results provide initial elements towards the implementation of online monitoring and predictive maintenance of the tools and machine.

ACKNOWLEDGEMENTS

We would like to thank our Professor and Head of the Department Dr. G. Murali Ph.D. for providing us with all the support for the successful completion of our Project. Also we would like to express our deepest gratitude to Dr. M. Gopal for providing us the support and arranging the necessary machines. Also we would like to express deepest gratitude to our guide, Ms. G.Madhumitha M.E for her valuable guidance, consistent encouragement, personal caring, timely help and providing me with an excellent atmosphere for doing the research. All through the work, in spite of her busy schedule, she has extended cheerful and cordial support to us for completing this Project.

Subham Agrawal

Maitreya Kulkarni

Emmanuel.P.Singson

ABBREVIATIONS

IoT – Internet of Things
KNN – K Nearest Neighbor
ML – Machine Learning
SVM – Support Vector Machine

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

With the advent of Industry 4.0 for current automation and data exchange in all fields of Engineering. This includes cyber-physical systems, the IoT, cloud computing and cognitive computing. ‘Smart factory’ can be created with the arrival of Industry 4.0. Over the Internet of Things, cyber –physical systems communicate and co-operate with each other and other humans in real time, and via cloud computing, both internal and cross-organizational services are offered and used by participants of value chain.

There are four design principles in Industry 4.0. These principles support companies in identifying and implementing Industry 4.0 scenarios.

• **Interoperability**: The ability of machines, devices, sensors, and people to connect And communicate with each other via the Internet of Things (IoT) or the Internet of People (IoP).

• **Information transparency**: The ability of information systems to create a virtual Copy of the physical world by enriching digital plant models with sensor Data. This requires the aggregation of raw sensor data to higher-value context Information.

• **Technical assistance**: First, the ability of assistance systems to support humans By aggregating and visualizing information comprehensibly for making informed Decisions and solving urgent problems on short notice. Second, the ability of Cyber physical systems to physically support humans by conducting a range of Tasks that are unpleasant, too exhausting, or unsafe for their human co-workers.

• **Decentralized decisions**: The ability of cyber physical systems to make decisions On their own and to perform their tasks as autonomously as possible.

The Internet of Things (IoT) is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these objects to connect and exchange data.
Each thing is uniquely identifiable through its embedded computing system but is able to inter-operate within the existing Internet infrastructure. The IoT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities of or more direct integration of the physical world into computer based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human an intervention.

When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, virtual power plants, smart homes, intelligent transportation and smart cities.

1.1.1 OBJECTIVE

The objective of the project is to develop an intelligent module for tool shank with the help of sensors and control logic and machine learning algorithms for prediction of tool wear using vibrations that occur during the machining process. The idea is also try and implement a similar system in a Lathe machine and predict the tool wear.

1.1.2 NEED OF THE PROJECT

- For easier and prior detection of tool wear in Lathe Machine.
- Improving surface finish of the work piece.
- Increasing the efficiency and accuracy of the Lathe Machine

1.2 MACHINE VIBRATIONS

Machining vibrations also called chatter, correspond to the relative motion between the work piece and the cutting tool. The vibrations gives rise to the waves on the machined surface. This affects typical machining processes, such as turning, milling, drilling and grinding. A chatter marks are irregular and the surface finish is affected by the grinding wheel which is out of true or regular mark left when turning a long piece on a Lathe with tool that is not sharp, due to machining vibrations.

The use of high speed machining (HSM) has enabled an increase in productivity and the realization of work pieces that were impossible before. Due to high dynamic movements in these machines the machine centres are less rigid. In many applications, i.e. long tools, thin work pieces, the appearance of vibrations is the most limiting factor and compels the machinist to reduce cutting speeds and feeds well below the capacities of machines or tools.

Vibrations result in noise bad surface finish and too failure. The main source of the vibrations are of 2 types: forced vibrations and self-generated vibrations. Forced vibrations are mainly generated by interrupted cutting (inherent to milling), run out, or vibrations from outside the machine. Self-generated vibrations are related to the fact that the actual chip thickness depends also on the relative position between tool and work piece during the previous tooth passage. Thus increasing vibrations may appear up to levels which can seriously degrade the machined surface quality. Hard spots or a crust in the material being machined impart small shocks to the tool and work piece, as a result of which free vibrations are set up. When machining is done under conditions resulting in discontinuous chip removals, the segmentation of chip elements results in a fluctuation of the cutting thrust. If the frequency of these fluctuations coincides with one of the natural frequencies of the structure, forced vibration of appreciable amplitude may be excited. However, in single edge cutting operations. Whether the segmentation of the chip is a primary effect or whether it is produced by other vibration, without which continuous chip flow would be encountered.

The breaking away of a built-up edge from the tool face also imparts impulses to the cutting tool which result in vibration. However, marks left by the built-up edge on the machined surface are far more pronounced than those caused by the ensuing vibration; it is probably for this reason that the built-up edge has not been studied from the vibration point of view. The built-up edge on the machined surface are far more pronounced than those caused by the ensuing vibration; it is probably for this reason that the built-up edge has not been studied from the vibration point of view. The built-up edge frequently accompanies certain types of vibration, and instances have been known when it disappeared as soon as the vibration was eliminated.

1.3 LATHE MACHINE

The lathe is a machine tool which holds the work piece between two rigid and strong supports called centers or in a chuck or face plate which revolves. The cutting tool is rigidly held and supported in a tool post which is fed against the revolving work. The normal cutting operations are performed with the cutting tool fed either parallel or at right angles to the axis of the work.The cutting tool may also be fed at an angle relative to the axis of work for machining tapers and angles.

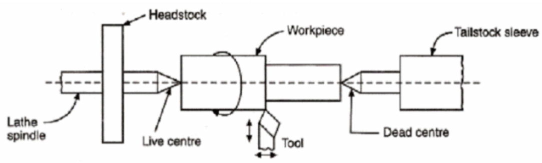

Figure 1.1: CUTTING TOOL ANGLE

1.3.1 LATHE MACHINE CONSTRUCTION

The main parts of the lathe are the bed, headstock, quick changing gear box, carriage and tailstock.

Bed

The bed is a heavy, rugged casting in which are mounted the working parts of the lathe. It carries the headstock and tail stock for supporting the work piece and provides a base for the movement of carriage assembly which carries the tool.

