Expertise
and insight
for the future

Thanh Nguyen

Development of Predictive Maintenance Model for Induction Motor

Metropolia University of Applied Sciences

Bachelor of Engineering

Degree Programme in Electronics

Bachelor's Thesis

12 April 2021

Author Title Number of Pages Date	Thanh Nguyen Development of Predictive Maintenance Model for Induction Motor 56 pages + 2 appendices 12 April 2021
Degree	Bachelor of Engineering
Degree Programme	Degree Programme in Electronics
Professional Major	Electronics
Instructors	Anssi Ikonen, Senior Lecturer

The aim of this project was to create a model to predict the remaining useful life of induction motor with minor investment available in the plant. The project helps to broaden electronics student's horizons of automation and machine learning field. It answers whether Siemens Programmable Logical Controller (PLC) can work with other Internet of Things (IoT) tools in the Fourth Industrial revolution. This project's result can be used as one method alternative for a maintenance engineer to perform predictive maintenance without attaching additional sensor devices on a machine or buying expensive predictive software from a third party.

In this project, the development of a predictive maintenance model is started by collecting induction motor data. The induction motor data is collected by the PLC and stored on the Microsoft SQL local server. The raw data is fetched and loaded into the data frame for predicting the induction motor's remaining useful life. The remaining useful life of the induction motor is estimated by using the linear regression model. Lastly, the efficiency of a predictive model is evaluated by calculating its root mean square error.

Keywords	Predictive maintenance, Induction motor, Linear regression model, PLC

Contents

List of Abbreviations

AC	Alternating current
ASCII	American Standard Code for Information Interchange
CE	Critical event
CM	Communication module
CPU	Central processing unit
CRC	Cyclic Redundancy Check
CSV	Comma-Separated Values
DB	Data Block
DBMS	Database Management System
DC	Direct current
FBD	Function Block Diagram
HMI	Human-machine interface
HVAC	Heating, Ventilating, and Air Conditioning
I	Input
IDE	Integrated Development Environment
IIoT	Industrial Internet of Things
IoT	Internet of Things
L	“Temp” memory

LAD	Ladder logic
LAN	Local Area Network
M	Bit Memory
NC	Normally close
OEE	Overall Equipment Effectiveness
P/F	Potential failure – Functional failure
PdM	Predictive maintenance
PLC	Programmable logic controller
PLCs	Programmable Logic Controllers
PM	Preventive maintenance
PSI	Pound per square inch
PU	Physical Unit
Q	Output
RJ45	Registered Jack 45
RLO	Result of the logic operation
RMS	Root Mean Square
RMSE	Root Mean Squared Error
RPMs	Round Per Minutes
RTU	Remote terminal unit

RUL	Remaining Useful Life
SQL	Structured Query Language
STL	Statement List
TCP/IP	Transmission Control Protocol/Internet Protocol
V/F	Voltage/Frequency
VB	Visual Basic
VFD	Variable Frequency Drive

1 Introduction

In the industrial field, maintenance cost dominates the significant proportion of total manufacturing operating costs. The percentage of maintenance cost varies from 15 per cent to 60 per cent, depending on the specific industry. However, the effectiveness of maintenance achieves merely two-thirds of the result. It means that the maintenance cost is wasted one-third because of unnecessary or improperly carried out maintenance. Furthermore, ineffective maintenance management also significantly increase production loss time and degrade product quality. Therefore, suitable maintenance method and administration in the industrial area is ability-enhancing competitiveness of manufacturing quality product in the world market. [1,1.]

There are two types of traditional maintenance methods, which are reactive maintenance and preventive maintenance. Reactive maintenance can be understood simply as run-to-failure as straightforward. It means that the machines or equipment do not need to be fixed and replaced until they break. This method has dominated the industrial field since the first build of manufacturing. However, the high expenditure in spare parts inventory, overtime labour, machine downtime and the low production availability made this method gradually lost credibility. As a result, along with this method, people also performs the next level of maintenance method, which is called preventive maintenance. In this method, many preventive maintenance tasks like lubrication, machine adjustment, and machine rebuild are performed after an interval to prolong machinery's normal life. However, this scheduled method leads the plant to repair machinery and equipment; even it is unnecessary repairs or not catastrophic failures. Therefore, the material and labour used for repairing are wasted. [1,2.]

Predictive maintenance eventually is developed to eliminate those ineffective points of two former maintenance strategies. Predictive maintenance can estimate the average machine's failure time and identify particular faults based on the machine's data. Hence, this method has tackled those weaknesses of former maintenance methods. The benefits of predictive maintenance that deserve mention are the meaning of downtime for maintenance, better managing inventory, eliminating unplanned downtime, and maximising the equipment lifetime. [2,2-3.]

The project's aim was to simulate predictive maintenance with minor investment based on having equipment in the plant. The project helps to broaden the electronics student horizons of the automation field working with the Internet of Things (IoT). Therefore, it also answers whether Siemens Programmable Logical Controller (PLC) can work with other Internet of Things (IoT) tools in the Fourth Industrial revolution. To be more specific, the objective is the successful estimating the remaining useful life of induction motor, which is the equipment existing in every plant in the world. Siemens Programmable Logical Controller (PLC) CPU executes the command to run the motor, get electrical information data from the motor and storing them in the Microsoft SQL Server database for further data analysis action. Furthermore, Data query also can be performed remotely through a website with the same network. Ultimately, this project's result can be used as one method alternative for a maintenance engineer to perform predictive maintenance without attaching additional sensor devices on a machine or buying expensive predictive software from a third party.

The thesis report includes the detailed definition of each maintenance method, the relation of predictive maintenance with condition monitoring, the procedure for developing predictive maintenance, the explanation of programming code, the project's result, and the conclusion.

2 Predictive Maintenance Background

2.1 Reactive and Preventive Maintenance

In the Introduction chapter, reactive and preventive maintenance is briefly introduced. Then, this chapter explains the difference in roles, benefits and disadvantages of reactive and preventive maintenance in the industrial fields.

2.1.1 Reactive Maintenance

The purpose of reactive maintenance is to restore equipment to its normal operating condition when the breakdown occurs. The initial cost of reactive maintenance is one of the benefits that deserves mentioning. The plant does not need to spend time and money on building the maintenance planning system. Therefore, the number of needed staff for this strategy is also smaller than other maintenance methods. Those expenditures can instead be used for other investments in the plant. In the industrial area, unpredictable faults always happen despite having well-planning maintenance. Therefore, reactive maintenance still has a place in a well-rounded maintenance strategy. However, the industry rule of thumb states that reactive maintenance should occupy only 20 per cent of maintenance time in the field. The severe risk and disadvantage of reactive maintenance is the result of this statement. [3.]

Despite having a lower initial cost, reactive maintenance has a lot of serious disadvantages. Without planning maintenance, an industrial plant must face numerous unexpected downtimes during production time. Those unexpected downtimes lead the plant to late orders, damaged reputation, unavailability of emergency maintenance labour and spare parts. Consequently, the expenditure on premium fast shipping and overtime labour has gone far greater than the low initial cost. Not only causing higher cost but unexpected critical damage also causes unsafety or can be fatal to the operator on the production line. Those factors prove that reactive maintenance is only a short-sighted approach or proper for a particular process. Reactive maintenance conjunction with preventive or predictive maintenance, is a critical approach in the industrial area to achieve the best result. [3;4.]

2.1.2 Preventive Maintenance

Unlike reactive maintenance, preventive maintenance is the planned maintenance that can prolong the lifespan of the equipment. This strategy aims to maximise the average useful life of any equipment by adjusting, cleaning, lubricating, repairing, and replacing the schedule before a breakdown. Hence, the plant has to identify all equipment in the machine and its specific life cycle. Checking and needed replacement action need to be done before the degraded time of the machine. [5.]

Figure 1 below illustrates the operation cycle of preventive maintenance action.

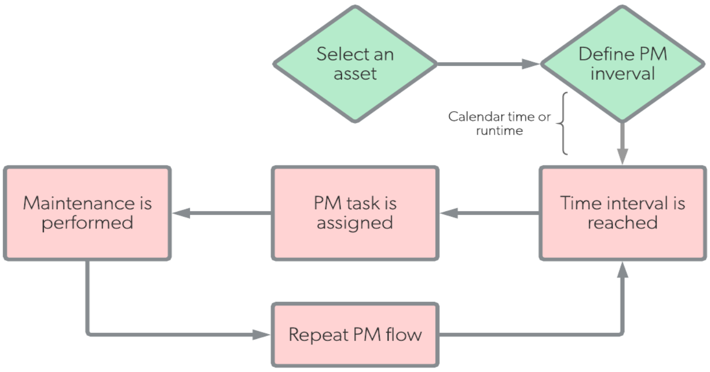

Figure 1. Preventive maintenance cycle Reprint from Why is preventive maintenance important for organisations? [6].

One equipment's particular life cycle can be determined by measuring its data with specific tools, the manufacturer's threshold, and the average working hours of its equipment. However, setting the machine's proper maintenance schedule becomes complicated if it has an enormous asset in the plant. [5.]

The benefits of preventive maintenance are the enhancement of safety, longer equipment lifespan, increased productivity, reduced costs and lower energy consumption. When technicians regularly inspect the machine, the defects on the machine can be spotted. Therefore, the plants can reduce the unexpected downtimes and risks for labours in the fields. The fewer unexpected downtimes happen, the more output products are produced for the plant. Apart from that, the cost of repairing equipment to the point of failure may take ten times more than scheduled maintenance action. [5.]

Therefore, applying preventive maintenance also saves an average of 12 to 18 per cent cost for the company. Undoubtedly, good maintaining assets can minimise electrical and mechanical stress for equipment. Therefore, the energy consumed can be reduced due to fewer unexpected faults. However, the remaining disadvantages of preventive maintenance are the barrier for many plants to approach this method. [5.]

The approach of this strategy requires enormous times and the capacity of the plant to achieve. Utilising this periodic maintenance requires regular monitoring and inspecting the equipment to get the proper maintenance schedule. It also costs much time for employees to statistic data and makes maintenance checklists of needed assets. Therefore, predictive maintenance is an advanced form of preventive maintenance that can tackle some cons and ease the needed planned industrial labours. [5.]

2.2 Predictive Maintenance

2.2.1 Relationship of Condition Monitoring and Predictive Maintenance

As described in previous chapter 2.1, predictive maintenance is an advanced form of preventive maintenance in which condition monitoring is vital for both strategies' success. Therefore, the relation between predictive maintenance and condition monitoring needs to be explored for further understanding.

In a sense, the term "condition monitoring" has been in plant early, ever since the outset of machine presence. People began to perfect the process of catching machine degradation before it turns to failure or fatal. Everything evolved when people moved into the 20th century; enterprising engineers developed more sophisticated ways to measure equipment health. Condition monitoring was based merely on analysis techniques, which are vibration analysis in 1938, oil analysis in 1946, infrared thermography in the 1950s, acoustic emission analysis in 1970, and motor current signature analysis in the 1970s. Before the computers' adoption was widespread in the 1980s, those techniques above require plant engineer to be tough in manual collecting data and expertise in analysing and interpreting data. However, most plants still use manual condition monitoring system in their field. Apart from that, many developed industries have yet to complete move from offline to online system. When the industry is stepping into the 21st century, those former analytics techniques merge with the cutting-edge technology of deep learning and the Industrial Internet of Things (IIots). Therefore, maintenance engineer and data analyst can apply automated, expert updated analysis to a vast amount of continuous data collected by wireless connection and storing those data in a secured database or/and on a cloud. It can be seen clearly that the Internet of Things and the advancement of storage helped the industrial field performing more favourably the process of the planned task. [7.]

To be more specific, the reason why condition monitoring is a key in predictive maintenance can be explained through the potential failure – functional failure (P/F) curve. This curve is a performance description of a component in the machine (See Figure 2). In the

P/F curve, the majority proportion of usage time is the stable performance of the component. It is the straight line and parallels the time axis (x-axis). The Critical Event(CE) represents something that happens on the component that makes the curve start to deteriorate gradually or dramatically. The deteriorating level of the curve indicates the critical component failures that shorten its life cycle. Since the CE point, the loss of function increases, and the component's problem becomes more apparent to detect. [8, 133.]

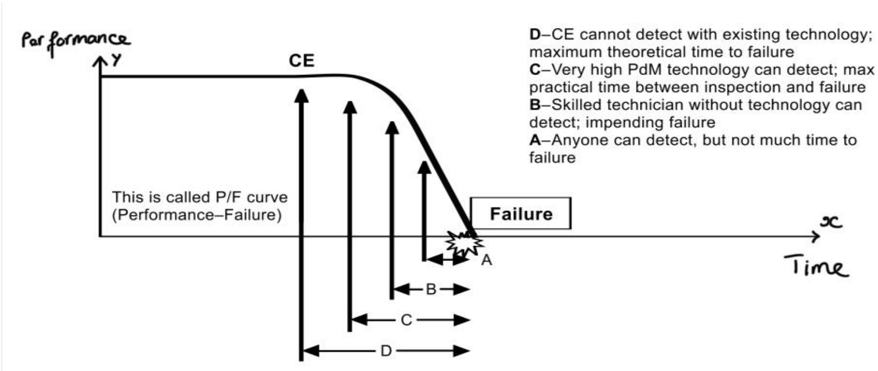

Figure 2. Performance/Failure Curve: What do Inspection and Predictive maintenance(PdM) buy you? Modified from Joel Levitt (2011) [8,134].

With condition monitoring through the P/F curve and predictive formulas, plants can estimate the degraded components' remaining useful time. The predictive formulas are the factors that go beyond condition-based care. The collected data is analysed using predictive algorithms. Next, the result can determine the trends and detect the repairing time of the asset. Therefore, the maintenance team can have time to repair spare parts, schedule downtime and training time. These condition monitoring and predictive formulas are the deciding values that make predictive maintenance unique compared to other strategies. [11.]

Remarkably, each P/F curves is the progression of one component's failure mode from its inception through the increasing loss of function to breakdown. There are many components in the machine, and one component can have several failure modes. As a result, there are many monitoring graphs of one machine, and each graph has multiple P/F curves representing several failure modes of that component. [8,134.]

The condition monitoring graph and predictive formulas prove their benefits by aggregating data and showing the component's supposed interval life before the failure occurs. This combination brings many benefits to predictive maintenance, competing with other traditional maintenance methods, illustrated explicitly in the next chapter 2.2.2. Eventually, the plant can have the maintenance decision based on reliable source instead of frequently replacing parts after duration.

2.2.2 The Importance/Benefits of Predictive Maintenance

In predictive maintenance, the equipment is inspected through monitoring and replaced based on some specific conditions. For example, a filter needs to be changed monthly according to regular preventive maintenance; however, predictive maintenance can help plants decide to change the filter based on differential pressure monitoring. If the pressure exceeds a specific value of PSI, the filter need to be changed. From the explanation above, predictive maintenance can ease to maximise the used time of equipment and minimise the extravagant replacing parts in the plant. The case is true that preventive maintenance becomes worthless when the detection of faults to failure is short. The equipment goes quickly to failure from profitable operation through deterioration to failure in one hour make it possible for preventive maintenance to schedule the plan. However, full-time monitoring can cycle at a minute to second speeds to detect equipment's fast deterioration. Predictive maintenance tools are permanently mounted or controlled by a Programmable Logic Controller (PLC) and computer controller, then eventually by a standard desktop computer. If the data exceed a threshold value, the alarm or notification shows up, and controllers safely shut down the sequence. It is an advantage of predictive maintenance that can spot in real-time any extraordinary on the working machine. Those unnatural data also can serve for further analysis execution. Another advantage of predictive maintenance is that condition monitoring is non-interruptive, which means PM inspections are going on while the machine is making money. [8, 148.]

According to Mobley [9], the benefits of applying predictive maintenance are apparent in cost deduction. The maintenance cost is decreased by 50 per cent, and spare parts inventory is minimised by 30 per cent compared to before use. Predictive maintenance also saves a significant amount of time for the company, which is determined as money. The case is true that production time increase due to a 50 per cent unexpected failure reduction. Moreover, the maintenance and overhaul time also decline by 60 per cent. Besides bringing benefits in terms of time and money, predictive maintenance also increases the plant's safety and reputation in the industrial field. [9.]

Unlike other maintenance methods, predictive maintenance can early spot machine problem, which reduces the risk of catastrophic failures, can lead to personal injury or death. This benefit further can reduce several insurances for companies. The availability of the machine on the production line means more qualified products are produced punctually. This consequence satisfies customers and improves brand reputation in manufacturing. In the future, many old and new customers find the plant for competency in manufacturing. Ultimately, predictive maintenance can make the plant for getting higher margins in production and revenue. Those are all possible benefits of predictive maintenance. The following sub-chapter 2.2.3 is the approach of the predictive maintenance process, especially for the thesis topic. [10.]

2.2.3 The Approach of Predictive Maintenance

Based on the previous description, predictive maintenance(PdM) is shaped by three main elements: condition monitoring to assess asset health in real-time, predictive formulas to diagnostic equipment data and Internet of Things(IoT). In general, PdM is a tool for collecting and analysing asset data. These data allow the maintenance team can further to identify the needed maintained area on the machine.