Legs

The legs carry the entire load of machine and are firmly secured to floor by Foundation bolts.

Headstock

The headstock is clamped on the left hand side of the bed and it serves as housing for the driving pulleys, back gears, headstock spindle, live centre and the feed reverse gear. The headstock spindle is a hollow cylindrical shaft that provides a drive from the motor to work holding devices.

Gear Box

The quick-change gear-box is placed below the headstock and contains a number of different sized gears.

Carriage

The carriage is located between the headstock and tailstock and serves the purpose of supporting, guiding and feeding the tool against the job during operation. The main parts of carriage are:

- **The saddle**: It’s an H-shaped casting mounted on the top of lathe ways. It provides support to cross-slide, compound rest and tool post.

• **The cross slide**: which is mounted on the top of saddle, and it provides a mounted or automatic cross movement for the cutting tool.

• **The compound rest**: The compound rest is fitted on the top of cross slide and is used to support the tool post and the cutting tool.

• **The tool post**: Which is mounted on the compound rest, and it rigidly clamps the cutting tool or tool holder at the proper height relative to the work centre line.

• **The apron**: The apron is fastened to the saddle and it houses the gears, clutches and levers required to move the carriage or cross slide. The engagement of split nut lever and the automatic feed lever at the same time is prevented she carriage along the lathe bed.

Tailstock

The tailstock is a movable casting located opposite the headstock on the ways of the bed. The tailstock can slide along the bed to accommodate different lengths of workpiece between the centers. A tailstock clamp is provided to lock the tailstock at any desired position. The tailstock spindle has an internal taper to hold the dead centre and the tapered shank tools such as reamers and drills.

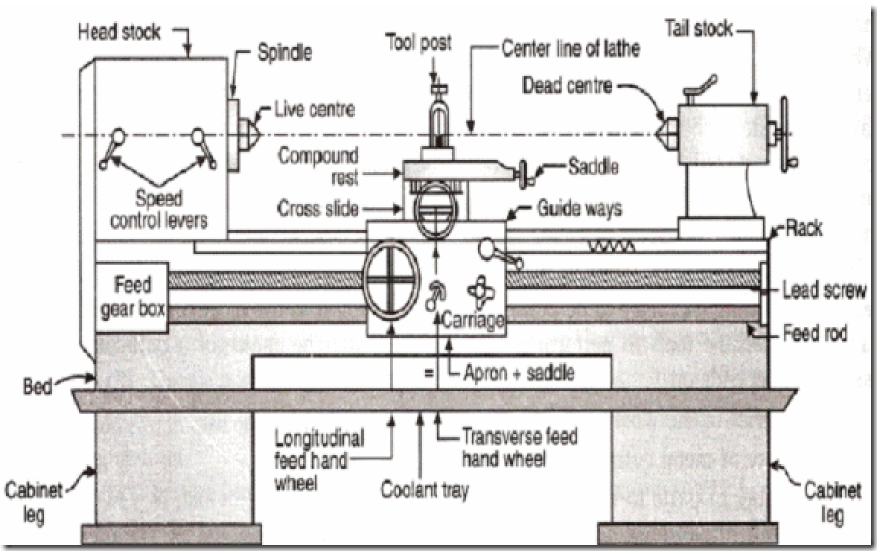

Figure 1.2: LATHE MACHINE CONSTRUCTION

CHAPTER 2

LITERATURE SURVEY

2.1 THE CONCEPT AND PROGRESS OF INTELLIGENT SPINDLE

Hongrui cao, xingwu zhang, xuefeng chen, China 17 October 2016. This paper provided the clarification for concept of intelligent spindles and provide an in- depth review of the state-of-the-art of related technologies. A new integrated concept for intelligent spindles is proposed and the key enabling technologies and expected intelligent functions.They also made relevant research that may be beneficial to the development of intelligent spindles is reviewed from six thrust areas, which include monitoring and control of tool condition, chatter, spindle collision, temperature/thermal error, spindle balance, and spindle health. The capabilities of sensing, decision making and control, which guarantee the optimum machining process. The sensing module, displacement sensors and accelerometers are integrated into the spindle structure to measure the vibration signals of the tool holder and spindle housing, and thermocouples are employed to measure the temperature of the motor and all of the bearings. Afterwards, the measured signals are imported into the decision-making and control modules as inputs with the decision-making module, the measured signals are analysed by advanced signal processing techniques.

2.2 TOOL WEAR IN TURNING: NUMERICAL AND EXPERIMENTAL ANALYSIS

Balla Srinivasa Prasad, M. Prakash Babu, Mechanical Engineering, Visakhapatnam, India (July 2016). A correlation between vibration amplitude and tool wear when in dry turning of steel using uncoated carbide insert is analysed with experiments and finite element simulations. Results are utilized to predict the evolution of cutting forces, vibration displacement amplitudes and tool wear in vibration in turning with relative vibration and tool wear with the variation of process parameters.

The cutting forces in the feed direction are also predicted and compared with the experimental trends. A laser Doppler vibrometer is used to detect vibration amplitudes and the usage of dynamometer for recording the cutting forces during the cutting process is demonstrated in this experimental aspect. They also showed their investigation that influence of spindle speed, feed rate, depth of cut on vibration amplitude and tool flank wear at different levels of workpiece hardness. They also developed empirical models for second order polynomial equations for correlating of various process parameters. Analysis of variance is carried out to identify the significant factors that are affecting the vibration amplitude and tool flank wear and surface methodology is also implemented to investigate the progression of flank wear and displacement amplitude based on experimental data. These data confirms the close relation between experimental values and numerical values in evaluating the tool wear.

2.3 INTELLIGENT MATLAB MODEL FOR PREDICTING OF VARIOUS MACHINING PROCESSES

Vivekananda Reddy, Mythra Varun A.V,MVGR, Kakinada (February 2016). We learned about Surface finish which is an important factor in determining the satisfactory functioning of the machined components. Surface finishing improves appearance, adhesion, and corrosion resistance, wear resistance, hardness, modify electrical conductivity, remove burrs and other surface flaws, and control the surface friction. With the advancement in image processing techniques have basis for developing image-based surface finish measuring techniques. Using a CCD camera and polychromatic light source, low incident- angle images of machined surfaces with different surface finish values were captured. The RGB values of the images are produced and were processed using MATLAB software. A database of reference images with known surface finish values was established. Here an attempt is made to simultaneously classify the machined piece into a category as well as to predict its surface finish, using signal vector regression models developed in the MATLAB by suitably training them with machined surface finish values and corresponding RGB values obtained from Image Processing.

2.4 ACOUSTIC EMMISION METHOD FOR TOOL WEAR MONITERING

Xiaoli Li, department of Electric Engineering, China (July 2001). In this journal they used the effectiveness of acoustic emission (AE)-based sensing methodologies for machine condition analysis and process monitoring. Acoustic emission (AE) is the class of phenomena whereby transient elastic waves are generated by the rapid release of energy from a localized source or sources within a material, or the transient elastic wave(s) so generated. They clearly showed, an AE is a sound wave or, more properly, a stress wave that travels through a material as the result of some sudden release of strain energy. AE-based tool wear condition monitoring in turning, which includes AE signal generation and correction in cutting processes, AE signal processing, and tool wear estimation. To estimation the tool wear condition they used, pattern classification, GMDH methodology, fuzzy classifier, neural network, and sensor and data fusion.

CHAPTER 3

SYSTEM DESIGN

3.1 COMPONENTS

3.1.1 NodeMCU Wi-Fi MODULE

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS.

NodeMCU has 14 active pins with a pitch of 0.1 in. The antenna present in the NodeMCU is a PCB trace antenna. A PCB trace antenna has large bandwidth and hence are used for establishing robust communication between 2 microcontrollers or between microcontroller and the cloud platform. Uses the ESP-12 module; includes USB to serial interface.

Fig. 3.1 NodeMCU bottom

Specification

- Power supply: 5V to 12V
- Working current: 80mA to 200mA

- IO pins : 13 GPIO pins including Transmitter and Reciever
- Range : 10m indoors Line of Sight
- Data Rate:115200 baud rate
- Operating Temperature: -40ºC to 125ºC

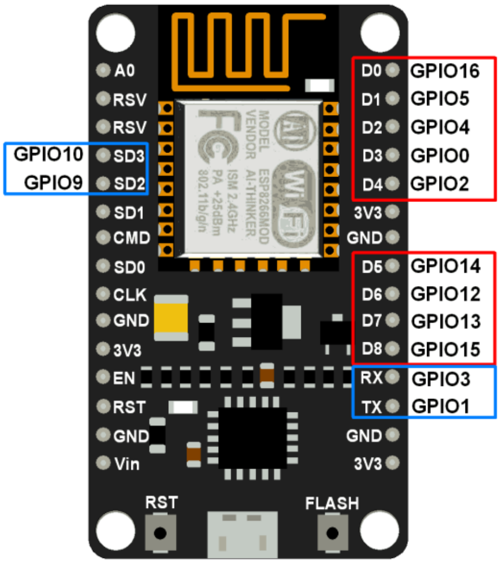

Fig. 3.2 NodeMCU pins

TABLE 3.1 NodeMCU PIN FUNCTIONS

Pin Names on NodeMCU Development Kit	ESP8266 Internal GPIO Pin number
D0	GPIO16
D1	GPIO5
D2	GPIO4
D3	GPIO0
D4	GPIO2
D5	GPIO14
D6	GPIO12
D7	GPIO13
D8	GPIO15
D9/RX	GPIO3
D10/TX	GPIO1
D11/SD2	GPIO9
D12/SD3	GPIO10

3.1.2 MPU-6050 ACCELEROMETER + GYRO

MPU-6050 sensor contains a MEMS accelerometer and a MEMS gyro in a single chip.It is very accurate, as it contains 16-bits analog to digital conversion hardware for each channel. Therefor it captures the x, y, and z channel at the same time. The sensor uses the I2C-bus to interface with the Arduino. The MPU-6050 is not expensive, especially given the fact that it combines both an accelerometer and a gyro. The MPU-6050 features three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs and three 16-bit ADCs for digitizing the accelerometer outputs. For precision tracking of both fast and slow motions, the parts feature a user-programmable gyroscope full-scale range of _250, _500, _1000, and _2000 degree/sec (dps) and a user-programmable accelerometer full-scale range of _2g, _4g, _8g, and _16g.

An on-chip 1024 Byte FIFO buffer helps lower system power consumption by allowing the system processor to read the sensor data in bursts and then enter a low-power mode as the MPU collects more data. With all the necessary on-chip processing and sensor components required to support many motion-based use cases, the MPU-60X0 uniquely enables low-power Motion Interface applications in portable applications with reduced processing requirements for the system processor. By providing an integrated Motion Fusion output, the DMP in the MPU-60X0 offloads the intensive Motion Processing computation requirements from the system processor, minimizing the need for frequent polling of the motion sensor output. Communication with all registers of the device is performed using either I2C at 400 kHz or SPI at 1MHz (MPU-6000 only). For applications requiring faster communications, the sensor and interrupt registers may be read using SPI at 20MHz (MPU-6000 only). Additional features include an embedded temperature sensor and an on-chip oscillator with _1 percentage variation over the operating temperature range.

Fig.3.3 MPU-6050 ACCELEROMETER

3.1.2.1 Accelerometer Features

The triple-axis MEMS accelerometer in MPU-60X0 includes a wide range of features:

• Digital-output triple-axis accelerometer with a programmable full scale range of $\pm 2g$, $\pm 4g$, $\pm 8g$ and $\pm 16g$

• Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer

• Accelerometer normal operating current: 500µA

• Low power accelerometer mode current: 10µA at 1.25Hz, 20µA at 5Hz, 60µA at 20Hz, 110µA at 40Hz