There are five main steps to achieve predictive maintenance presented in Figure 3 below. The first step is data acquiring which is one of the most crucial steps in PdM. The reason is that the more reliable data capture, the more precise result achieves. Each piece of equipment in predictive maintenance is monitored by sensors mounted on the machine or communicating devices for transmitting all data to the main controller. In this project, the variable frequency drive can transmit related data on the induction motor to the programmable logic controller through Modbus RTU. Then, data is stored in the Microsoft SQL database. The detail of this operation and specification of devices can be found in other chapters below. In figure 3, the second step of predictive maintenance is the pre-processing data. This step aims to remove all noise, outlier, missing value to form valuable condition indicators for the next step. The next step in performing predictive maintenance is condition indicators identification. Condition indicators are features whose behaviour changes predictably as the system degrades. These features are used to discriminate between healthy and faulty operation. To get the valuable condition indicators, maintenance engineers need to have good data analysis skills and use those techniques in figure 4 below to extract. In this project, the data analysing step is performed in Python language to get useful condition indicators. These useful condition indicators are a vital contribution to the next train model step. The train model step manipulates classification, regression, and time-series modelling to predict the time-to-failure of equipment. To be more specific, this step can track changes in the machines to determine anomalies. It also identifies which parts of the machine require attention. Moreover, the relationship between the extracted features and the degradation path of assets can estimate the time to failure of equipment with the proper train model. The pandas and seaborn package in Python can visualise the life intervals of parts in the induction motor for this thesis project. Ultimately, the last step of predictive maintenance is the deployment and integration step. After successful testing the algorithm, the algorithm can be run by deploying it on the cloud, edge device or both. The pre-processing and feature extraction steps can be performed on an embedded device and then send the extracted features to the cloud. [2.]

Those steps in figure 3 are the whole process for accomplishing the predictive maintenance in the plant. However, this thesis project is limited to the train model step due to the beginning's simulation objective. The following sections below are the details and functions of needed devices for the project.

Figure 3. Predictive maintenance workflow reprinted from Mathworks (2019) [2].

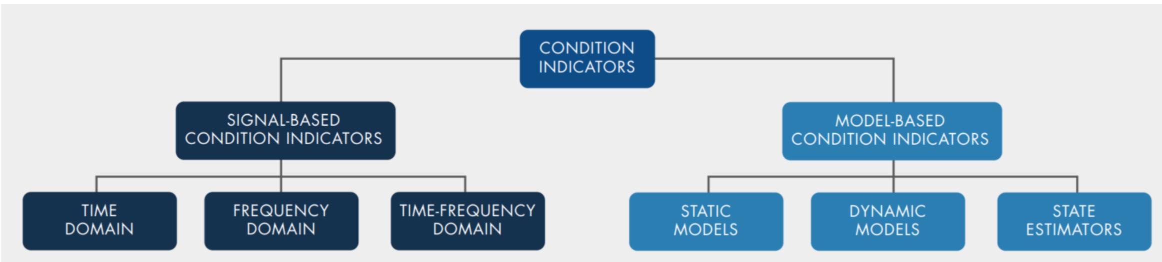

Figure 4. Define condition indicators using signal-based and model-based method reprinted from Mathworks (2019) [2].

3 Predictive Maintenance on Induction Motor

3.1 Induction Motor

3.1.1 Structure and Operation of Induction Motor

With the popularity of induction motor in various applications, the induction motor is shown as one of humankind's most essential inventions. In history, the first practical alternating current(AC) motor was constructed in the 1880s, then a wide range of AC motors is available nowadays. Unlike DC motors applying for electronic remote-control and hobbyist devices, most household or industrial appliances rely much on AC motors. The reason explaining for this is high-voltage AC power which carries great energy through distances by power lines, with minimum power loss. With its benefit in high efficiency, the electrical power in houses and other buildings is AC. Therefore, electric fans, blenders, or pumps with AC motors can use AC from the electrical outlets. AC motors are diverse with many types and classified in several ways. However, the AC motors can be classified into four main types in general based on their power supply and speed. The AC motors are polyphase or single-phase depend on the electrical content of the motor's incoming power. Similarly, the AC motors are classified as synchronous or asynchronous when their character has the relationship between the motor's speed and incoming power frequency. In this section, the structure and the operation of the AC asynchronous motor are explained. [12, 89.]

Like any other AC motors, the structure of an asynchronous motor has two parts rotor and stator for making the shaft turn in the machine. The rotor is the rotate part when the motor is powered, and the stator is the stationary part of the motor. The stator of asynchronous motors has the same structure as other types of motors. It contains windings, which are electromagnets, produces a rotating magnetic field. This rotating field of the stator causes the rotor to turn. On the other hand, the rotor in the asynchronous motor has a different structure from other AC motor types. Unlike the permanent magnet rotor of synchronous motors, the asynchronous motor rotor is a squirrel cage with many conductors embedded into its surface in a striped pattern. The name squirrel-cage is based on its shape, which is the same as a running wheel used by gerbils and hamsters. Figure 5 below shows the idea of a squirrel-cage rotor from the horizontal view section. [12, 97-98.]

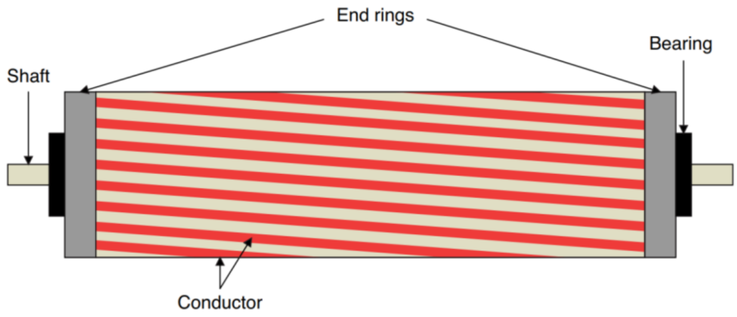

Figure 5. Squirrel-cage rotor Reprint from Motors for Makers – A Guide to Steppers, Servos, and Other Electrical Machines [12, 98].

Constituent conductors along the rotor's surface are usually made of copper or aluminium, and the core is commonly composed of steel. As shown in Figure 5, each end of conductors connect with the metal end rings and is skewed at an angle to reduce cogging when the rotor temporarily locks in place as it turns. Thanks to this arrangement of conductors, the rotor can operate normally and smoothly. However, making rotor turn relies significantly on the stator structure and the interaction between rotor and stator.

The stator of the induction motor is located outside of the rotors, and its winding receives AC power. This input AC power can be a single-phase or a three-phase power supply that depicts the sinusoidal provided power in single-phase and three-phase. Due to the power characteristic, stators' windings are grouped into sets called phases. Therefore, the induction motor can be called a single-phase asynchronous motor or a polyphase asynchronous motor. The fact is that the stator has one phase for each phase of the input power, and the windings in the same phase receive power from the same phase. Figure 6 below shows the structure of a stator of a three-phase AC motor. The induction motor' stator has the winding A connects with winding A', B connects with B', and C connects to C'. The number of windings per phase is the number of stator poles, and they are always even number. Hence, the stator is called two poles if the asynchronous motor has six windings and use a three-phase power supply. [12, 91.]

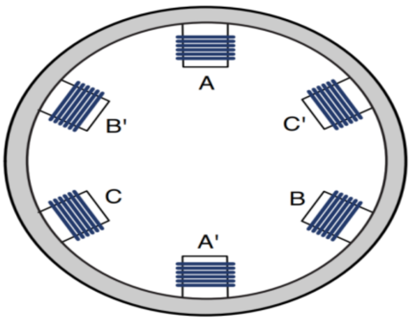

Figure 6. The stator of a three-phase AC motor with two poles reprint from Motors for Makers – A Guide to Steppers, Servos, and Other Electrical Machines [12, 92.]

In general, the changing rate of the stator electromagnetic field is the motor's synchronous speed. The rate of this speed depends on the frequency of input power and can be calculated by the following formula:

$$n_s = \frac{120}{p} f \qquad (1)$$

According to the formula, if the AC motor has p poles, the synchronous speed in round per minute is n_s. The more poles existing in the motor, the slower synchronous speed is because of more windings to pass in the electromagnetic field. The table in figure 7 illustrates the corresponding synchronous speed with the number of poles in the motor.
The operation of the asynchronous motor is based on the principle of electromagnetic induction. If the conductor above is brought into the stator's rotating field, the conductor forms the different voltage across its surface. This occurrence is referred to as electromagnetic induction, and the voltage is called induced voltage. Afterwards, induced voltage produces a current in the conductor as a red arrow illustrated in figure 8. When a current-carrying conductor is placed inside a rotating magnetic field, it receives a force that causes the conductor to move in the same direction as the stator's magnetic field. Hence, the rotor speed of the motor turns at speed slower than the synchronous speed. As a result, if a motor is asynchronous, the shaft's speed is always less than the synchronous motor speed until it reaches the full-load torque. Because of that, the term asynchronous motor is referred to according to this fundamental principle operation. Moreover, the asynchronous motor is also commonly called an induction motor based on the inducing phenomenon between stator and rotor.

No. of poles	Sync. speed (at 60 Hz)	Sync. speed (at 50 Hz)
2	3600	3000
4	1800	1500
6	1200	1000
8	900	750
10	720	600
12	600	500

Figure 7. Motor synchronous speeds and number of poles Reprint from Motor & Drives – A Practical Technology Guide [14, 96]

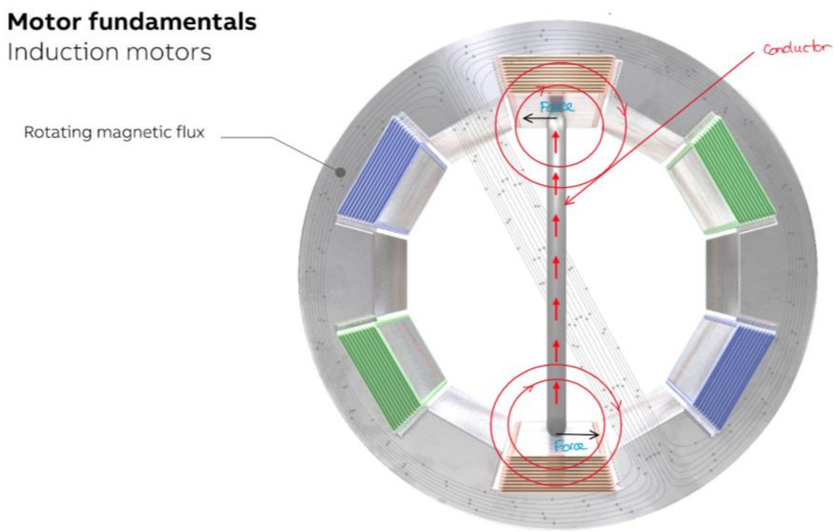

Figure 8. Conductor in the electromagnetic field of stator Modified from ABB Electric motor fundamentals Free online training program (2020) [15.]

3.1.2 Faults Diagnose on Induction Motor and Its Benefits.

One of the most popular malfunctions on induction motor is the eccentricity related to other faults such as broken rotor, bar faults, bearing fault and stator faults. The eccentricity of the induction motor dominates more than 90% of overall induction motor failures. Indeed, most induction motor in industrial facilities run under nonidea conditions which cause more than 40% bearing faults. Bearing faults can be classified as the outer raceway, inner raceway, ball defect and cage defect. Over time, these defects become significant and cause acoustic noise on the machine. These defects cause not only noise but also the primary vibration on the induction motor. Therefore, mechanical vibration, infrared or thermal and acoustic analyses are commonly used predictive maintenance methods to monitor the bearing health. Apart from bearing faults, stator faults withstand approximately 30% to 40% of all-electric motor failures. The cause of these faults mostly come from electric and mechanical stress on the stator windings. In a symmetrical induction motor, a stator winding turn fault can produce a large amount of current flowing through turns, meaning much current requirement from the source. The considerable current amount creates much heat for the insulation used in the stator winding. The insulation degrades and becomes damaged when its threshold reaches and cause failure on the induction motor. The temperature monitoring on the stator winding seems to be the best solution to detect the induction motor's insulation degradation. Besides, broken rotor bar faults also contribute to 5% of induction motor faults. The broken rotor bar fault consequence comes from thermal, environmental, mechanical and residual stress. Lastly, the eccentricity fault is the last fault type on the induction motor. The air-gap eccentricity is a typical defect finding in the motor rotor. Indeed, the centre of the rotor and stator on healthy motors is aligned perfectly. If the rotor is not aligned centrally, the developing in radial forces or magnetic pull cause the rotor-stator rub, which results in damage on both rotor and stator. The eccentricity is likely to diagnose with the harmonic line current signal. By monitoring the time variation of a full-load induction motor, the eccentricity fault can be classified.

It is apparent that faults diagnosis helps separate happening faults on induction and has time to plan downtime for maintenance. By defining each fault types, maintenance engineers can easily detect fault base on the failure mode through monitoring the induction motor's features. The maintenance time on the induction motor also is improved with predefined faults. Technician and maintenance engineer do not cost much time for investigating faults or identifying fault equipment. The revised spare part on the induction motor can have time to be prepared and directly changed the field. The low downtime, therefore, can enhance the productivity on the production line.

3.2 Siemens Programmable Logic Controller S1212

3.2.1 Programmable Logic Controller S1212 Function and Characteristic

In Predictive maintenance, the essential equipment is the controller for data controlling and executing all output tasks in the system. Thanks to the stability and reliability, the programmable logic controller (PLC) is chosen to monitor and control the system. In industry's environment, Siemens' PLCs are observed highly in this fiercely competitive environment. The model central processing unit (CPU) 1212C DC/DC/DC is one of the SIMATIC S7-1200 compact controller class of Siemens. Based on the model's name of the CPU, the configuration of hardware can be identified. The serial number on the CPU indicates that the CPU belongs to SIMATIC S7-1200 programmable logic controller class. This CPU is powered by 24V DC and has eight digital input 24V DC, six digital output 24V DC onboard. The serial number of the CPU indicates the CPU class, and the letter C represents that this is the compact CPU. The three last categories in the serial name tell the power supply, type of input, and output on the CPU. The short hardware description and features can be seen in figure 9 below when choosing the CPU on the TIA Portal software.

Figure 9. Description of CPU S1212 on TIA Portal which is the screenshot captured from TIA Portal software.

This class of Siemens PLC has many advantages that become essential equipment in the industrial environment. According to Siemens, the S7 – 1200 CPU supports monitoring and controlling a wide range of devices along with auxiliary automation protocols. The compact design, low cost, and powerful features make the S7-1200 compact controller fit in controlling a small automation system. The S7 – 1200 CPU is determined as a powerful controller with the microprocessor build in, an integrated power supply, input and output circuits, high-speed motion control input/output, and onboard ANALOG inputs in a compact housing. Hence, after uploading the logic programs into the CPU through TIA portal software, the CPU can monitor and execute the system's command. It can have the ability to monitor the inputs and changes the output command base on the logic of the user program. The logic program language on TIA Portal, which programmer can use, are boolean logic, ladder logic, counting, timing, math operations, and other intelligent devices. Thanks to the PROFINET port and the ability expansion with additional modules, the CPU can have many communication resources such as PROFINET, PROFIBUS, MODBUS, GPRS, RS485, RS232, IEC networks. With MODBUS networks, the PLC CPU can monitor the changes of current, voltages, power, and direction of VFD powering the induction motors. Through MODBUS, CPU can also send the digital signal to VFD and control the speed, direction of induction motors. Ultimately, this is an ideal space-saving controller for the small project of thesis and simulation work. [16, 18.]

This CPU is chosen to undertake this fundamental responsibility. Figure 10 below shows its duty in Predictive Maintenance is to collect all input information from the VFD powering the induction motor. The collected information, including current, voltage, direction, and power, is used for further data analysing.

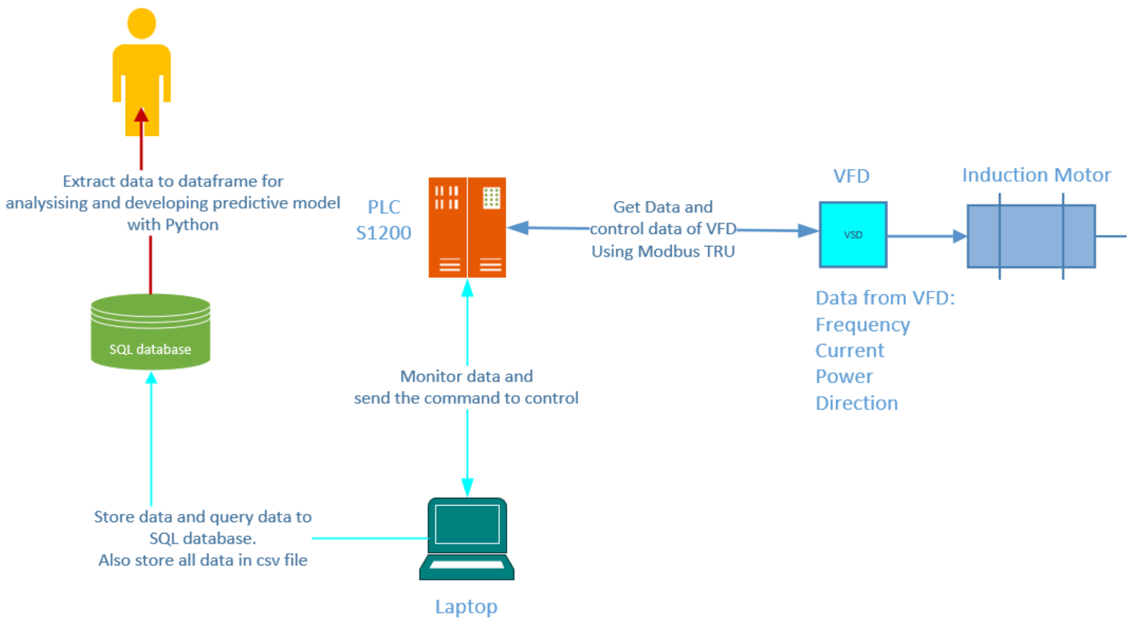

Figure 10. Induction motor data is collected and stored in the SQL database.