• Orientation detection and signalling

• Tap detection

• User-programmable interrupts

• High-G interrupt

3.1.2.2 Clocking

• On-chip timing generator ±1% frequency variation over full temperature range

• Optional external clock inputs of 32.768 kHz or 19.2 MHz

TABLE 3.2 ACCELEROMETER SPECIFICATIONS

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY						
Full-Scale Range	AFS_SEL=0		±2		*g*	
	AFS_SEL=1		±4		*g*	
	AFS_SEL=2		±8		*g*	
	AFS_SEL=3		±16		*g*	
ADC Word Length	Output in two's complement format		16		bits	
Sensitivity Scale Factor	AFS_SEL=0		16,384		LSB/*g*	
	AFS_SEL=1		8,192		LSB/*g*	
	AFS_SEL=2		4,096		LSB/*g*	
	AFS_SEL=3		2,048		LSB/*g*	
Initial Calibration Tolerance			±3		%	
Sensitivity Change vs. Temperature	AFS_SEL=0, -40°C to +85°C		±0.02		%/°C	
Nonlinearity	Best Fit Straight Line		0.5		%	
Cross-Axis Sensitivity			±2		%	
ZERO-G OUTPUT						
Initial Calibration Tolerance	X and Y axes		±50		m*g*	1
	Z axis		±80		m*g*	
Zero-G Level Change vs. Temperature	X and Y axes, 0°C to +70°C		±35			
	Z axis, 0°C to +70°C		±60		m*g*	
SELF TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	2
NOISE PERFORMANCE						
Power Spectral Density	@10Hz, AFS_SEL=0 & ODR=1kHz		400		µ*g*/ √Hz	
LOW PASS FILTER RESPONSE						
	Programmable Range	5		260	Hz	
OUTPUT DATA RATE						
	Programmable Range	4		1,000	Hz	
INTELLIGENCE FUNCTION INCREMENT			32		m*g*/LSB	

The MPU-6050 is 3-Axis accelerometer uses separate proof masses for each axis. Acceleration along a particular axis induces displacement on the corresponding proof mass, and capacitive sensors detect the displacement differentially. The MPU-6050 is architecture reduces the accelerometer susceptibility to fabrication variations as well as to thermal drift. When the device is placed on a flat surface, it will measure 0g on the X and Y axes and +1g on the Z axis.

CHAPTER 4

METHODOLOGY

The strategy of tool wear prediction using vibrations generally based on the concept, where the vibrations caused due to the machining process is correlated with the tool wear phenomenon. In order to provide a robust system that can predict the tool wear, thus increasing the machining performance. To overcome the problem a prototype was designed and assembled that could be attached to tool shank and the experiments were conducted on a conventional lathe machine. Measured parameters were acceleration for detection of vibrations in all the three axes which were measured for different inputs of machining parameters.

4.1 FLOWCHART

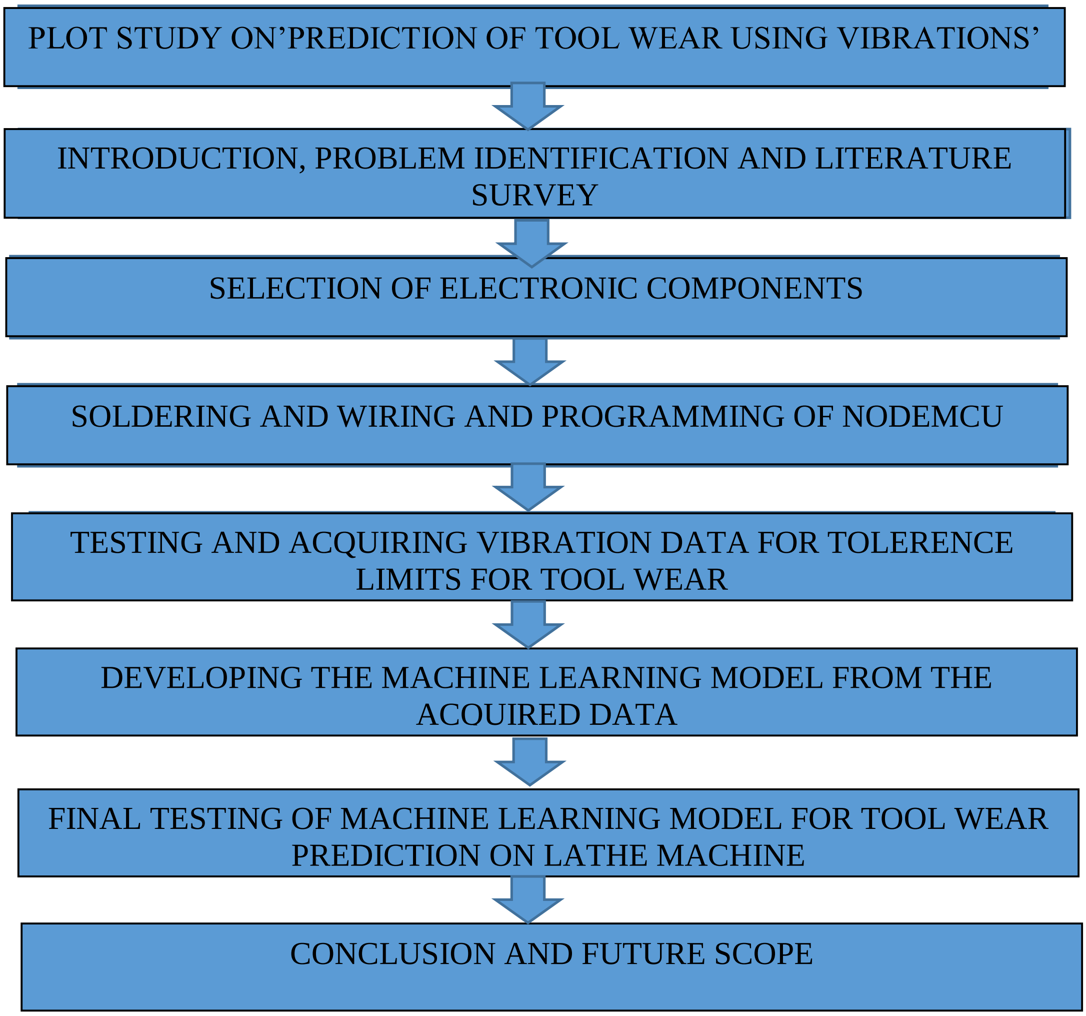

Fig. 4.1 Flowchart

4.2 MODULE [TOOL VIBRATION MONITOR]

ELECTRONIC COMPONENTS IN MODULE

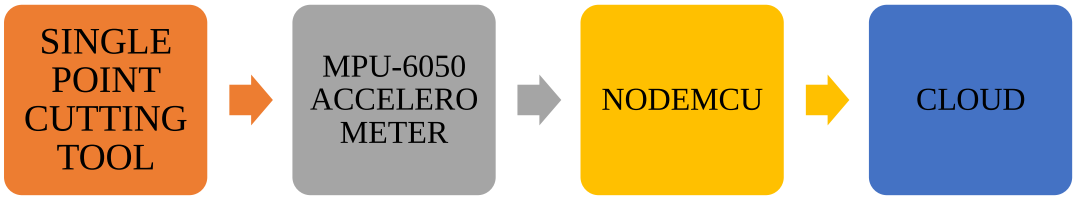

Fig. 4.2 Electronic components in module

In this module a NodeMCU and an accelerometer. NodeMCU is connected to laptop using a USB cable. Power to all the components is supplied by the laptop through the USB cable. The accelerometer is placed in tool shank of the single point cutting tool in the conventional lathe machine. The accelerometer is connected to NodeMCU using jumper wires which is placed on the side of the lathe machine. The function of the accelerometer is to detect the vibrations occurring during the machining process. The vibration data is obtained in digital mode and is processed by the NodeMCU. The output from the sensor is in the unit g. The data is sent wirelessly to a cloud platform that stores and displays the data. As the data is transmitted wirelessly there is no disturbance during the transmission.

4.3 TOOL AND WORK PIECE

SINGLE POINT CUTTING TOOL

The single point cutting tool has only one cutting point or edge. These tools used for turning, boring, shaping or planning operations. These tools used on lathe, boring and shaper machines. A single point cutting tool consists of a sharpened cutting part and the shank and main parts or elements which are

1: Shank

It is the main body of the tool.

2: Flank

The surface or surfaces below the adjacent to the cutting edge is called flank of the tool.

3: Face

The surface on which the chip slides is called the face of the tool.

4: Heel

It is the intersection of the flank and the base of the tool.

5: Nose

It is the point where the side cutting edge and end cutting edge intersect.

6: Cutting Edge

It is the edge on the face of the tool which removes the material from the work piece. The cutting edge consists of the side cutting edge (major cutting edge) and cutting edge (minor cutting edge) and the nose.

Fig. 4.3 SINGLE POINT CUTTING TOOL

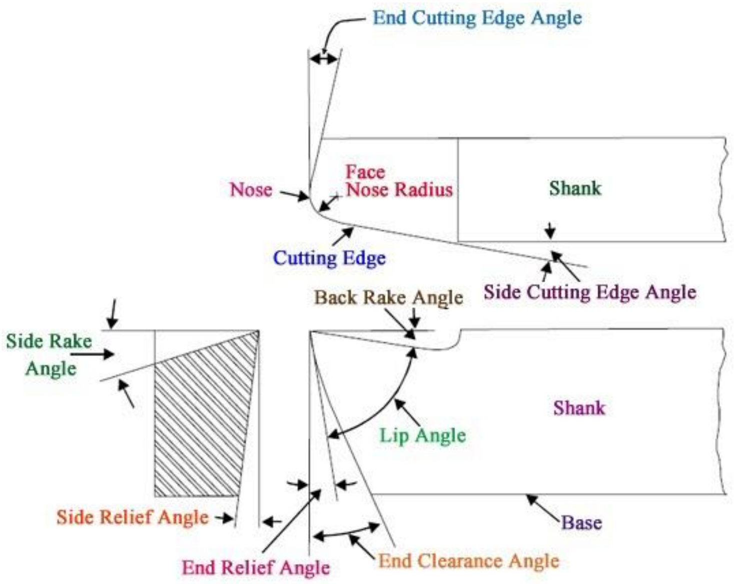

Fig. 4.4 SINGLE POINT CUTTING TOOL NOMENCLATURE

Fig. 4.5 Experimental Work piece

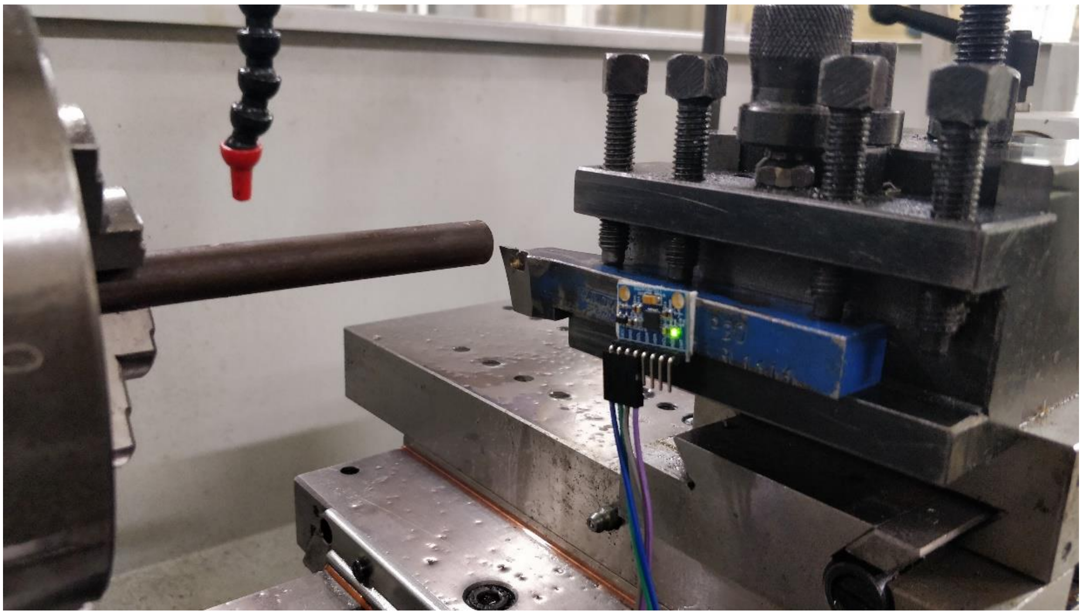

Fig 4.6 Module on Lathe

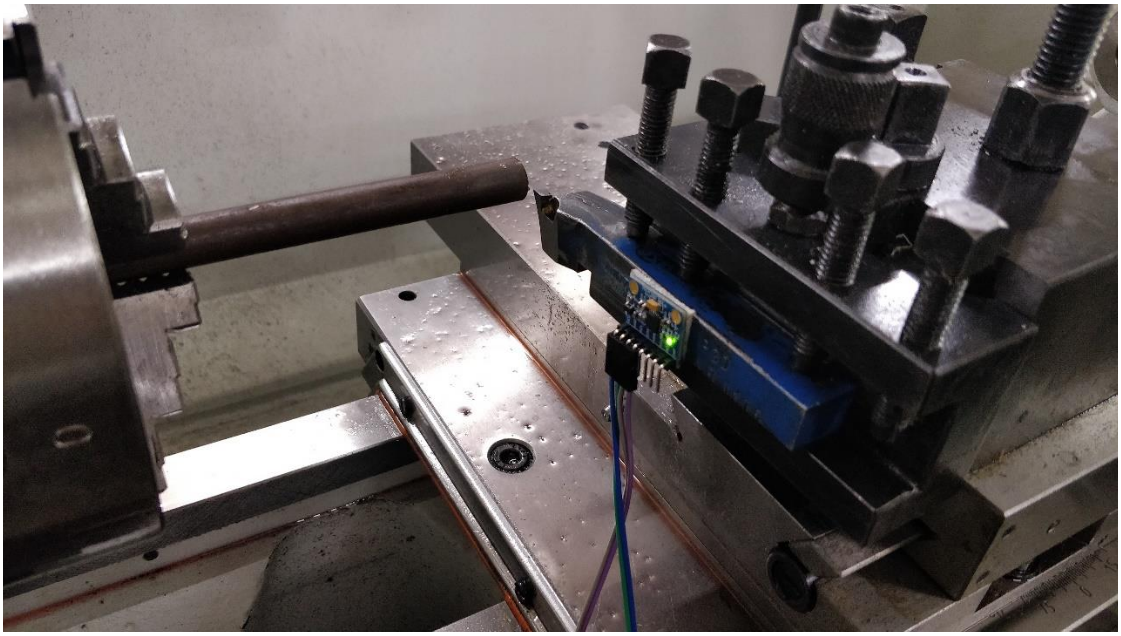

Fig 4.7 Module on Lathe

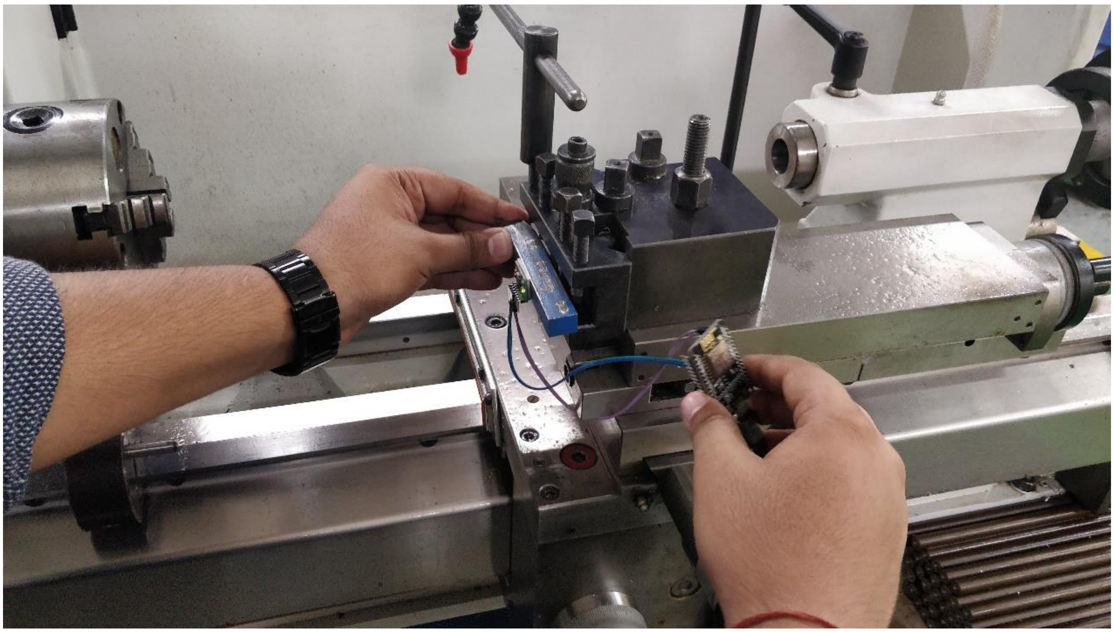

Fig. 4.8 Placement of tool on Lathe

Fig. 4.9 Machining process

4.4 EXPERIMENTAL DATASHEET

Using tool vibration monitor module, we have collected the experimental data. The experimental data has been tabulated and labelled.

TABLE 4.1 RAW DATA FROM MPU-6050

SL NO	SPINDLE SPEED	DEPTH OF CUT	X (g)	Y (g)	Z (g)	TEMPERATURE (C)
1	500	1	1.04	0.15	0.03	33.28
2	500	1	1.09	0.08	0.04	33.19
3	500	1	1.19	0.21	0.04	33.14
4	500	0.5	1.11	0.06	0.01	33.71
5	500	0.5	1.07	0.13	0.05	33.52
6	500	0.5	1.11	0.1	0.05	33.57