3.2.2 Configuration on Programmable Logic Controller CPU

Configuration of the needed programmable logic controller (PLC) is a vital step in building a Predictive maintenance system. The chosen programmable logic controller needs to ensure the ability to handle all tasks and appropriate with available needed hardware in the system. To be more specific, the chosen PLC can tackle all present tasks and development modification tasks in the future. For example, the number of digital or analogue inputs and outputs should be taken into this preparation step. Additional modules for communicating with web server and database storing ability also need to be considered in Predictive Maintenance. A wrong calculation can lead to the consequence of changing PLC along with any compatible hardware. Moreover, programs' reconfiguration also costs much time to be finished and means causing an enormous money loss on the production line. These expenses must be avoided in the industrial environment to preserve inventory and achieve the highest productivity. After identifying the appropriate PLC, the TIA Portal software's configuration is the next step. According to the Siemens manual, the STEP 7 TIA Portal offer users a friendly environment to build control logic programs, configure HMI visualisation and set up network communication. Hence, programmers can use the latest software to intuitively code from small to a large project with everything else in between. The section below is the instruction that shows how to start with Siemens TIA Portal from setting up the using PLC to adding some functionality to a program. There are two different views of the project to help users increase their productivity. The first view is called a task-oriented set of portals, or Portal view arranged based on the tools' functionality. The second view is called the Project view, a project-oriented view of the project's elements. Most PLC programmers often use the Portal view to create and chose the location of the project. The project view is the place for integrating equipment, additional module, creating logic programs and simulating their projects. Those two figures, 11 and 12 below, show two of these views as well as the explication. [16, 18.]

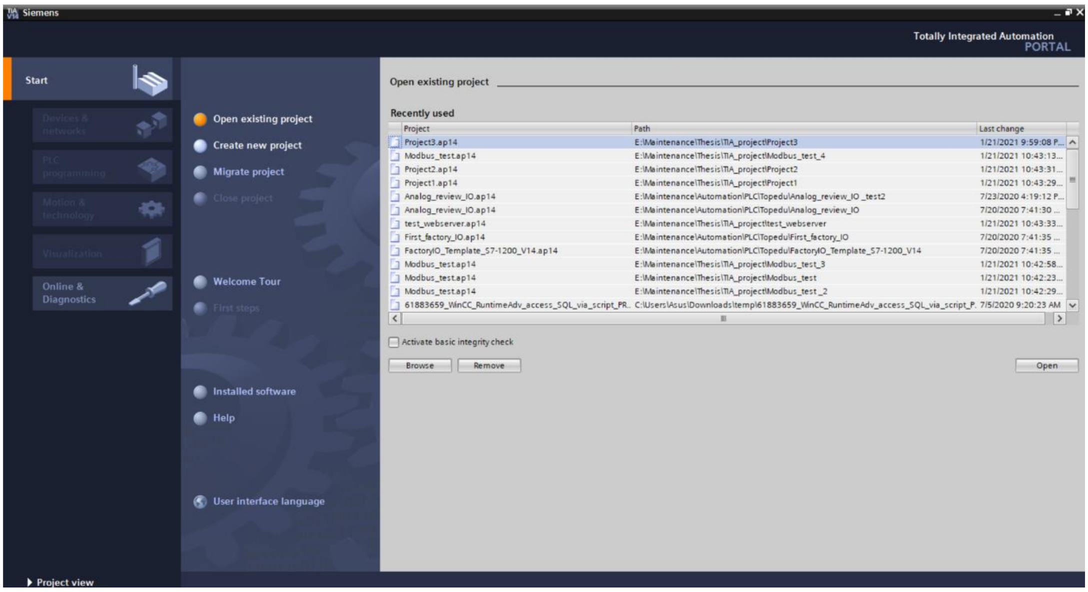

Figure 11. The Portal view interface Captured from TIA Portal software.

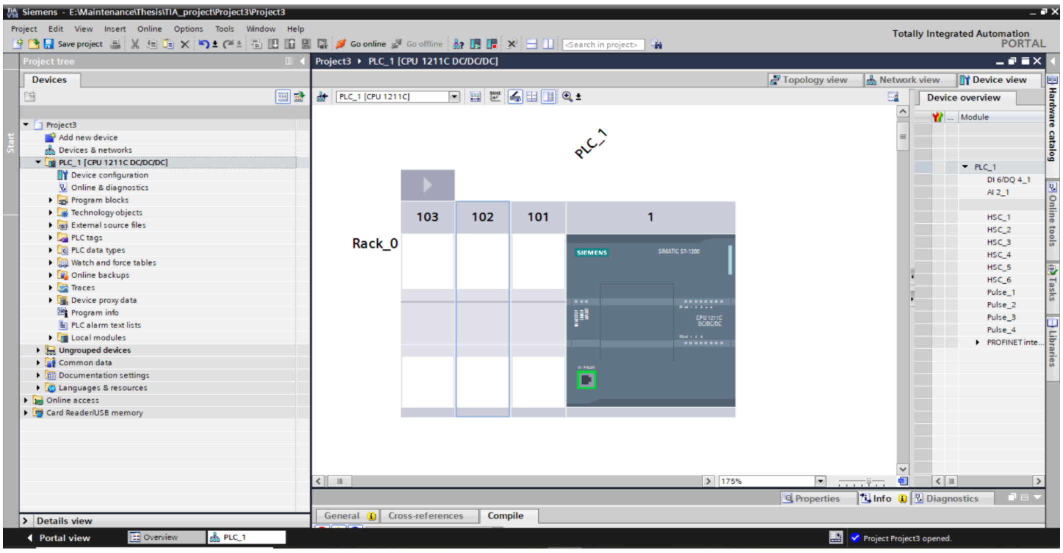

Figure 12. The Project view interface Captured from TIA Portal software.

To start the configuration, programmers need to set up TIA Portal by opening up the TIA Portal and creating a new project. Once it opens, users can click “Create new project”, type the project’s name of user’s choice in space “Project name”, then click “Create” in the Portal view. After creating the project, programmers now can add a compatible PLC to the project. However, programmers need to create a connection protocol between the laptop/computer and the PLC to add virtual PLC and additional modules to the project. This connection helps programmers’ computers communicate to the PLC and later can download logical programs to it. The connection between Siemens PLC and the programmer’s computer is Transmission Control Protocol/Internet Protocol (TCP/IP). This protocol connection needs an ethernet cable RJ45 and connects this cable from the computer to the PLC. After connecting the PLC and computer through an ethernet cable, programmers need to change Internet protocol from automatic to static mode on the computer. To be more specific, programmers click on “Properties” of Local Area Network (LAN) from the computer and switch TCP/IP to static form. In this form, the “Subnet mask” on both PLC and computer should have the same value. However, PLC and computer’s IP address need to be different from each other for communication purpose. An example of IP address structure and subnet mask for connecting the computer and PLC can be found in figure 13 below.

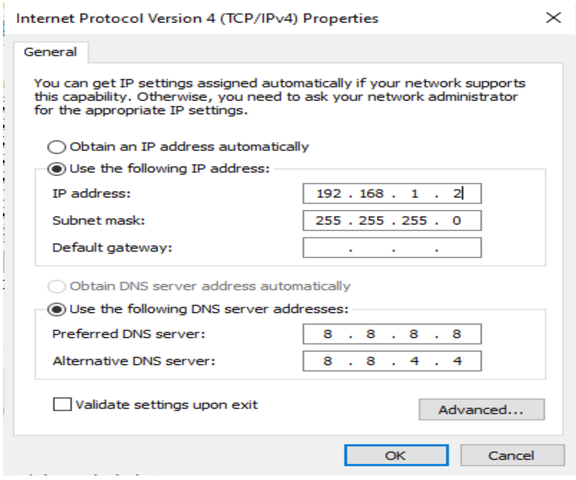

Figure 13. An example of a static IP address and Subnet mask is used to set up the computer's connection with the PLC.

When the connection protocol between PLC and computer is established, the next step of the configuration procedure is adding a virtual PLC to the TIA Portal project. There are two scenarios at this step. Suppose the PLC model has been identified beforehand by programmers. In that case, the new PLC can be directly added by choosing “Open the project view” and double click on “Add new device” to add a compatible PLC to the project, as illustrated in figure 14. On the other hand, the second scenario is that programmer cannot determine PLC's model name. In this case, TIA Portal can detect the PLC model through the command “Add new device” and chose “Unspecified CPU 1200” as in figure 15 if the PLC belong to the 1200 series. One table then comes up, and programmers press “detect” for TIA Portal automatically detect the available PLC series. In this Predictive maintenance, the chosen PLC model name is PLC S1212C DC/DC/DC, which has been clarified from the previous chapter. Ultimately, creating a project and adding a compatible PLC to the project are the first configuration steps on the PLC 1200 series.

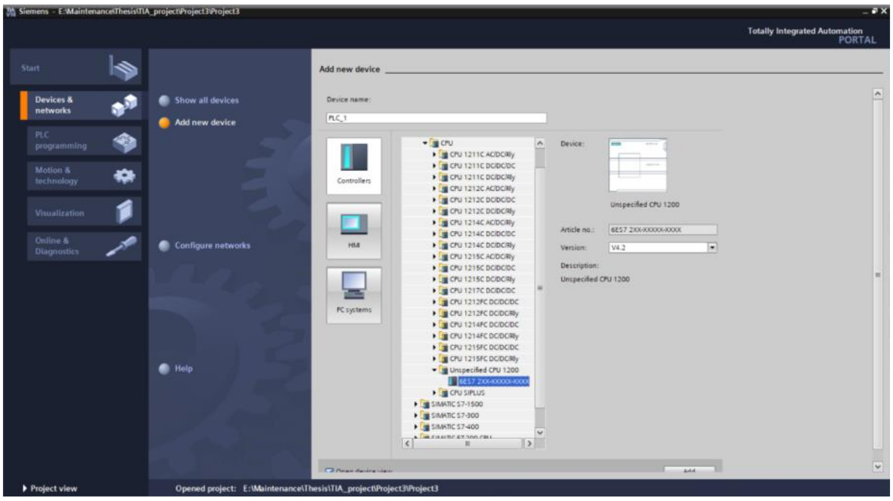

Figure 14. Unspecified CPU 1200 is used for detecting available PLC Captured from TIA Portal software.

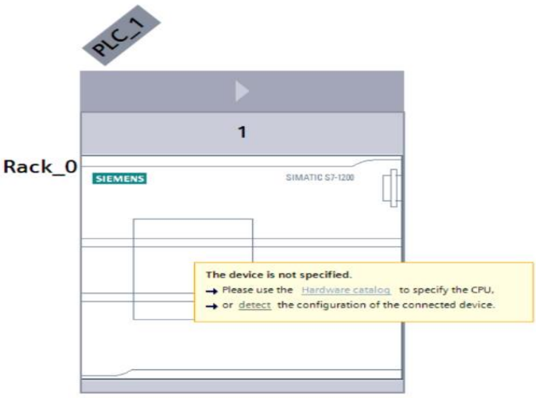

Figure 15. A part of the project view displays an unspecified PLC model S1200 for detecting available PLC Captured from TIA Portal.

3.2.3 Programming on TIA Portal V14SP1

After the configuration on the compatible hardware through SIMATIC, programmers can perform logical programming on the main block-OB1. By default, only two networks are available in the main block-OB1; however, programmers can add the number of networks to the OB1 based on their algorithm logic. With TIA Portal, programmers can program the logic of the system mainly in three programming languages such as Ladder logic (LAD), Statement List (STL) and Function Block Diagram (FBD). New programmers first-time access TIA Portal often with Ladder logic because it is explicit and straightforward. Ladder logic (LAD) is the programming language that displays the graphical symbol for representing the function used in hard-wire control diagrams. The following programming language in TIA Portal is Statement List (STL). It is a list of many instructions which allow programmers to create a program by entering the mnemonic commands. Because it is specific, STL can create a program that LAD and FBD cannot do. The remaining language that programmers can use in TIA Portal is the Function Block Diagram (FBD). It displays the program in many blocks connecting, taking multiple inputs, sending them through various instruction block, and changes specified outputs. The LAD language programming structure is shown in figure 16 below. In this project's scope, Ladder logic (LAD) is used as the primary language program for creating commands and connections of the system.

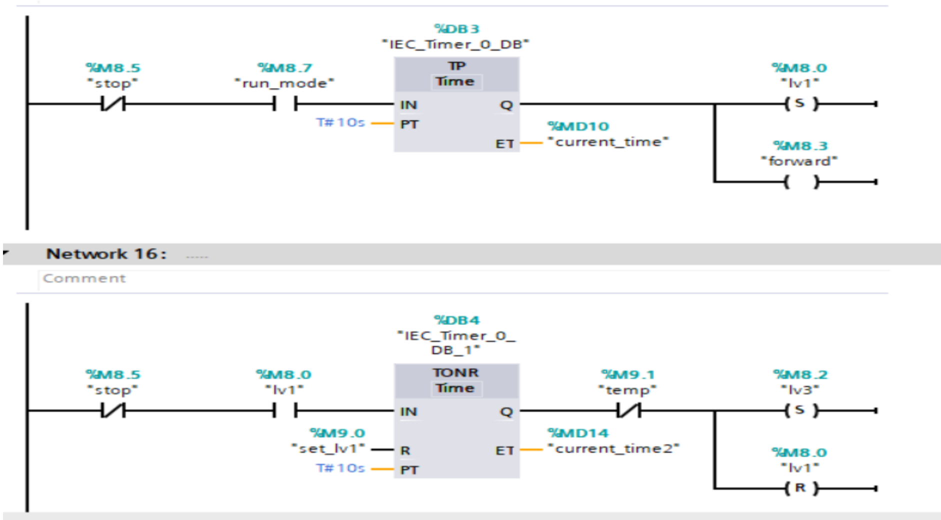

Figure 16. A part of Ladder logic (LAD) controls the speed and direction of the Induction Motor.

The LAD language is a conversion of an electrical circuit diagram rotating at 90 degrees. As in figure 16, the power rail is located on the left-hand side, and the right side is the grounding rail. All switching elements of the circuit are in the middle between two of these lines. The PLC now handles a machine's logic, including time relays or flip-flops wiring with switches, auxiliary contactors, or control contactors. On the other hand, the PLC receives the signal from other input elements (such as inputs switches or sensors) to close or open the contact handled by the PLC. Hence, the power contactors on the output side (such as motor contactors, polarity reversers, or valves) can receive the power to execute. However, the electrical diagram symbol differs from a programming symbol on Siemen's PLC and other PLC brands. Programmers need one more step to convert the electrical symbol to the LAD symbolic form on the PLC. To achieve this, programmers must recognise and understand the function of LAD's symbol on Siemens's PLC. Some basic LAD symbols are shown in figure 17 below. Programmers using TIA Portal can find these symbols for creating a logic program from the list cards on the right of the main block (OB1), as shown in figure 18. Furthermore, whenever having trouble recognising the specific Ladder symbol's function, programmers can click on that symbol and press "F1" to call the help assistant in a new window. [17.]

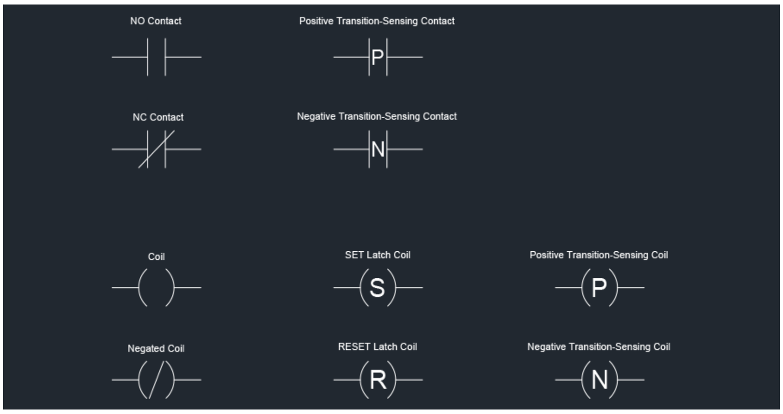

Figure 17. Some basic LAD symbols using in TIA Portal software Reprinted from Ladder Logic Symbols – All PLC Ladder Diagram Symbols [18].

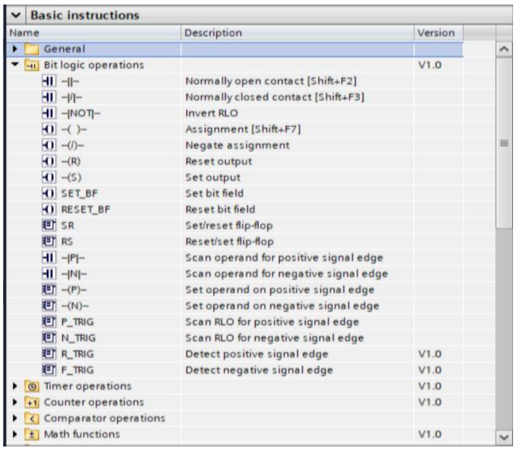

Figure 18. Bit logic operation symbols in the TIA portal Captured from TIA Portal.

In PLC CPU, all command and digital data are stored in bits, bytes, words and more, just like in a computer. The smallest digital data unit is "bit", also known as the binary number "0" for the false state or "1" for the true state. This binary state simulates the current flow operation on the electrical circuit. The "1" state determines the current flow through elements, and the "0" state determines no current flow through elements. A switch to turn the light on and off is an example of storing data in one bit. The switch close (state "1") determines a connection through the wire and vice versa. Programming in PLC is not limited to binary number. It is a combination of multiple complex values and condition.

Therefore, there is a byte which is a group of eight bits. Each bit has a separate location with its address ranging from 0 to 7. Moreover, two bytes combination is called a “word”, and four bytes cooperation demonstrate a “double word”. To be clear, a “word” support representing an integer number from -32768 to +32767. The last bit at 215 indicates telling a sign of integer number. The number is positive if bit position 215 is “1” and negative if bit position 215 is “0”.