7	1000	1	1.17	0.11	0.07	33.61
8	1000	0.5	1.11	0.02	0.01	33.94
9	1500	1	0.98	0.04	0.06	35.03
10	1500	1	1.11	0.1	0.05	34.84
11	1500	1	1.22	0.1	0.01	34.84
12	1500	1	0.84	0.04	0.05	34.46

The above raw data is manipulated and all the acceleration values are converted into hertz. The table below gives the manipulated data in tabulated format.

TABLE 4.2 MANIPULATED DATA IN HERTZ

SL NO	SPINDLE SPEED	DEPTH OF CUT	TEMPERATURE (C)	MAX VIBRATION (HZ)	MIN VIBRATION (HZ)	LABEL
1	500	1	33.28	359.43	61.047	0
2	500	1	33.19	367.969	70.49	0
3	500	1	33.14	384.478	70.49	0
4	500	0.5	33.71	371.33	35.245	0
5	500	0.5	33.52	364.578	78.811	0
6	500	0.5	33.57	371.33	78.811	0
7	1000	1	33.61	407.594	99.698	0
8	1000	0.5	33.94	397.005	37.682	0
9	1500	1	35.03	402.91	81.4	0
10	1500	1	34.84	428.802	91.008	0
11	1500	1	34.84	449.547	40.7	0
12	1500	1	34.46	373.022	81.4	0
13	1500	1	34.46	421.005	81.4	0
14	1500	0.5	34.28	424.921	81.4	0
15	1500	0.5	34.18	424.921	81.4	0
16	1500	0.5	34.18	430.729	91.008	0
17	1500	0.5	33.99	428.802	91.008	0
18	1500	1	33.8	426.866	57.559	1

4.5 TOOL WEAR PREDICTION

In this module the components are a NodeMCU and a MPU-6050 accelerometer + gyro which is connected to laptop through a USB cable which is source of power for the NodeMCU. The vibration from the machining process is sensed by the sensor is sent to NodeMCU, all the data received is sent to MATLAB cloud platform Thingspeak. The machining parameters such as spindle speed and depth of cut are hardcoded in the code. The data that was uploaded is stored and was imported as a CSV file in the MATLAB workspace. The data that was imported contained raw acceleration data, this data was worked upon and manipulated and all the acceleration values were changed to corresponding frequency values using the formula given in following table.

TABLE 4.3 MANIPULATION FORMULAS

FORMULA FOR CONVERSION		
F = (1/2*PI)*(SQRT(G/L))		
SPEED	DEFLECTION	FORMULA
500	2 micro meter	352.45*sqrt(g)
1000	1.75 micro meter	376.82*sqrt(g)
1500	1.5 micro meter	407*sqrt(g)

Using the new manipulated data a machine learning model was trained using classification learner in MATLAB. We used two different models for the prediction of the tool wear namely: Fine KNN and SVM. The above model types were chosen due to the high accuracy The above models also gave a high cross-validation accuracy. As our prediction is based on classification we chose to go with Fine KNN. The Fine KNN model gave us the training accuracy of 97.1%. The trained model was later imported and prediction is made using the last data uploaded to the Thingspeak cloud.

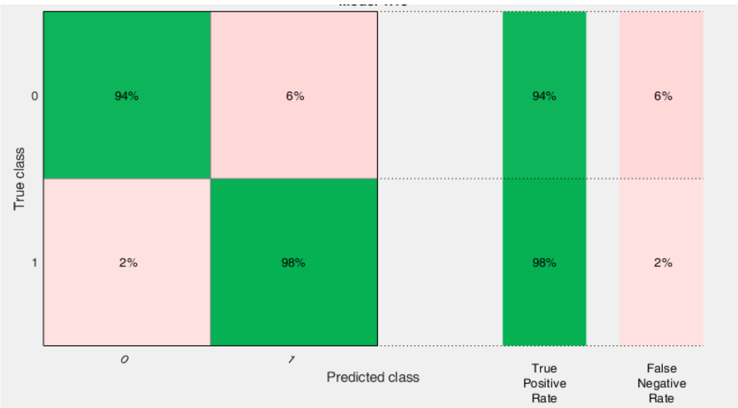

Fig. 4.10 Confusion Matrix

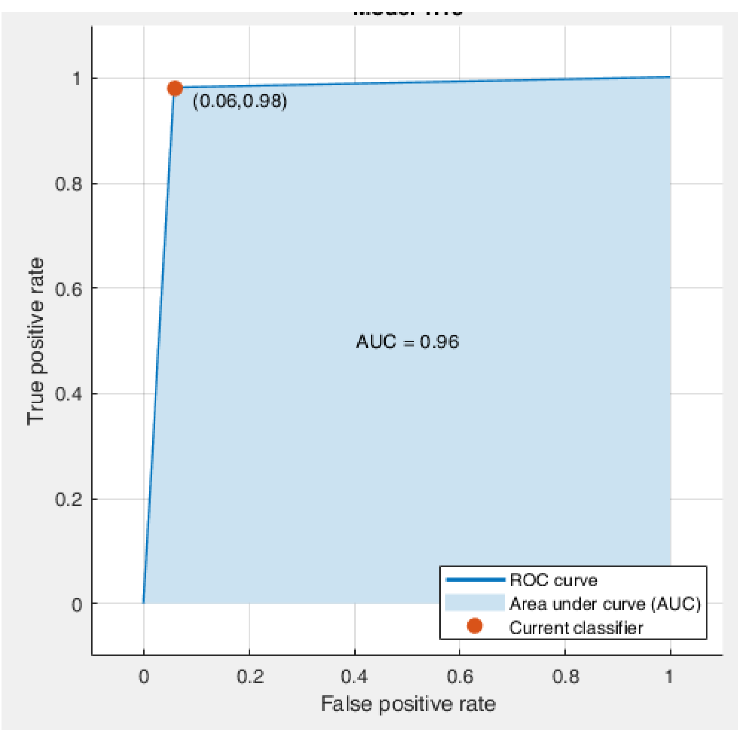

Fig 4.11 ROC

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

A module was developed for measuring vibrations and uploading the data to the cloud. A data pipeline was established between the NodeMCU and the Thingspeak cloud platform**.** A ML model was trained on MATLAB that could predict whether the tool will be wearing or not. The operation was achieved with minimal human interference.

5.2 FUTURE SCOPE

Industry 4.0 is a model that shows how industrial production follows the latest developments and changes over time. Thereby, the man, machine and the production itself constitute the force in one intelligent and independent network. In a networked factory, robots and men are becoming equal partners, having a higher degree of artificial Intelligence in relation to the previous generation of robots. The sense sensors that respond to the slightest signal are embedded into the robots, which enables the cooperation between robots and workers. IoT is going to transform our lives beyond imagination. It would make our life faster, easier and more productive. It can greatly improve productivity and our lives. And unsurprisingly, its great market potential is attracting investments from governments, telecom operators, manufacturers, and industry users. Also called M2M standing for Machine to Machine, Machine to Man, Man to Machine, or Machine to Mobile, the Internet of Things intelligently connects humans, devices, and systems.

5.3 CHALLENGES FACED

The initial aim of the project was machine health monitoring of the lathe machine, but the lathe machine has very less or no defects for over a long period of time hence we chose the prediction of tool wear.

REFERENCES

1. B.Sick, Online and indirect tool wear monitoring in turning with artificial neural networks: a review of more than a decade of research, Mechanical Systems and Signal Processing, 2002.

2. X.Li, A brief review: acoustic emission method for tool wear monitoring during turning, Machine Tools & Manufacture, 2002.

3. Ouahabi,W. Rmili, R. Serra, C. Gontier, R.Leroy Vibratory analysis for tool wear monitoring during machining (Turning, Milling), October 2004.

4. E. Kuljanic, M. Sortino, TWEM, a method based on cutting forces monitoring tool wear in face milling, Machine Tools & Manufacture, 2005.