In PLC programming, each logic operation and other operation elements retain a specific PLC value to understand and store data. Apart from binary, data values also can be decimal, integer, hexadecimal number, or characters. These values range from minimal to considerable value, such as from binary number to real number. Programmers need to concern the range of value when programming on TIA Portal. This concern helps the programmers choose the correct data type, defined as the data element’s size and the bit’s structure within the data. More information regarding data type is illustrated in figure 19 below, including size, range, and entry value for programmers to refer to when programming on TIA Portal.

Data type	Size	Range	Constant Entry Examples
Bool (Boolean)	1 bit	0 to 1	TRUE, FALSE, 0, 1
Byte (byte)	8 bits (1 byte)	16#00 to 16#FF	16#12, 16#AB
Word (word)	16 bits (2 bytes)	16#0000 to 16#FFFF	16#ABCD, 16#0001
DWord (double word)	32 bits (4 bytes)	16#00000000 to 16#FFFFFFFF	16#02468ACE
Char (character)	8 bits (1 byte)	16#00 to 16#FF	'A', 't', '@'
SInt (short integer)	8 bits (1 byte)	-128 to 127	123, -123
USInt (unsigned short integer)	8 bits (1 byte)	0 to 255	123
Int (integer)	16 bits (2 bytes)	-32,768 to 32,767	123, -123
UInt (unsigned integer)	16 bits (2 bytes)	0 to 65,535	123
DInt (double integer)	32 bits (4 bytes)	-2,147,483,648 to 2,147,483,647	123, -123
UDInt (unsigned double integer)	32 bits (4 bytes)	0 to 4,294,967,295	123
Real (real or floating point)	32 bits (4 bytes)	+/-1.18 x 10^{-38} to +/-3.40 x 10^{38}	123.456, -3.4, -1.2E+12, 3.4E-3
LReal (long real)	64 bits (8 bytes)	+/-2.23 x 10^{-308} to +/-1.79 x 10^{308}	12345.123456789 -1.2E+40
Time (time)	32 bits (4 bytes)	T#-24d_20h_31m_23s_648ms to T#24d_20h_31m_23s_647ms Stored as: -2,147,483,648 ms to +2,147,483,647 ms	T#5m_30s 5#-2d T#1d_2h_15m_30x_45ms
String (character string)	Variable	0 to 254 byte-size characters	'ABC'
DTL[1] (date and time long)	12 bytes	Minimum: DTL#1970-01-01-00:00:00.0 Maximum: DTL#2554-12-31-23:59:59.999 999 999	DTL#2008-12-16- 20:30:20.250

[1] The DTL data type is a structure of 12 bytes that saves information on date and time in a predefined structure. You can define a DTL in either the Temp memory of the block or in a DB.

Figure 19. Data type table includes clarifying their size, range and constant entry examples Reprinted from Getting started with S7-1200 [17].

The unique of TIA Portal from Siemens is its symbolic programming. Programmers can create symbolic names or "tags" related to the memory address, I/O points or local variable used within a code block. The PLC CPU requires programmers to specify name tags and data type, and memory areas of a variable. The memory areas are diverse with inputs(I), outputs(Q), bit memory/M), data block (DB), and local or temporary memory (L). User programs access these memory areas to read from and write to stored data. Each memory location is unique and numbered to identify which not causing overlap with others. The overlapping between tags can destroy the system's logic and make the system unoperated. Figure 20 below shows the correlation between PLC tags and the "absolute" addressing for better understanding how the CPU structure and address the memory area.

Memory area	Description
I Process image input	The CPU copies the state of the physical inputs to I memory at the beginning of the scan cycle. To immediately access or to force the physical inputs, append a ":P" to the address or tag (such as "Start:P" or I0.3:P).
Q Process image output	The CPU copies the state of Q memory to the physical outputs at the beginning of the scan cycle. To immediately access or to force the physical outputs, append a ":P" to the address or tag (such as "Stop:P" or Q0.3:P).
M Bit memory	The user program reads and writes the data stored in M memory. Any code block can access the M memory. You can configure addresses within M memory to retain the values of the data after a power cycle.
L "Temp" memory	Whenever a code block is called, the CPU allocates the temporary, or local, memory (L) to be used during the execution of the block. When the execution of the code block finishes, the CPU reallocates the local memory for the execution of other code blocks.
DB Data block	Use the DB memory for storing various types of data, including intermediate status of an operation or other control information parameters for FBs, and data structures required for many instructions such as timers and counters. You can specify a data block to be either read/write or read only. You can access data block memory in bits, bytes, words, or double words. Both read and write access is permitted for read/write data blocks. Only read access is permitted for read-only data blocks.

Whether you use a tag (such as "Start" or "Stop") or an absolute address (such as "I0.3" or "Q1.7"), a reference to the input (I) or output (Q) memory areas accesses the process image and not the physical output. To immediately access or force the physical (peripheral) input or output in your user program, append the reference with ":P" (such as "Stop:P" or "Q0.3:P").

Figure 20. Description of Memory area Reprinted from Getting started with S7-1200 [17].

3.2.4 Explaining the Project's Algorithm

From the picture of figure 21 below, this project's idea is to predict the remaining useful life of induction motor powered by VFD. The PLC CPU plays an essential role as the centre in the project, which sends the command and receives data from the VFD, pushes data to SQL database, and views real-time the performance of VFD on the human-machine interface (HMI). The VFD communicate with PLC CPU through a MODBUS connection. With this connection, the PLC CPU as a master can send digital data as a command to the VFD. On the other hand, the VFD as a slave device sends information under digital data type, including direction, current, line voltage, power, and frequency, to the

PLC CPU. After that, those data are pushed to the Structured Query Language (SQL) database for storing. This sending requires some scripted programming in Visual Basic (VB) language integrated into the TIA Portal project for execution. Extending if possible, these data can be queried on a webserver by time from the database for other people viewing. Inside the PLC CPU is the logical ladder programs that are uploaded from TIA Portal software. The designing process of predictive maintenance includes many steps, as the flow chart in figure 21 below. The simulation feature allows programmers to imitate the command and observe the parameters' signal before uploading the ladder logic program to the PLC. The simulation function limits unexpected faults of the logic that can destroy the system's running test on the field. From the flow chart in figure 21, the "Configure Modbus" and "Create PLC program" stage has the individual flow chart for explaining the algorithm inside.

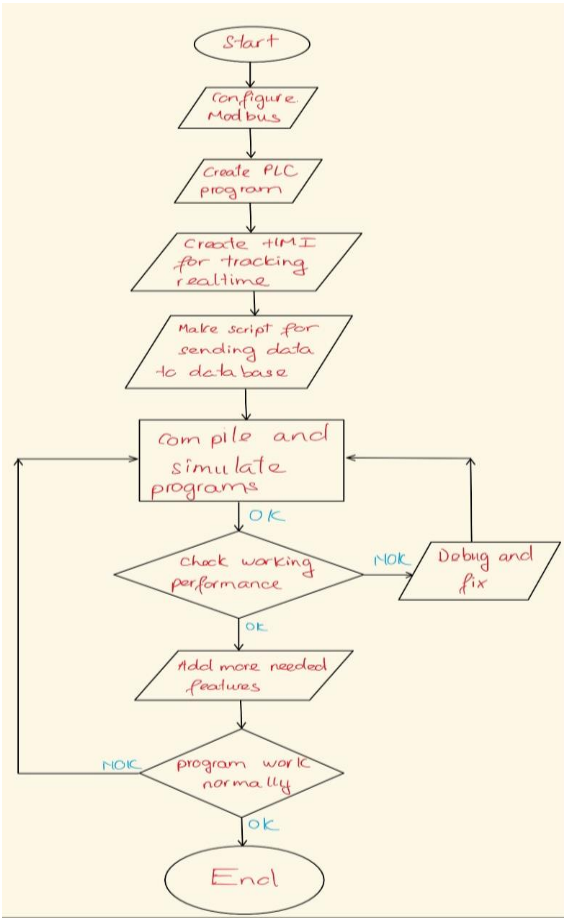

Figure 21. The process of creating PLC's project on TIA Portal Captured from OneNote.

Before creating the ladder logic program, programmers need to configure the MODBUS connection between the PLC as the master device and the VFD as the slave device in this project. The MODBUS configuration includes five steps, as figure 22 below. The MODBUS communication is the key to successful transmit data between Siemens PLC and the VFD FRE-740. The first step of the MODBUS configuration is setting characteristics on the slave device VFD FRE-740. At this step, the relevant parameters on the VFD using for MODBUS need to be configured appropriate according to the predefined role. These parameters can be configured either manually or remotely by software. Significantly, parameter 77, which is defined as a parameter write selection mode, needs to change to “2” for enabling writing mode on the slave device regardless of operation status. The next step is the integration of the MODBUS communication module (CM). Programmers need to integrate CM 1241 (RS422/485) into the TIA Portal project for enabling the MODBUS communication port on the master device. After having the CM 1241 (RS422/485), programmers also need to set the baud rate appropriately, parity check, data bits, and stop bits parameters the same value on both the PLC and the VFD. The similarity setting of those parameters on both master and slave device ensure the stable in transmitting data signals. The next step after finishing setting on both master and slave devices is adding master-slave function block into the main program of TIA Portal. These function blocks are the “MB_COMM_LOAD” and “MB_MASTER” block.

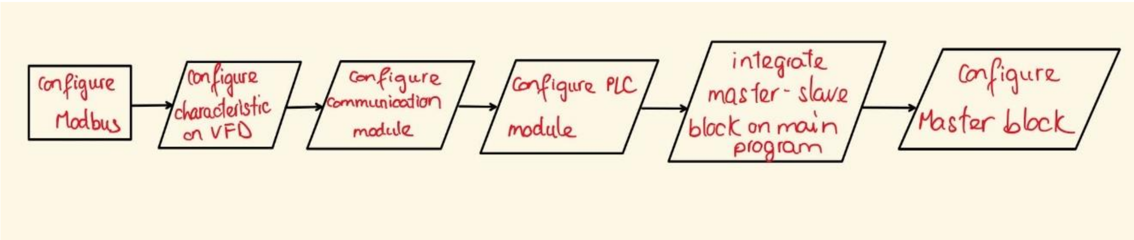

Figure 22. MODBUS configuration steps use in the Predictive maintenance project.

The “MB_COMM_LOAD” is determined as a MODBUS RTU communication port. This block must be called for the PLC to transmit data through the Modbus communication port. Completed configuration on this port help programmers to use and configure the “MB_MASTER” block. The “MB_MASTER” block allows the PLC program to communicate as a Modbus master on the point-to-point module. Hence, as a master, the PLC can access data in the Modbus slave device, a VFD. With the helping function by pressing F1, the description and information of these blocks’ parameters are shown in the popup window. After reviewing all Modbus configuration steps, the ladder logic program’s algorithm is then established in the following networks of the main block-OB1.

The ladder logic program's algorithm is divided into two flow charts for simply explaining purpose. The first chart in figure 23 shows the primary operation of the predictive maintenance process after pressing the start button. Moreover, the second graph in figure 27 shows the system's algorithm in running mode.

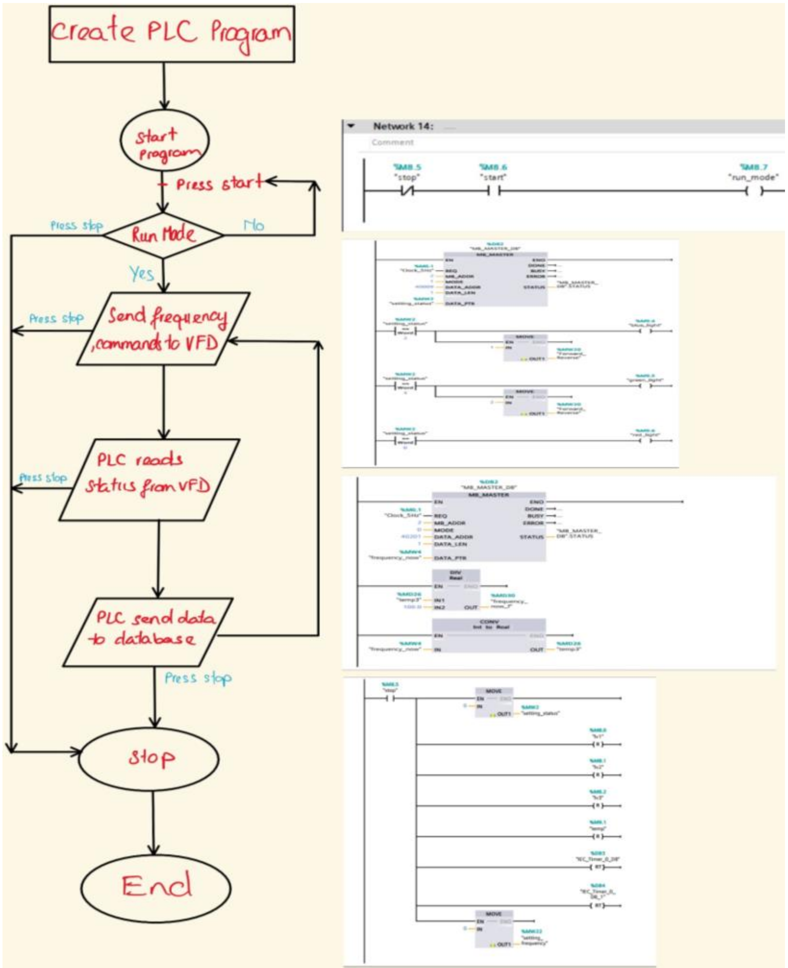

Figure 23. The flow chart explains the primary operation of the predictive maintenance project.

Figure 23 shows that the program goes to running mode after pressing the start button. The picture on the right-hand side of the flow chart illustrates each stage in the ladder program language. The start switch is a normally open contact. The switch close enables the system to go into running mode. At this stage, the PLC has many command signals send to the slave device and begin turning the induction motor. With the MB_MASTER_DB function block, the PLC as a master device can write register #40009's value from DATA_PTR to the slave device. By default, the manual of VFD FRE-740 indicates register #40009 as the inverter status, which means force the VFD to turn in forward/reverse rotation or stop turning. The VFD force the VFD to turn in the forward rotation if the slave device receives “2” from the receiving register #40009 of the master device.

Similarly, the VFD is forced to turn in the reverse rotation if the value in register #40009 receiving from the PLC is one. Other control modes happen the same procedure if programmers send another register containing appropriate values to the slave device. Figure 24 below is the example of value on register #40009 with their corresponding control command. The other proper writing and reading registers of the VFD FR-E740 are shown in figure 25 and 26, with the function and appropriate value. In figure 23, the function block “MOVE” indicates the status of the induction motor storing on the database. This function block moves the input “IN” parameter’s value to the output parameter “OUT“ when the condition in the enable parameter “EN” is satisfied. Furthermore, the enabling colour of the warning light on the field helps operators and field engineers easily observe the induction motor’s status and direction.

<Inverter status/control input instruction>

Bit	Definition	
	Control input instruction	**Inverter status**
0	Stop command	RUN (inverter running) *2
1	Forward rotation command	Forward rotation
2	Reverse rotation command	During reverse rotation
3	RH (high-speed operation command)*1	SU (up-to-frequency)
4	RM (middle-speed operation command)*1	OL (overload)
5	RL (low-speed operation command)*1	0
6	0	FU (frequency detection) *2
7	RT (second function selection)	ABC (fault) *2
8	AU (current input selection)	0
9	0	0
10	MRS (output stop) *1	0
11	0	0
12	RES (reset) *1	0
13	0	0
14	0	0
15	0	Fault occurrence

<Operation mode/inverter setting>

Mode	Read Value	Written Value
EXT	H0000	H0010
PU	H0001	—
EXT JOG	H0002	—
PU JOG	H0003	—
NET	H0004	H0014
PU+EXT	H0005	—

The restrictions depending on the operation mode changes according to the computer link specifications.

*1 The signal within parentheses is the default setting. The description changes depending on the setting of *Pr.180 to Pr.184 (input terminal function selection)*. *(Refer to page 139)*
Each assigned signal is valid or invalid depending on NET. *(Refer to page 205)*

Figure 24. The value meaning in register #40009 Reprinted from Inverter FR-E740 instruction manual [19].

(6) Modbus registers

● System environment variable

Register	Definition	Read/write	Remarks
40002	Inverter reset	Write	Any value can be written
40003	Parameter clear	Write	Set H965A as a written value.
40004	All Parameter clear	Write	Set H99AA as a written value.
40006	Parameter clear *1	Write	Set H5A96 as a written value.
40007	All parameter clear *1	Write	Set HAA99 as a written value.
40009	Inverter status/control input instruction*2	Read/write	See below.
40010	Operation mode/inverter setting *3	Read/write	See below.
40014	Running frequency (RAM value)	Read/write	According to the Pr.37 settings, the frequency and selectable speed are in 1r/min increments.
40015	Running frequency (EEPROM value)	Write	

*1 The communication parameter values are not cleared.
*2 For write, set the data as a control input instruction.
For read, data is read as an inverter operating status.
*3 For write, set data as the operation mode setting.
For read, data is read as the operation mode status.

Figure 25. MODBUS registers use for controlling the VFD FR-E740 Reprinted from the Inverter FR-E700 instruction manual [19].

●Real time monitor

Refer to page 154 for details of the monitor description.