5. V. Ts. Zoriktuev, Mechatronic Machine-Tool Systems, 2008.

6. Hongrui Cao, The concept and progress of intelligent spindles,Machine Tools & Manufacture,12 October 2016.

7. J. ARIVARASAN ,Project report on ,Optimization strategy of tool wear monitoring using vibrations ,2018

APPENDIX I

PROGRAMMING

A 1.1 NodeMCU Cloud Interface Program

```
#include <Wire.h>
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>

char ssid[] = "SSID";
char pass[] = "PASSWORD";
int keyIndex = 0;
WiFiClient  client;

unsigned long myChannelNumber = CHANNEL NUMBER;
const char * myWriteAPIKey = "WRITE API KEY";
int number = 0;
const uint8_t MPU6050SlaveAddress = 0x68;
const uint8_t scl = D1;
const uint8_t sda = D2;
const uint16_t AccelScaleFactor = 16384;
const uint16_t GyroScaleFactor = 131;
const uint8_t MPU6050_REGISTER_SMPLRT_DIV   =  0x19;
const uint8_t MPU6050_REGISTER_USER_CTRL    =  0x6A;
const uint8_t MPU6050_REGISTER_PWR_MGMT_1   =  0x6B;
const uint8_t MPU6050_REGISTER_PWR_MGMT_2   =  0x6C;
const uint8_t MPU6050_REGISTER_CONFIG       =  0x1A;
const uint8_t MPU6050_REGISTER_GYRO_CONFIG  =  0x1B;
const uint8_t MPU6050_REGISTER_ACCEL_CONFIG =  0x1C;
const uint8_t MPU6050_REGISTER_FIFO_EN      =  0x23;
const uint8_t MPU6050_REGISTER_INT_ENABLE   =  0x38;
const uint8_t MPU6050_REGISTER_ACCEL_XOUT_H =  0x3B;
const uint8_t MPU6050_REGISTER_SIGNAL_PATH_RESET  = 0x68;
int16_t AccelX, AccelY, AccelZ, Temperature, GyroX, GyroY, GyroZ;

void setup() {
  Serial.begin(115200);
  Wire.begin(sda, scl);
  MPU6050_Init();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
```

```
Serial.print(".");

 }
  Serial.println("");
  Serial.println("WiFi connected");
  ThingSpeak.begin(client);  // Initialize ThingSpeak
  }

void loop() {
  double Ax, Ay, Az, T;
  Read_RawValue(MPU6050SlaveAddress,
MPU6050_REGISTER_ACCEL_XOUT_H);
  //divide each with their sensitivity scale factor
  Ax = (double)AccelX/AccelScaleFactor;
  Ay = (double)AccelY/AccelScaleFactor;
  Az = (double)AccelZ/AccelScaleFactor;
  T = (double)Temperature/340+36.53; //temperature formula
  Serial.print("Ax: "); Serial.print(Ax);
  Serial.print(" Ay: "); Serial.print(Ay);
  Serial.print(" Az: "); Serial.print(Az);
  Serial.print(" T: "); Serial.print(T);
  ThingSpeak.setField(1,"500");
  ThingSpeak.setField(2,"1");
  ThingSpeak.setField(3,(float)Ax);
  ThingSpeak.setField(4,(float)Ay);
  ThingSpeak.setField(5,(float)Az);
  ThingSpeak.setField(6,(float)T);
  int t = ThingSpeak.writeFields(myChannelNumber,myWriteAPIKey);
  if(t==200){
    Serial.println("\n Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(t));
  }

 delay(10000);
}

void I2C_Write(uint8_t deviceAddress, uint8_t regAddress, uint8_t data){
  Wire.beginTransmission(deviceAddress);
  Wire.write(regAddress);
  Wire.write(data);
  Wire.endTransmission();
}
// read all 14 register
void Read_RawValue(uint8_t deviceAddress, uint8_t regAddress){
  Wire.beginTransmission(deviceAddress);
```

```
  Wire.write(regAddress);
  Wire.endTransmission();
  Wire.requestFrom(deviceAddress, (uint8_t)14);
  AccelX = (((int16_t)Wire.read()<<8) | Wire.read());
  AccelY = (((int16_t)Wire.read()<<8) | Wire.read());
  AccelZ = (((int16_t)Wire.read()<<8) | Wire.read());
  Temperature = (((int16_t)Wire.read()<<8) | Wire.read());
}

void MPU6050_Init(){
  delay(150);
  I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_SMPLRT_DIV, 0x07);
  I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_PWR_MGMT_1,
0x01);
  I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_PWR_MGMT_2,
0x00);
  I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_CONFIG, 0x00);
  I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_GYRO_CONFIG,
0x00);//set +/-250 degree/second full scale
  I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_ACCEL_CONFIG,
0x00);// set +/- 2g full scale
  I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_FIFO_EN, 0x00);
  I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_INT_ENABLE, 0x01);
  I2C_Write(MPU6050SlaveAddress,
MPU6050_REGISTER_SIGNAL_PATH_RESET, 0x00);
  I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_USER_CTRL, 0x00);
}
```

A 1.2 MATLAB Prediction Code

```
data = thingSpeakRead(748297,'ReadKey','UUWV7AEUZKBC6727');
depth_of_cut = data(1,2);
speed = data(1,1);
x = abs(data(1,3));
y = abs(data(1,4));
z = abs(data(1,5));
temperature = data(1,6);
max_accl = max([x,y,z]);
min_accl = min([x,y,z]);
if(speed==500)
   max_vibration = 352.45*sqrt(max_accl);
   min_vibration = 352.45*sqrt(min_accl);
elseif (speed==1000)
   max_vibration = 376.82*sqrt(max_accl);
   min_vibration = 376.82*sqrt(min_accl);
elseif (speed==1500)
   max_vibration = 407*sqrt(max_accl);
   min_vibration = 407*sqrt(min_accl);
end
t =
table(depth_of_cut,max_vibration,min_vibration,speed,temperature,'VariableNames',
{'DEPTHOFCUT','MAXVIBRATION','MINVIBRATION','SPEED','TEMPERATUR
E',});
label = trainedModel.predictFcn(t);
if(label==0)
   label="Wear";
elseif(label==1)
   label="Not Wear";
end
t_predicted =
table(depth_of_cut,max_vibration,min_vibration,speed,temperature,label,'VariableNa
mes',{'DEPTHOFCUT','MAXVIBRATION','MINVIBRATION','SPEED','TEMPER
ATURE','PREDICTED_LABEL'});
disp(t_predicted)
```