Register	Description	Unit
40201	Output frequency/speed *1	0.01Hz/1
40202	Output current	0.01A
40203	Output voltage	0.1V
40205	Output frequency setting/speed setting *1	0.01Hz/1
40207	Motor torque	0.1%
40208	Converter output voltage	0.1V
40209	Regenerative brake duty	0.1%
40210	Electronic thermal relay function load factor	0.1%
40211	Output current peak value	0.01A
40212	Converter output voltage peak value	0.1V
40214	Output power	0.01kW
40215	Input terminal status *2	—

Register	Description	Unit
40216	Output terminal status *3	—
40220	Cumulative energization time	1h
40223	Actual operation time	1h
40224	Motor load factor	0.1%
40225	Cumulative power	1kWh
40252	PID set point	0.1%
40253	PID measured value	0.1%
40254	PID deviation	0.1%
40258	Option input terminal status*4	—
40259	Option input terminal status 2*5	—
40260	Option output terminal status *6	—
40261	Motor thermal load factor	0.1%
40262	Inverter thermal load factor	0.1%
40263	Cumulative power 2 Ver.UP	0.01kWh

Figure 26. Another MODBUS register use for monitoring parameter on the VFD FR-E720 Reprinted from the Inverter FR-E700 instruction manual [19.]

After sending the command to the slave device, programmers also need another MB_MASTER_DB function block for reading status and information on the slave device. The function block MB_MASTER_DB has the value "1" on parameter MODE if the PLC write data to the VFD. On the contrary, it has the value "0" on parameter MODE if the PLC read data from the slave device. At this stage, the frequency value that the PLC read from the VFD is an integer number. The conversion method is needed for engineers to easily observe these values on the human-machine interface display and database. The database storing requires an integrated script programming in Visual Basic (VB) on the PLC. The explaining of this Script is illustrated later in the following chapter. The PLC repeat the process of writing data value, reading data value to/from the VFD, and storing them on the database. This procedure only stops if the stop button or stop switch is energised. The stop button is a normally close (NC) contact switch and is located at the beginning of each network. They break the contact of all networks when it energises and therefore shut down the system. However, the PLC program has a network that includes the energising stop switch for resetting all temporary parameters, timers, and the value on the data address to the original value. The stop switch in this network forces the whole system to entirely shut down and ensure people's safety in contact with the machine.

In figure 27 below, the induction motor is designed to run in auto or manual mode. In running mode, the induction motor firstly runs in forward rotation for 10 seconds. This ramp-up time can be adjusted depending on the need of the application. In manual mode, the induction motor can run at three predefined frequency speeds: 10Hz, 30Hz or 50Hz. The induction motor can also be chosen to run any frequency speed between 0 and 50 Hz depending on the need. After pressing the "set frequency" button, the induction motor can switch from auto mode to manual mode. The idea of this algorithm is considered as the primary demand in the industrial field. The auto mode runs default products first, along with the ramp up time. Then, the system can switch to manual mode depending on product types. The manual mode is chosen to run different customised products that require the appropriate speed for each product line. For example, two predefined speed can use for running two other popular products on this production line. Moreover, there also have many other product lines with a specific speed requirement. This specific speed also increases overall equipment effectiveness (OEE) if the plant's output demands are high. Like figure 23, figure 27 has a network including ladder logic programs at each stage for representing.

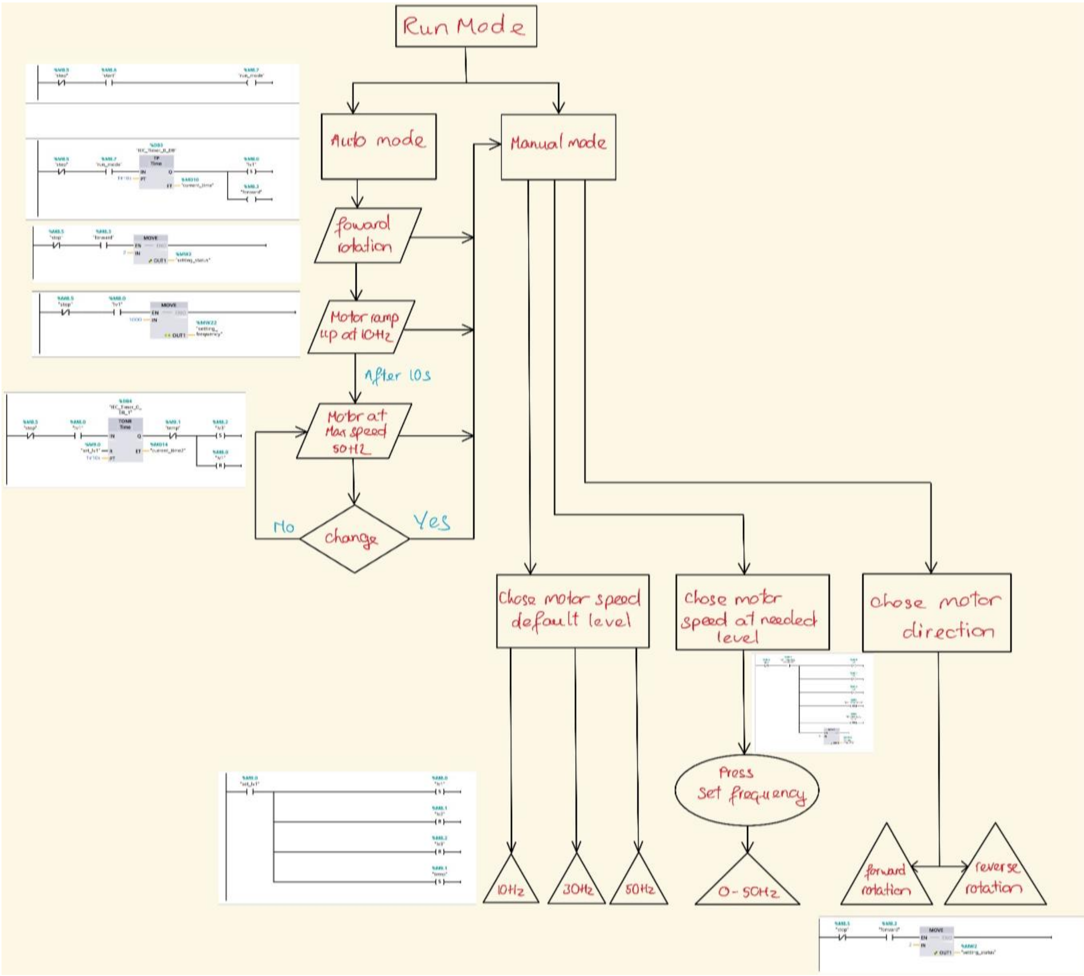

Figure 27. The auto mode and manual mode turn the induction motor at many speed levels and direction.

The auto mode is enabled instantly when the “start” variable has close contact. The output variable “run_mode” therefore energise and turn on the timer data block-DB3. This timer data block “TP” is called a generate pulse timer. The timer starts counting when a result of the logic operation (RLO) at input parameter “IN” change from “0” to “1”. This timer allows the output side of parameter “Q” to turn on during “PT” time and only turn back to the “0” stage after the elapsed time. It means that the set output and forward parameter state the logic “1” during the setting time at parameter “PT”. At the time set output “lv1” and output coil “forward” parameters are “on” stage, the “MOVE” data block does its function which transfer content at the input value to the output parameter. The corresponding MB_MASTER block does the job of sending these values to the slave device – the VFD. Consequently, the VFD receives commands from the master device and turn the induction motor in the forward rotation at a low speed for 10 seconds.

After 10 seconds, the program wants the VFD to continue turning in forward rotation but with the maximum frequency speed of 50Hz. The start on-delay timer “TON” is used for setting on the “lv3” parameter and reset the stage of the “lv1” parameter back to “0” after the ramp-up duration. The “TON” block runs an International Electrotechnical Commission (IEC) timer with a specified duration as on-delay. This “TON” timer must count on delay time in parallel with the “TP” timer to enable the “lv3” and empower the “lv1” parameter. Unlike the “TP” timer, the IEC timer of “TON” is reset if the RLO of input change to “0” before the time expires. Thanks to the set output function, the logic of parameter “lv1” remains unchanged to keep the “TON” timer count the on-delay time.

After achieving the maximum speed, the induction motor has finished the auto mode process since the start button is pressed. If there are any demand changes in the motor’s running frequency and direction during or after the auto mode, the induction motor’s manual mode control can be enabled. On the field, operators can choose to run the induction motor at three predefined running frequencies. For example, the operator wants to run the motor at the frequency speed 10Hz, which means speed level 1. On the ladder logic program, the program enables the set output “lv1” and put RLO of other running frequencies back to state “0” by enabling reset output of parameters “lv2” and “lv3”. The ladder program’s logic for enabling the motor to run at frequency 30Hz and 50Hz has the same structure. Moreover, the application’s demand also wants the motor to run at a specific frequency speed between 0Hz and 50Hz. In this case, the button “Set frequency”, which is the parameter “set_frequency_demand”, is activated. The activation of this parameter put the temporary system stop but still in run mode. On the ladder logic program, the setting on parameter “set_frequency_demand” drag the “setting_frequency “parameter is reset, and the slave device receives 0Hz value and wait for the new input frequency speed from the master device. The operator then can make the slave device run in a new frequency by typing any demand frequency values in the text field on the human-machine interface, like in figure 28 below. The slave device runs the induction motor in the demand frequency speed by receiving the content from “DATA_ADDR” of register #40014. The explanation of the ladder logic program ends at this stage. The following chapters are the explanation of designing a human-machine interface for the system.

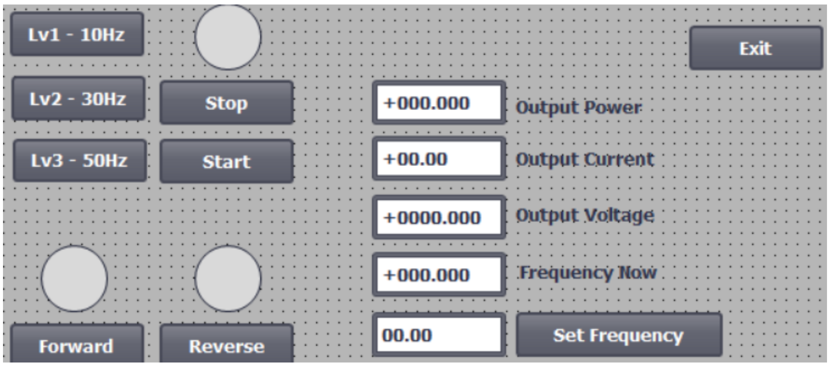

Figure 28. A part of the human-machine interface controls and monitors parameters on the induction motor.

3.2.5 Control and Monitoring on The Human-Machine Interface

The human-machine interface (HMI) design is an essential step that allows operators and other field engineers to control and monitor the machine. To add an HMI for designing, programmers need to click on “Add new device” on the top left corner of the TIA portal software and chose a compatible HMI on tab HMI. In advance, the HMI of WinCC Runtime Professional is located in the PC systems tab, allowing programmers to have more customisation in graphical design and advanced monitoring features. On the same “Add new device” window, programmers can choose “PC systems” and click on “WinCC RT Advanced” in the “SIMATIC HMI application” tab, as in figure 29. Programmers then need to create an HMI connection from the CPU to the WinCC RT Advanced PC station. This creation supports the software for finding the compatible terminal display.

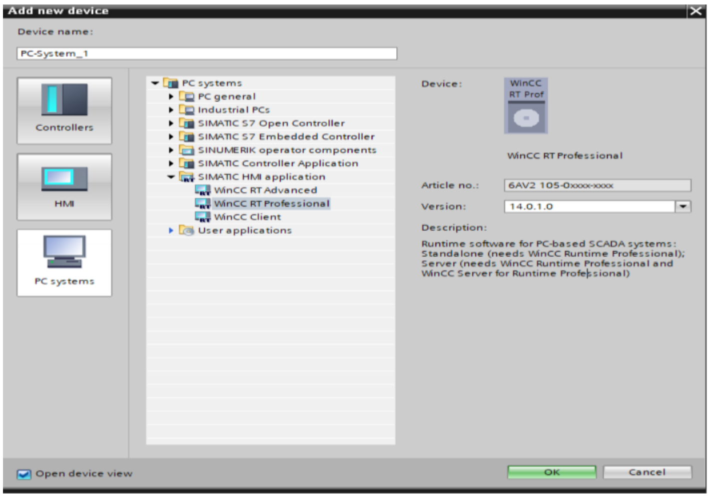

Figure 29. The WinCC RT Advanced HMI is integrated into the PLC program captured from the TIA Portal.

Consequently, programmers can start to design the control and monitoring interface on the HMI display. The HMI display of this project includes two parts which are the control part and graphical monitoring section. The control interface part is previously shown in figure 28, and the monitoring section is shown in figure 30 below. In figure 30, the monitoring part displays four real-time line graphs, including line current, line voltage, power and running frequency feed to the induction motor for field engineers to observe.

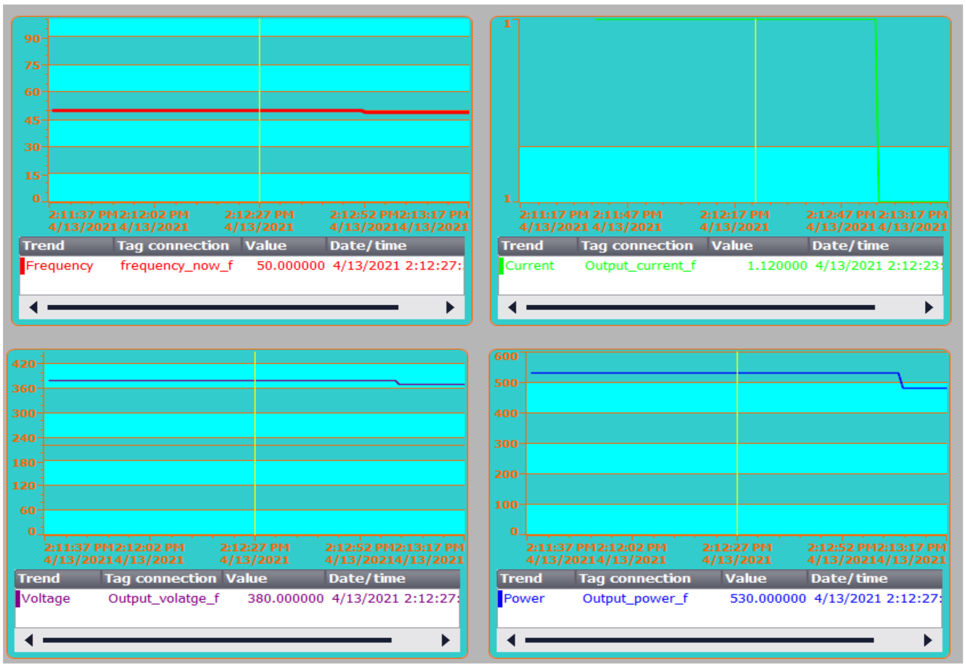

Figure 30. Four line-graph display the information value of the slave device.

Programmers can integrate many controllable and monitoring features on display by powerful objects in the toolbox locates on the PLC project’s right side. These objects include many buttons, comment text fields, elements for controlling and monitoring parameters in the project, such as slider, gauge meter, and more. The control and monitor commands are executed by assigning parameters, event action, format display, animations, and more in the chosen object’s properties. For example, the “start” parameter is assigned to the start button with the pressing event and the releasing event. Hence, the start switch is customised to go on state “1” when pressing and go back to state “0” when releasing the button, as in figure 31. The other objects have the same configuration and come with more customisation depend on their function.

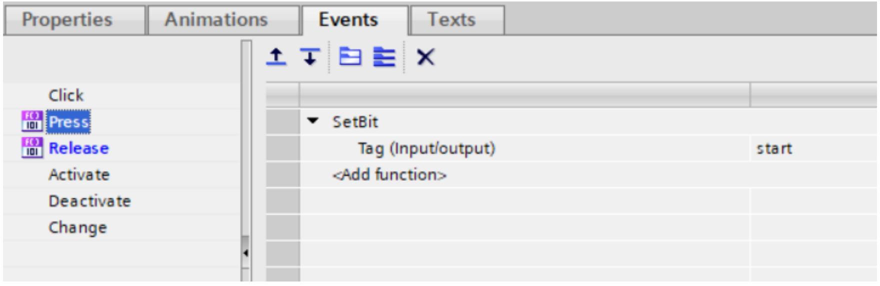

Figure 31. The configuration of the start button on the human-machine interface Captured from TIA Portal.

3.2.6 Historical Data and Script

As mentioned in the previous chapter, the PLC program needs integrated Visual Basic Compiler Scripts to retain data from the VFD on the database. Moreover, the TIA Portal also has historical data under comma-separated values (CSV) type, fetched if the data-base storing has malfunctioned. The CSV file includes data value, feature name and recording time of each value in plain text.

First, the historical data feature is configured on the TIA Portal program. Programmers need to choose the “Historical data” tab on the project tree and create data logs that include a data log name, file type, number of data records, and location path to store data. Each data logs can have many logging tags, which are the parameter’s value de-manding for storing. In each logging tags, programmers can change the acquisition mode, which is the method for fetching data. For example, if programmers chose cyclic as in figure 32, the data is record and store in a CSV file after the chosen duration time in the logging cycle attribute. Then this cycle is repeated until the program stop. There are two more choosing acquisition modes in WinCC RT Advanced: on-demand and on-change for programmers.

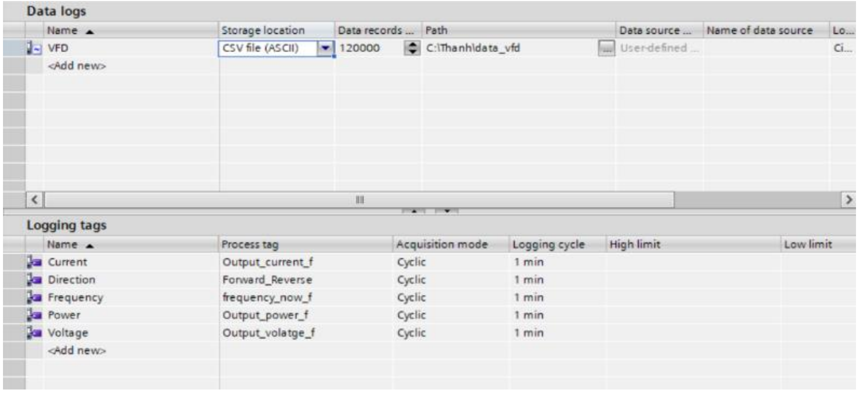

Figure 32. The historical data configuration captured from TIA Portal.

Apart from historical data, the TIA Portal can store data in an SQL database for further application usage. The program needs a Visual Basic compiler Script, as in figure 33, for achieving this purpose. From the Script in figure 33, the first line and the last line are the declaration name and ending command of the Script. The Visual Basic (VB) compiler uses the “Dim” statement from line 2 to line 10 to declare variables and determine their data types. The following statement in line 11 creates a portal for enabling TIA Portal to log into the local Microsoft SQL Server. The statement includes the name of the data source, “SQL server”, the name server, database name, username, and password for logging into the server. Following lines from line 12 to line 16 are the assigning parameters in the PLC program to declared variable in the VB. SQL syntax command on line 17 is an introductory statement in this Script which is used for writing data into the table in the SQL server.

Siemens offer programmers the Script along with their instruction for using in the demand application. Programmers merely modify some parameters accordingly under the instruction and can apply for the retaining data purpose.

```vb
Sub write_all()
Dim objConnection
Dim strConnectionString
Dim lngValue
Dim Voltage
Dim Power
Dim Frequency
Dim Status
Dim strSQL
Dim objCommand
strConnectionString = "DRIVER={SQL Server}; SERVER=DESKTOP-6MMU3Q4\WINCCPLUSMIG2014; DATABASE=VFD_test;UID=;PWD="
lngValue = SmartTags("Output_current_f")
Voltage = SmartTags("Output_volatge_f")
Power = SmartTags("Output_power_f")
Frequency = SmartTags("frequency_now_f")
Status = SmartTags("Forward_Reverse")
strSQL = "INSERT INTO VFD_all (Direction, VFD_power, VFD_voltage, Frequency, VFD_current, Update_time) VALUES ("& Status &", "& Power &", "& Voltage &", "& Frequency &","& lngValue &",GETDATE());"
Set objConnection = CreateObject("ADODB.Connection")
objConnection.ConnectionString = strConnectionString
objConnection.Open
Set objCommand = CreateObject("ADODB.Command")
With objCommand
.ActiveConnection = objConnection
.CommandText = strSQL
End With
objCommand.Execute
Set objCommand = Nothing
objConnection.Close
Set objConnection = Nothing


End Sub
```

Figure 33. Visual Basic (VB) script integrates into the PLC project for storing data on the Microsoft SQL Server database [20].

3.2.7 MODBUS-RTU Configuration

From the previous chapter, the point to point or MODBUS communication has been briefly introduced. In this chapter, the configuration of MODBUS between Variable Frequency Drive (VFD) and the PLC communication module is explored more.

Modbus is a point-to-point communication protocol developed for use with programmable logic controllers (PLCs). In simple term, the function of Modbus is to transmit information over serial lines between electronic devices. This communication requires at least two devices, one with the role of the Modbus Master and another in the network is Modbus slave. In the network, the master device function is sending/writing data to slave devices. Following, Slave devices in the same network receive and respond to the requested information from the master device. The information that the slave device receives from the master device is the amount of voltage. These values are a series of binary bits, zeroes as a positive and negative voltage, and their typical transmission speed of these bits is 9600 bits per second/baud. Modbus has two transmission modes: American Standard Code for Information Interchange (ASCII) and the Remote Terminal Unit (RTU). [21.]

The PLC CPU enable MODBUS-RTU communication through the CPU extension with the interface RS485. The PLC CPU mainly communicate with the VFD from the CM1241 communication module to the communication port on the variable frequency drive FR-

E740 0.75kW. The pinout of the RS485 communication port on the CM1241 includes nine pins, connects to the 8-pins registered jack 45 (RJ45) port on the VFD. The information of the 9-pins on the RS485 connector is shown in figure 34. Moreover, the information of 8 pins on the RJ45 connector is shown in figure 35. With a robust connection, the PLC can write and read data from the slave device by connecting the sending and the receiving data pins on both devices. Indeed, PIN 3 on module CM1241 connects to PIN 3 and 5 on the Inverter's PU connector for sending and receiving data. However, the signal transmits and receive in pair of A and B on both devices. Therefore, PIN 8 on module CM1241 also needs to connect to PIN 4 and 6 on the Inverter's PU connector. Finally, the connection between the ground PIN 5 on module 1241 and ground PIN 7 on the inverter connector is essential for creating a robust and safe transmission signal.

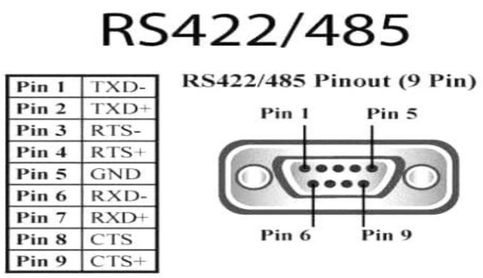

Figure 34. All pins of the RS485 connector Reprinted from RS-485 Masterclass - What is it and Why use it? [24.]

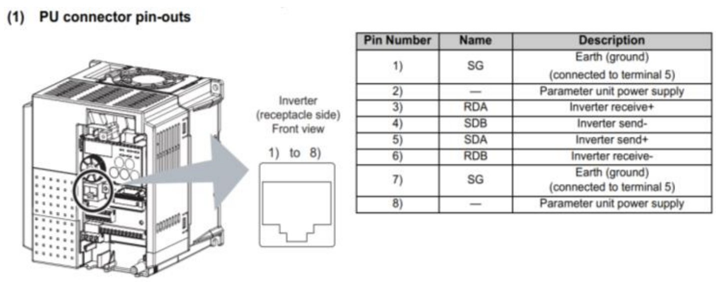

Figure 35. All pins of PU connector on the variable frequency drive FRE-700 Reprinted from Inverter FR-E700 instruction manual [19.]

After forming the physical connection, both devices' configuration is needed for successfully transmitting and reading data. Programmers firstly do configuration on the VFD by switching the VFD into physical unit (PU) mode. The PU mode allows the programmer to manually setting up some parameters related to Modbus communication. The parameter 77 (Pr77) on the VFD is the parameter write selection mode. Furthermore, the

MODBUS configuration on the VFD is achieved by setting multiple variables in figure 36 in the setting value. [19,211-214.]

Parameter Number	Parameter Name	Description	Setting Value
77	Parameter write selection	Parameter write is enable in any operation mode regardless of operation status	2
79	External operation modeExternal operation mode	External operation modeExternal operation mode	0
340	PU operation mode	Operation mode	1
117	PU communication station number	Inverter station number	2
118	PU communication speed	Communication speed. The setting value * 100 equals 9600bps	96
120	PU communication parity check	Even parity check	2
122	PU communication check time interval	Communication check (signal loss detection) time interval	2
343	Communication error count	Display the number of communication errors during MODBUS_RTU communication	0
502	Stop mode selection at communication error	Stop at fault occurence	0
549	Protocol selection	MODBUS_RTU protocol	1
338	Communication operation command source	Start command source communication	0
339	Communication speed command source	Frequency command source communication	0
550	NET mode operation command source selection	Automatic communication option recognition	9999
551	PU mode operation command source selection	USB automatic recognition	9999

Figure 36. Needed parameter configuration use for MODBUS on the VFD Modified from Inverter FR-E700 instruction manual [19,211-214].

After finishing the inverter configuration, the next step is MODBUS-RTU configuration on TIA Portal's program before programming ladder logic. After integrating the CM1241 module into the PLC project, programmers choose the properties tab on this module and configure the communication setting as figure 37 below. Like the VFD configuration, programmers must choose the same baud rate, parity check, number of data bits per character and stop bits. These value on the program should match with the value on the Inverter to create the transmission. This step is the end of the MODBUS-RTU configuration on both master and slave device in the network. Programmers can start to program on the project for the application.

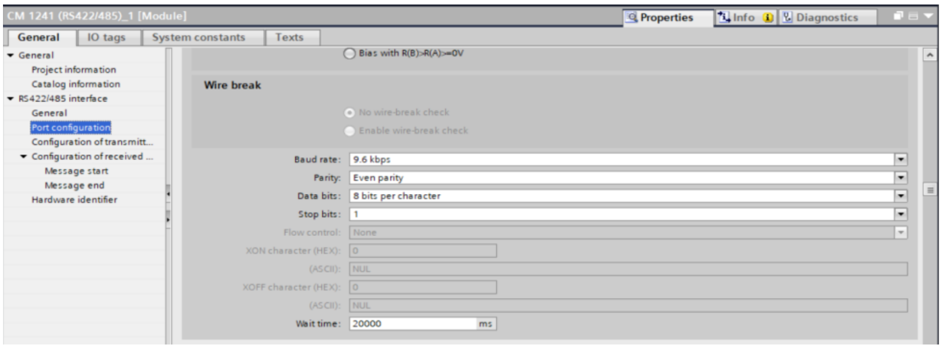

Figure 37. CM1241 configuration for MODBUS-RTU communication Captured from TIA Portal.

3.3 Variable Frequency Drive FR-E740

3.3.1 Variable Frequency Drive Function

A variable frequency drive (VFD), also known as an AC drive or Inverter, is a motor controller that drives an AC motor at a variable speed by varying its power supply frequency and voltage. By controlling the frequency and voltage of power supplied to the motor, the VFD's function subsequently is the motor speed adjustment. The applied frequency is proportional to the rotating speed of the motor. In other words, the faster frequency applied to the motor, the more round per minutes (RPMs) that motor can cycle. Thus, the VFD can turn up and turn down the motor speed to meet the application's required speed. [26.]

The most popular benefit of the electric drive is its saving energy function to improve system efficiency. Other benefits of the VFD worth mentioning are reducing the mechanical stress on the machine, parameters feedback, and the system's control. First, the VFD is best suited in the variable torque load application, where the motor speed must change according to the demands. Without the VFD, the fixed AC motor speed and torque make the motor operate at a constant speed and torque. It means that these energies are used more than the system's actual need, which determines the wasting energy. On the other hand, the output power is controlled by directly changing the supplied frequency and voltage to the motor. At low demand, the motor can run slowly with low power consumption, increase efficiency, and achieve energy savings. As a result, the VFD controlled fans in the Heating, Ventilating, and Air Conditioning (HVAC) system have saved significantly up to 70% of the energy consumption. In the production plant, this energy reduction means large money-saving in the operating cost. Second, VFD usage can reduce mechanical stresses and make the industrial process operate more smoothly. Moreover, there are high starting torque and inrush current when powering the AC motor without the VFD usage. The inrush current is the maximum peak current that appears at the beginning of turning on the motor. The inrush current value is twice or ten times greater than the normal rated current, as in the diagram of figure 38 below. The high starting torque can affect the motor's mechanical system, and the inrush current can damage the electrical system. These factors are determined as one of the causes shorten the motor's life span. When the VFD starts the AC motor, the system gradually ramp-ups to the operating speed. Therefore, the motor can start smoothly with the low starting torque and the minimisation of inrush current. Third, another great thing about

VFD is that it can be controlled with a PLC or a controller. With the MODBUS communication protocols explaining in the previous chapter, the VFD can help plant engineers monitor and control motor speeds on the need through RS485 signal cable like the picture in figure 39 below. Hence, most irregularities on the motor can be watched and detected thanks to the real-time monitoring feature. [28; 27; 25.]

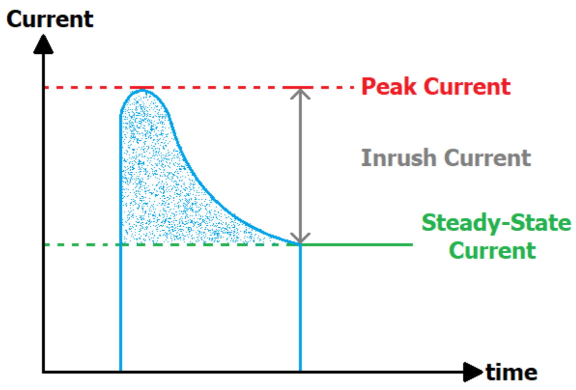

Figure 38. The diagram shows the maximum of inrush current compare to the steady-state current Reprinted from What is Inrush Current and How to Limit it? [27].

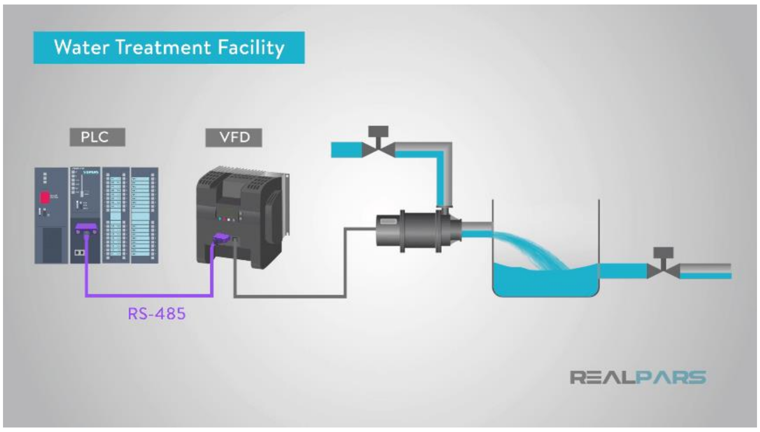

Figure 39. The VFD help engineer to control and monitor the motor speed of water treatment facility application Reprinted from What is a VFD? [25].

3.3.2 Variable Frequency Drive and V/F Characteristic

One of the most popular features of the VFD is its constant Voltage/Frequency (V/F) apply to the induction motor. This information is valuable for helping to simulate the induction motor data in the project. The benefit of keeping the V/F constant is that it can maintain the induction motor speed despite having a load. For keeping the speed of the induction motor stable with load, the idea is that the motor slip should be minimised as much as possible to limit the drop-in speed with the load. The slip for a given torque is dependent and disproportional to the amplitude of the rotating flux wave. Therefore, the slip can be reduced by enhancing the rotating flux. The rotating speed of the flux wave can be adjusted by controlling the output frequency of the Inverter. Besides, the desired magnitude of the flux is adjusted at its full (rated) value regardless of the rotation speed. Indeed, this requirement can be achieved by making the output voltage from the VFD varying appropriately with the frequency. It turns out that the flux wave amplitude is proportional to the output voltage and inversely proportional to the supplied frequency. Therefore, the flux wave can have a constant value if the Inverter's supplied voltage varies in direct proportion to the frequency. This operation mode is also mean keeping the voltage/frequency (V/F) ratio constant. Apart from maintaining the induction motor speed having load, the V/F ratio constant also prevents the motor from drawing excessive current and saturating its core. Hence, it can prevent excessive heat to the motor and extend the motor's lifetime. The constant ratio of V/F is also applied in the VFD FR-E740 of the project. [29, 325.]

3.3.3 Variable Frequency Drive and Induction Motor

The induction motor is generally chosen based on the application of the field and VFD's technical characteristic. The application requirement includes the speed range of induction motor, load capacity and more. However, the project merely simulates the pattern and approach the goal with no load requirement. Hence, this project's chosen induction motor only based on the specification compatible between the VFD and the induction motor. The VFD FR-E740-0.75K may supply three-phase output voltage from 380 to 480 volts, 1.6A rated current and functional motor capacity below 0.75kW. From the nameplate in figure 40 below, the electric motor 0.37kW Universal B3 from Defan is considered suitable because of its compatible specification. The suitable converter should meet all criteria that can power the chosen induction motor, such as supply voltage, frequency, and rated current motor and motor capacity. For example, the selected VFD has a rated current higher than the chosen motor's rated current. This induction motor also needs a supply voltage at 380 volts in star connection with the rated frequency 50Hz that the converter can supply. For the installation, the three wires from the Inverter go directly to the U, V and W terminal of the induction motor.

Figure 40. The three-phase motor nameplate of 0.37kW Defan Universal B3 Reprinted from eBay [online].

3.3.4 Simulate Induction Motor Data

The voltage and frequency data on the induction motor is simulated based on the V/F characteristic. The average output voltage of the VFD that apply to the induction motor can have a maximum value of 380 volts at 50 Hz as it cannot exceed the power supply voltage of the VFD. Hence, the V/F ratio can be turned out with the division of 380/50. As a result, the supposed V/F ratio is 7,6. When frequency changes, the corresponding average output voltage can be determined by multiplying frequency with 7,6. For example, the induction motor's average input voltage is 228 volts at 30 Hz or is 76 volts at 10Hz. Other values that the VFD can supply to the induction motor can be referenced from figure 41 below.

● **Three-phase 400V power supply**

	Model FR-E740-□K(SC)*9(-C)*10	0.4	0.75	1.5	2.2	3.7	5.5	7.5	11	15
Applicable motor capacity (kW)*1		0.4	0.75	1.5	2.2	3.7	5.5	7.5	11	15
Output	Rated capacity (kVA)*2	1.2	2.0	3.0	4.6	7.2	9.1	13.0	17.5	23.0
Output	Rated current (A)*7	1.6 (1.4)	2.6 (2.2)	4.0 (3.8)	6.0 (5.4)	9.5 (8.7)	12	17	23	30
Output	Overload current rating*3	150% 60s, 200% 3s (inverse-time characteristics)								
Output	Voltage*4	Three-phase 380 to 480V								
Output	Regenerative braking torque *5	100%		50%	20%					
Power supply	Rated input voltage/frequency	Three-phase 380 to 480V 50Hz/60Hz								
Power supply	Permissible AC voltage fluctuation	325 to 528V 50Hz/60Hz								
Power supply	Permissible frequency fluctuation	±5%								
Power supply	Power supply capacity (kVA)*6	1.5	2.5	4.5	5.5	9.5	12	17	20	28
Protective structure (JEM1030)		Enclosed type (IP20). IP40 for totally enclosed structure series.								
Cooling system		Self-cooling		Forced air cooling						
Approximate mass (kg)		1.4	1.4	1.9	1.9	1.9	3.2	3.2	6.0	6.0

Figure 41. Characteristic table of model FR-E740 use three-phase 400V power supply Reprinted from Inverter FR-E700 instruction manual. [19,326].

Moreover, the motor efficiency and power factor of the induction motor in figure 42 are proportional to the load. When the induction motor operates at full load, the maximum speed achieves and almost reach the synchronous speed. Moreover, the VFD can limit the inrush current, making the motor current not exceed the rated current on the motor nameplate. Hence, the induction motor current slightly increases from no load to full load when the VFD is used.

Based on those conclusions above, the VFD's induction motor experience full load with the maximum speed when the current, power factor and efficiency reach the rated values on the induction motor nameplate. Indeed, the VFD output power is the input power of the induction motor. The output load of the induction motor increase means more power supply to the motor. According to equation (2), the simulation input power of the induction motor can be calculated as it is proportional to the increasing power factor (cos φ).

$$Pin = \sqrt{3} * U * I * \cos\varphi \tag{2}$$

The random data of induction motor are generated, which imitate the curve pattern in figure 42. Indeed, the random data is limited by the rated value on the nameplate of chosen induction motor.

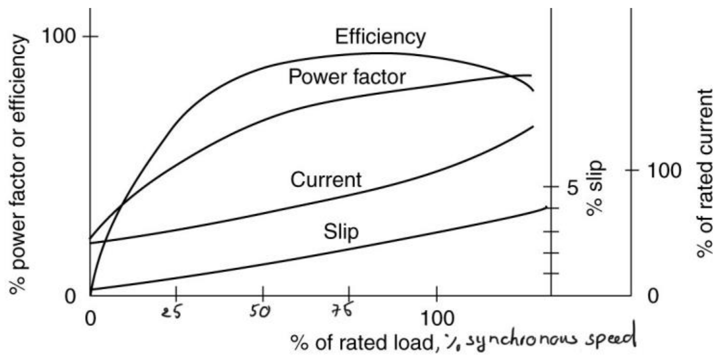

Figure 42. The characteristic curve of induction motor Modified from Three Phase Induction Motor Performance [37.]

The induction motor's healthy data simulation (in Appendix 1) is presented at each frequency in the range of 0 to 50Hz. Based on this table, the random data is performed by using the random method of Python. At each selected frequency, the corresponding line current, line voltage, input power of the induction motor is created randomly from the supposed healthy state to faulty in the limited range. As in figure 43 below, the data frame simulating at 16Hz is created in Python and show on the last five rows.

```
np.random.seed(2) #reserve random data values
data_16 = {
    "Voltage-V": 121.6,
    "Frequency-Hz": 16,
    'Current-A': sorted(pd.Series(np.random.uniform(61, 72, 581)/100)),
    'Power-W': sorted(pd.Series(np.random.uniform(41, 110, 581))),
    "Forward": 0,
    "Reverse": 1
}

df_16Hz = pd.DataFrame(data_16) #Create data frame
df_16Hz.tail(5) #Show last 5 rows of dataframe created by random number
```

	Voltage-V	Frequency-Hz	Current-A	Power-W	Forward	Reverse
576	121.6	16	0.718775	109.425308	0	1
577	121.6	16	0.719324	109.442841	0	1
578	121.6	16	0.719445	109.651466	0	1
579	121.6	16	0.719811	109.700336	0	1
580	121.6	16	0.719835	109.924199	0	1

Figure 43. One dataset is created by using Python's NumPy method Captured from Jupyter notebook

3.4 Python and Predictive Model Approaching

3.4.1 Python and Jupyter Notebook

As mentioned in previous chapters, the data analysis and the prediction of remaining useful life (RUL) on induction motor are executed by Python. Python language needs an integrated development environment (IDE) to make data analysis and predicting RUL action easier. IDE is a coding tool that allows data analysts or users to inspect data by writing, testing, and debugging programming code straightforwardly. One of the most popular Python IDE is Jupyter notebook which allows data scientist and analyst to statistics and visualises data more efficiently than the others. Jupyter Notebook is a web application based on the server-client structure, which was born in 2014. Indeed, Jupyter Notebook works as an IDE and has a role as a presentation or education tool perfect for fresher starting work with data science. With Jupyter Notebook, user can easily express data visualisation using data visualisation libraries such as Matplotlib and Seaborn. Furthermore, Jupyter Notebook also supports markdown allowing users to write the note and instant perform statistic idea directly on the project. Because of its benefits, Jupyter Notebook is chosen for inspecting data and creating a predictive model in this project.

3.4.2 Performing Statistic Data on Induction Motor

As mentioned in the previous chapter, predictive maintenance is approached in five steps: acquiring data, pre-process data, identifying condition indicators, selecting a model, and training the model. The first step of acquiring data has been done by storing data on SQL local server. The python-based framework needs to create a connection for logging into the SQL server and fetching data. The requirement variable for logging into SQL server includes database management system (DBMS) Driver name, Server name, Database name and connection setting. The connection setting often has two methods which are trusted connection or authentication. The authentication method requires user identification (ID) and password for logging into the SQL server. The chosen connection setting is a trusted connection where user ID and password can be skipped due to simulation purpose. Furthermore, Jupyter notebook also needs to install the pypyODBC library, allowing users to connect to the SQL server. The connection is created using the “connect” method integrating with the pypyODBC library. The query statement often is the amount of data and attribute that users need for the analysis work on Jupyter Notebook. In this project, the idea is to fetch all monitor data to structure a predictive model. The whole command for fetching all induction data from the SQL server is shown in figure 44 below. This action goal is to show the possibility of accessing SQL server data for approaching predictive maintenance action.

```
import pypyodbc
import pandas as pd

cnxn = pypyodbc.connect(Driver="{SQL Server}",
                        Server="DESKTOP-6MMU3Q4\WINCCPLUSMIG2014",
                        Database="VFD_test",
                        uid='',
                        pwd='')
query = "select *from VFD_all"
df = pd.read_sql(query, cnxn)
df
```

	direction	vfd_power	vfd_voltage	frequency	vfd_current	update_time
0	0	0	0	0	0	2021-04-05 22:09:02.137
1	1	390	380	50	0.82	2021-04-05 22:10:01.723
2	0	0	0	0	0	2021-04-06 10:28:02.277
3	1	0	76.1	10	0.41	2021-04-06 10:29:01.280
4	1	830	380	50	0.83	2021-04-06 10:30:01.613
...	...	...	...	...	...	...
475	1	380	380	50	0.83	2021-04-06 10:39:01.987
476	1	380	380	50	0.83	2021-04-06 10:40:01.307
477	1	380	380	50	0.83	2021-04-06 10:41:01.627
478	1	130	190	25	0.56	2021-04-06 16:00:01.653
479	1	130	190	25	0.56	2021-04-06 16:01:01.313

480 rows × 6 columns

Figure 44. Code structure and result of fetching data from SQL server performs by Python Captured from Jupyter notebook.

However, the collected data storing on an SQL server is generated by hand, costing much time and inconvenience. The data used for doing analysis and training the predictive model in this project is created from the previous chapter. Hence, the random data generating by Python are determined as the loading data from the SQL server.

After loading data into a data frame on the Jupyter notebook, the next step for approaching predictive maintenance is pre-process data. The dataset understanding includes the number of columns, number of rows, and data types are vital for achieving the pre-process data. The reason is that each column value in the data frame is a factor contributing to predict the time to failure of the induction motor. The pre-process data step is also filtering out of the “null” value for having a clean data set. “Null” or invalid value indicates the value in a cell is missing or not existing. The existence of invalid data makes noise for the predictive model and lead to the wrong prediction for the RUL. There is no invalid data by inspecting the loaded dataset, which is appropriate for the next step, observing all column data distribution. The histogram in figure 45 shows that each column of data has a normal distribution and no need to drop any extraordinary observation. Besides histogram, boxplots or scatter are often used for observing the data distribution.

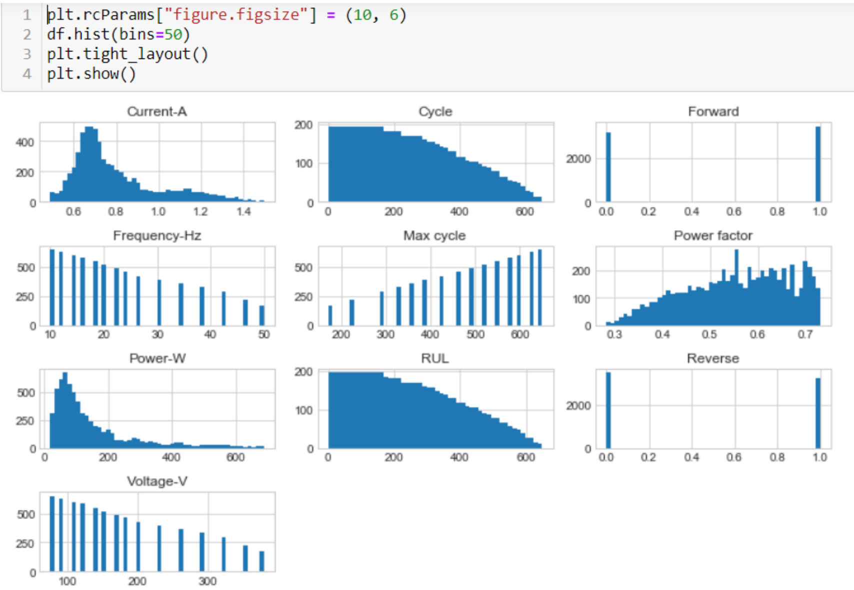

Figure 45. The distribution of all data’s column Captured from the Jupyter notebook.

In the pre-process data step, “Power factor” is likely to be a helpful variable for predicting the induction motor’s RUL. Hence, the “Power factor” column is then calculated using equation (2) and appended into the data frame. Moreover, each row in the dataset is one completed cycle of induction motor, and the maximum cycle at each frequency is the last cycle that the induction motor can operate. As a result, the induction motor's actual remaining useful life can be calculated using the maximum cycle minus the current induction motor's cycle. As a result, three more columns: "Cycle", "Max cycle", and actual "RUL", are appended to the dataset. The assumption is that the induction motor at the specific frequency is continuously observed from the healthy state to its last cycle. By concatenating these frequency's datasets into one data frame, the remaining useful life can be more confident to be predicted. The dataset after concatenation is formed and shown in figure 46 below. It should be noted that the "RUL" column in figure 46 is the actual remaining useful life. This column is the target to be predicted in the predictive model.

```
1 df
```

	Cycle	Voltage-V	Frequency-Hz	Current-A	Power-W	Forward	Reverse	Power factor	Max cycle	RUL
0	1	76.0	10	0.490521	20.090425	1	0	0.311510	650	649
1	2	76.0	10	0.490700	20.103825	1	0	0.311604	650	648
2	3	76.0	10	0.490750	20.355074	1	0	0.315466	650	647
3	4	76.0	10	0.490979	20.361133	1	0	0.315413	650	646
4	5	76.0	10	0.491170	20.372895	1	0	0.315472	650	645
...	...	...	...	...	...	...	...	...	...	...
6651	166	380.0	50	1.488821	684.738101	0	1	0.699604	170	4
6652	167	380.0	50	1.489706	686.877613	0	1	0.701373	170	3
6653	168	380.0	50	1.493598	687.302138	0	1	0.699978	170	2
6654	169	380.0	50	1.495634	688.217935	0	1	0.699956	170	1
6655	170	380.0	50	1.497664	689.192042	0	1	0.699997	170	0

6656 rows × 10 columns

Figure 46. The completed data frame use for predicting RUL after doing pre-process data.

After doing pre-process data, the condition indicators are identified by selecting column data having a solid or medium correlation to the RUL. The medium and robust correlation of column data helps to increase the accuracy in predicting the RUL. The correlation level can be identified by looking into the scatter plot in figure 47 or correlation calculation in figure 48. Figure 47 and 48 show that Power Factor, Cycle, Current and Power are selected as the condition indicators because they correlate enough with the target column, RUL.

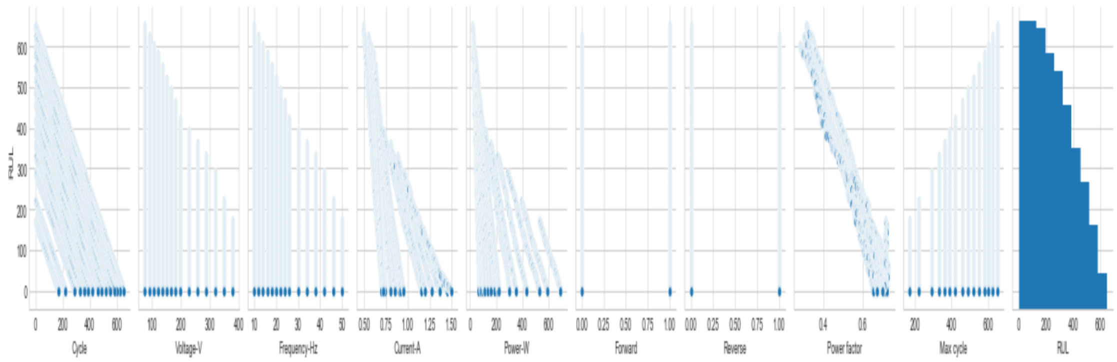

Figure 47. The scatter plot shows the correlation of RUL with other attributes Captured from the Jupyter notebook.

```
df.corr()["RUL"].abs().sort_values(ascending=False)
```

```
RUL             1.000000
Power factor    0.974766
Cycle           0.679990
Current-A       0.645102
Power-W         0.552994
Max cycle       0.400006
Voltage-V       0.397785
Frequency-Hz    0.397785
Reverse         0.041989
Forward         0.041989
Name: RUL, dtype: float64
```

Figure 48. The absolute correlation between RUL and other column data Captured from the Jupyter notebook.

3.4.3 Train Predictive Model and Result

From figure 47, the target column RUL correlate with other condition indicators in the linear regression trending. Therefore, the chosen model for predicting RUL in this project is the linear regression model. The model is a simplified function that describes the reality of data. For example, in Mathematics, the result of y in linear equation (3) depends on the x value. In equation (3) below, “b” is determined as the intercept and “a” as the coefficient of variable x. Variable x is condition indicator values of training data, and y is the outcome of the equation that relies on the value of variable x. The number of condition indicator indicates the number of inputs x and their coefficients in the linear regression’s equation. Hence, the predicted RUL value is described as output y, where its value depends on condition indicators’ value.

$$y = a * x + b \tag{3}$$

The training data action often come with testing the training data. If the predictive model first time is built, the idea is that collected data need to be split into two parts: the training and testing part. The training part often dominates the 75 per cent portion of the dataset and the remaining 25 per cent for testing. The testing part is used to compare the actual value with the predicted value and evaluate the confidence of the predictive model. Equation (4) shows the Root Mean Squared Error (RMSE) between the predicted y value and the actual y value. RMSE also can be understood as the average variance of the predicted value with the actual value. Hence, the RMSE value is usually applied for evaluating the linear regression model's accuracy.

$$RMSE = \frac{1}{n}\sum_{i=1}^{n}(\mathrm{ypre} - \mathrm{yact})^2 \quad (4)$$

In figure 49, an empty linear regression's equation is generated by the sci-kit learn's method. The training model action is performed by adding condition indicators' values into an empty linear regression equation. After fitting data into an empty equation, the model returns the intercept and four coefficients, as in figure 50. This completed equation with the intercept and coefficients is used for predicting the remaining useful life of the induction motor.

```
#Train model
from sklearn.linear_model import LinearRegression
```

```
model1 = LinearRegression()
```

```
model1.fit(X_train, y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Figure 49. LinearRegession method from the sci-kit learn library is used for training the model Captured from the Jupyter notebook.

```
#view train models
model1.intercept_
```

```
1329.1998766155696
```

```
#View coefficient
model1.coef_
```

```
array([-1.16035668e+03, -6.74958298e-02, -6.93221513e+02,  8.27557209e-01])
```

Figure 50. The intercept and coefficient of linear regression model after fitting training data Captured from Jupyter notebook.

The predicted RUL is achieved by fitting the x value of the testing part into the completed equation. Figure 51 shows the first five rows of predicted RUL comparing with the actual

RUL from the testing part. The predicted RUL value almost equal to the actual RUL. The average variation between predicted RUL and actual RUL is the RMSE of the linear regression model.

```
#Make predictions
y_pre_1 = model1.predict(X_test)
y_pre_1.tolist()[:5] #Result of prediction
```

```
[243.90127581873207,
 111.56186556408625,
 22.921453450390345,
 201.7990753937404,
 403.61399412878575]
```

```
#Compare to test data
y_test.tolist()[:5]
```

```
[223, 97, 5, 216, 386]
```

Figure 51. The first five rows of predicted RUL compare to the actual RUL Captured from Jupyter Notebook.

With the sci-kit learn library, the RMSE can be calculated easily by importing the mean square error function. The RMSE for the linear regression model in this application is acceptable, as in figure 52. If the predicted RUL is “243” cycles, then the actual RUL is varied with the deviation ± 16.348 cycles. The RUL prediction results compared to the actual RUL are fully shown in appendix 2.

```
#Evaluate model
from sklearn.metrics import mean_squared_error
```

```
#Squareroot of MSE
np.sqrt(mean_squared_error(y_test, y_pre_1))
```

```
16.34805882492846
```

Figure 52. The RMSE calculation of the linear regression model Captured from the Jupyter notebook.

Moreover, the “Alert” column is added for quickly filtering which observation has the RUL below 20 cycles. The “Alert” shows “1” for alerting the predicting observation having less than 20 cycles and “0” for observation having predicted RUL over 20 cycles. Figure 53 below is the first five-row slicing of predicted RUL, which has less than 20 cycles.

```python
result1["RUL-predict"] = y_pre_1.tolist()
result1["RUL"] = y_test.tolist()
result1["Alert"] = None
result1.loc[result1["RUL-predict"] < 20, ["Alert"]] = 1
result1.loc[result1["Alert"] != 1, ["Alert"]] = 0
result1["Alert"].value_counts()
display(result1.loc[result1["Alert"] == 1, :].head(5))
display(result1.loc[result1["Alert"] == 1, :].shape)
```

	Power factor	Cycle	Current-A	Power-W	RUL-predict	RUL	Alert
3514	0.710061	509	0.844311	157.647862	16.088073	11	1
5267	0.652499	372	1.134995	292.115512	1.899422	18	1
4459	0.712447	444	0.908754	204.300886	11.641228	16	1
1873	0.724871	599	0.739725	98.700336	16.547973	1	1
6235	0.697583	260	1.314945	506.538464	9.847323	30	1

```
(56, 7)
```

Figure 53. The RUL prediction values compare to the actual RUL with the alert for predicted RUL less than 20 cycles Captured from the Jupyter notebook.

4 Conclusion

The aim of this project was to create a predictive model to predict the remaining useful life of induction motor with minor investment available in the plant. The project helps to broaden electronics students' horizons in the automation and machine learning field. The development of predictive maintenance on induction motor is achieved by successfully estimating the induction motor's remaining useful life. It also answers that the approaching predictive maintenance on induction motor is possible with the minor investment available in the plant. Moreover, it proves that the Siemens Programmable Logical Controller is a powerful controller that controls the system and can adapt to work with other Internet of Things tools for receiving and transferring data. Besides the linear regression model, another predictive model such as random forest regression or logistic regression can improve the RSME and increase the accuracy in predicting the RUL. Ultimately, the result in this project is one alternative method for the industrial plant to perform predictive maintenance on induction motor and maximise the equipment lifetime.

References

1. R. Keith Mobley. AN INTRODUCTION TO PREDICTIVE MAINTENANCE. United States of America. Butterworth-Heinemann; 2002.

2. The MathWorks, Inc. Introduction to Predictive Maintenance with MATLAB [serial online] 2019. The MathWorks.
   URL: https://de.mathworks.com/content/dam/mathworks/ebook/predictive-maintenance-ebook-part1.pdf
   Accessed 28 September 2020.

3. Fiix Inc. Reactive maintenance [online]. Canada: Fiix Inc; 8 October 2020.
   URL: https://www.fiixsoftware.com/maintenance-strategies/reactive-maintenance/#:~:text=Reactive%20maintenance%20(also%20known%20as,to%20its%20normal%20operating%20condition.
   Accessed 8 October 2020.

4. Prasanna Kulkarni. Why Reactive Maintenance Can Be an Effective Strategy [online]. USA: Noria; 8 October 2020.
   URL: https://www.reliableplant.com/Read/31679/reactive-maintenance-strategy
   Accessed 8 October 2020.

5. MaintainX. The Ultimate Preventive Maintenance Guide: Nearly Everything You Need to Know. San Francisco: MaintainX; 6 June 2020.
   URL: https://www.getmaintainx.com/what-is-preventative-maintenance/
   Accessed 8 October 2020.

6. Ryan Chan. Why is preventive maintenance important for organisations? [online]. Canada: UpKeep; 18 April 2019.
   URL: https://www.onupkeep.com/answers/preventive-maintenance/preventive-maintenance-importance/
   Accessed 8 October 2020.

7. SEMIOTICLABS. The condition monitoring comparison guide [serial online] 2020. SEMIOTICLABS.
   URL: https://www.semioticlabs.com/resources/condition-monitoring-comparison
   Accessed 10 October 2020.

8. Joel Levitt. Complete Guide to Preventive and Predictive Maintenance. New York. Industrial Press Inc; 2011.

9. Allied Reliability. The Top 6 Benefits of Predictive Maintenance [online]. Reliabilityweb; 27 July 2020.
   URL: https://reliabilityweb.com/tips/article/the_top_6_benefits_of_predictive_maintenance/
   Accessed 13 October 2020.

10. Tobias Schroeder. The benefits of predictive maintenance [online]. Enterprise; 23 October 2017.
URL: https://blog.softexpert.com/en/the-benefits-of-predictive maintenance/ ?fbclid=IwAR3UEjmQa8dakxsxct-maoyiPQCeNM1cRMt2eYR1ZTkb4woD8HCynGGH5Tik
Accessed 13 October 2020.

11. Bryan Christiansen. A Complete Guide to Predictive Maintenance [online]. Lehi. LimbleCMMS; 13 October 2020.
URL: https://limblecmms.com/blog/predictive-maintenance/#
Accessed 13 October 2020.

12. Mathew Scarpino. Motors for Makers – A guide to Steppers, Servos, and Other Electrical Machines. United States of America: Pearson Education; 2016.

13. Augie Hand. Electric Motor Maintenance and Troubleshooting. United States of America: The McGraw-Hill Companies; 2011.

14. Dave Polka. Motor and Drives – A Practical Technology Guide. United States of America: The Instrumentation, Systems, and Automation Society; 2003.

15. Electric motor fundamentals Free online training program. Motor fundamentals [online]. ABB; 07 October 2020.
URL: https://www.youtube.com/watch?v=Jz0XO4wk5Rk
Accessed 07 October 2020.

16. Simatic. Getting started with S7-1200 [online]. Germany: Siemens; October 2009.

17. Evghenii. The basics of Siemens PLC's and programming in Simatic Step7 [online]. ISD; 16 January 2017.
URL: https://isd-soft.com/tech_blog/basics-siemens-plcs-programming-simatic-step7/

18. Peter. Ladder Logic Symbols – All PLC Ladder Diagram Symbols [online]. Learn PLC Programming Online (For Free); 28 June 2015.
URL: https://www.plcacademy.com/ladder-logic-symbols/
Accessed 24 February 2021.

19. Mitsubishi Electric. Inverter FR-E700 instruction manual. Japan: Mitsubishi Electric Corporation.

20. Siemens. Connecting a S7-1200 / S7-1500 to a SQL Database [online].
URL: https://support.industry.siemens.com/cs/document/109779336/ connecting-a-s7-1200-s7-1500-to-a-sql-database-?dti=0&lc=en-WW
Accessed 18 February 2021.

21. Simply Modbus. About MODBUS RTU [online]. 2020
URL: https://www.simplymodbus.ca/FAQ.htm
Accessed 16 March 2021.

22. Bruce Cyburt. Introduction to Modbus. Acromag; 24 February 2012.
URL: https://www.automation.com/en-us/articles/2012-1/introduction-to-modbus
Accessed 16 March 2021.

23. Ozeki 10. Modbus RTU [online]. Ozeki Ltd; 2021
URL: http://www.ozeki.hu/p_5854-ozeki-modbus-rtu.html
Accessed 12 March 2021.

24. Eamonn Walsh. RS-485 Masterclass - What is it and Why use it? Brainboxes; 27 March 2015.
URL: http://www.brainboxes.com/article/items/rs-485-masterclass-what-is-it-and-why-use-it
Accessed 15 March 2021.

25. Kevin Cope. What is a VFD? The RealPars Team; 29 October 2018.
URL: https://realpars.com/vfd/
Accessed 23 March 2021.

26. Danfoss. What is a variable frequency drive? Danfoss; 23 March 2021.
URL: https://www.danfoss.com/en/about-danfoss/our-businesses/drives/what-is-a-variable-frequency-drive/#:~:text=A%20variable%20frequency%20drive%20(VFD,during%20start%20or%20stop%2C%20respectively.
Accessed 23 March 2021.

27. Pankaj Khatri. What is Inrush Current and How to Limit it? CircuitDigest; 23 January 2019.
URL: https://circuitdigest.com/tutorial/what-is-inrush-current-and-how-to-limit-it
Accessed 23 March 2021.

28. Michael Owen. How To Save Energy With VFDs (variable frequency drives). Therma; 23 March 2021.
URL: https://www.therma.com/how-to-save-energy-with-vfds-variable-frequency-drives/#:~:text=With%20a%20VFD%2C%20the%20output,increased%20efficiency%20and%20energy%20savings.
Accessed 24 March 2021.

29. Austin Hughes, Bill Drury. Electric Motors and Drives Fundamentals, Types and Application. United Kingdom. Elsevier Ltd; 2019.

30. Matthew Scarpino. Motors for Makers: A Guide to Steppers, Servos, and Other Electrical Machines. Pearson Education; 2016.

31. ElectronicsTutorials. Full Wave Rectifier [online]. ElectronicsTutorials; 2021.
URL: https://www.electronics-tutorials.ws/diode/diode_6.html
Accessed 25 March 2021.

32. ElectronicsTutorials. Three Phase Rectification [online]. ElectronicsTutorials; 2021.
    URL: https://www.electronics-tutorials.ws/power/three-phase-rectification.html
    Accessed 25 March 2021.

33. Malcolm Barnes. Practical Variable Speed Drives and Power Electronics. Burlington. Newnes; 2003.

34. Electrical Technology. Full Bridge Inverter – Circuit, Operation, Waveforms & Uses [online]. Electrical Technology; 2021.
    URL: https://www.electricaltechnology.org/2020/09/full-bridge-inverter.html
    Accessed 25 March 2021.

35. Paul Evans. Variable Frequency Drives Explained – VFD Basics IGBT inverter [online]. Theengineeringmindset.com; 25 April 2020.
    URL: https://theengineeringmindset.com/variable-frequency-drives-explained/
    Accessed 26 March 2021.

36. Lutfi Al-Sharif. The VF Curve Used in VF Drives Simply Explained [online]. The University of Jordan; 23 February 2018.
    URL: https://www.researchgate.net/publication/
    323377197_The_VF_Curve_Used_in_VF_Drives_Simply_Explained
    Accessed 28 March 2021.
37. Electrical Academia. Three Phase Induction Motor Performance [online]. Electrical Academia; 2021.
    URL: https://electricalacademia.com/induction-motor/
    three-phase-induction-motor-performance/
    ?fbclid=IwAR3feEetnwNJ8TgVrtTvYsphmdNjfBciUiBs6yfUD1u2VSlm_BeNs-
    bmlAOU
    Accessed 29 March 2021.

The simulation data of induction motor in healthy condition

Voltage	Frequency	V/F	Current	Power factor	Efficiency	Output power of VFD	Ouput power of motor
76	10	7.6	0.58	0.3	0.1	22.903968	2.287752
83.6	11	7.6	0.585	0.305	0.104	25.83508356	2.683746094
91.2	12	7.6	0.59	0.31	0.108	28.89059136	3.116580883
98.8	13	7.6	0.595	0.315	0.112	32.07246588	3.587968238
106.4	14	7.6	0.6	0.32	0.116	35.3826816	4.099651584
114	15	7.6	0.605	0.325	0.12	38.823213	4.6534059
121.6	16	7.6	0.61	0.33	0.124	42.39603456	5.251037722
129.2	17	7.6	0.615	0.335	0.128	46.10312076	5.894385139
136.8	18	7.6	0.62	0.34	0.132	49.94644608	6.585317798
144.4	19	7.6	0.625	0.345	0.136	53.927985	7.3257369
152	20	7.6	0.63	0.35	0.14	58.049712	8.1175752
159.6	21	7.6	0.64	0.355	0.15	62.80425984	9.40976064
167.2	22	7.6	0.65	0.36	0.16	67.7641536	10.82974464
174.8	23	7.6	0.66	0.365	0.17	72.93334224	12.38435101
182.4	24	7.6	0.67	0.37	0.18	78.31577472	14.08056134
190	25	7.6	0.68	0.375	0.19	83.9154	15.925515
197.6	26	7.6	0.69	0.38	0.2	89.73616704	17.92650912
205.2	27	7.6	0.7	0.385	0.21	95.7820248	20.09099862
212.8	28	7.6	0.71	0.39	0.22	102.0569222	22.42659619
220.4	29	7.6	0.72	0.395	0.23	108.5648083	24.9410723
228	30	7.6	0.73	0.4	0.24	115.309632	27.6423552
235.6	31	7.6	0.74	0.41	0.26	123.8051613	32.15217179
243.2	32	7.6	0.76	0.42	0.28	134.4541901	37.60370074
250.8	33	7.6	0.78	0.43	0.3	145.6929302	43.65740808
258.4	34	7.6	0.8	0.44	0.32	157.5371776	50.35368448
266	35	7.6	0.82	0.45	0.34	170.002728	57.7341828
273.6	36	7.6	0.84	0.46	0.36	183.1053773	65.84181811
281.2	37	7.6	0.86	0.47	0.38	196.8609213	74.7207677
288.8	38	7.6	0.88	0.48	0.4	211.2851558	84.41647104
296.4	39	7.6	0.9	0.49	0.42	226.3938768	94.97562984
304	40	7.6	0.92	0.5	0.44	242.20288	106.446208
311.6	41	7.6	0.94	0.52	0.465	263.8010586	122.5258439
319.2	42	7.6	0.96	0.54	0.49	286.599721	140.2716995
326.8	43	7.6	0.98	0.56	0.515	310.6304589	159.789958
334.4	44	7.6	1	0.58	0.54	335.924864	181.1899584
342	45	7.6	1.02	0.6	0.565	362.514528	204.5841948
349.6	46	7.6	1.04	0.62	0.595	390.4310426	232.038218
357.2	47	7.6	1.06	0.64	0.625	419.7059994	262.013344
364.8	48	7.6	1.08	0.66	0.655	450.3709901	294.6523599
372.4	49	7.6	1.1	0.68	0.685	482.4576064	330.1018398
380	50	7.6	1.12	0.72	0.695	530.740224	368.4385152

The predictive RUL table of Induction motor

Power factor	Cycle	Current-A	Power-W	RUL-predict	RUL	Alert
0.710061264	509	0.844311245	157.6478621	16.08807291	11	1
0.652498738	372	1.134995118	292.1155118	1.899422029	18	1
0.712447389	444	0.908754374	204.3008862	11.64122828	16	1
0.724871063	599	0.739724725	98.70033557	16.54797334	1	1
0.697582956	260	1.314944983	506.5384642	9.847322838	30	1
0.650289748	363	1.127396702	289.1775792	7.906177758	27	1
0.693293741	256	1.308669623	501.0214125	14.87886404	34	1
0.655994151	384	1.151413435	297.9286097	-9.537313938	6	1
0.723489643	591	0.737302138	98.18961168	19.94761686	9	1
0.720817271	448	0.910735377	207.1516172	2.645065348	12	1
0.688555255	251	1.304947509	496.1817944	19.28986656	39	1
0.652023603	362	1.124121804	289.1063557	8.173071817	28	1
0.709638835	459	0.883857411	181.4270162	12.21738971	31	1
0.707254373	458	0.883563079	180.7571883	14.70142885	32	1
0.650194705	367	1.132183672	290.3629941	5.409046704	23	1
0.651091267	352	1.188232125	345.845216	12.44187175	8	1
0.725018455	635	0.694512201	66.2046717	18.39737015	15	1
0.668522153	304	1.24131846	414.6124214	15.56383397	26	1
0.709379164	460	0.884503363	181.4931727	12.05816557	30	1
0.723839508	459	0.918401523	209.7711775	-4.750762694	1	1
0.724826217	646	0.6983043	66.54850524	15.53375768	4	1
0.656320702	390	1.157778114	299.7245999	-13.24705251	0	1
0.702614405	267	1.321810743	512.8558485	4.005081165	23	1
0.715808732	447	0.910495191	205.6579892	7.45469284	13	1
0.709185895	507	0.843093758	157.2264664	17.73406397	13	1
0.656484486	380	1.144402475	296.3358577	-6.29423913	10	1
0.652106167	345	1.102323316	283.536028	19.72611353	45	1
0.696106388	287	1.365174074	524.7743954	-9.990328886	3	1
0.710002504	511	0.845502553	157.8572362	15.36869211	9	1
0.655917601	378	1.140968941	295.1916423	-4.068162325	12	1
0.710107335	462	0.884690127	181.7178356	11.13468833	28	1
0.708323254	436	0.904707449	202.2137128	18.04482235	24	1
0.724209292	484	0.895615974	187.6153206	-9.407046004	6	1
0.724561289	643	0.69766104	66.462901	16.41873553	7	1
0.724536052	482	0.894926121	187.5553947	-9.222583037	8	1
0.655171582	387	1.154944268	298.4674874	-10.78702511	3	1
0.696978191	257	1.309150258	503.8690355	12.55947689	33	1
0.724690426	645	0.698250937	66.53095327	15.78128694	5	1
0.651951693	358	1.195544837	348.4334961	8.111115836	2	1
0.712287838	445	0.909213657	204.3583637	11.48804962	15	1
0.648505664	346	1.181463985	342.5096983	17.77855751	14	1
0.695034693	280	1.351833989	518.8464273	-3.932408525	10	1
0.724155372	637	0.695901867	66.25817257	18.3447911	13	1
0.710381957	465	0.885224388	181.897893	10.39218821	25	1
0.653874273	377	1.139887	293.9930076	-1.871591863	13	1
0.709959399	466	0.887336376	182.2234124	9.620319884	24	1