Aalto-yliopisto
Insinööritieteiden
korkeakoulu

Hiep Nguyen

Remote Monitoring And Failure Prediction Of Guiding Elements And Diverting Pulleys In Passenger Elevators

Thesis submitted as a partial fulfilment of the requirements for the degree of Master of Science in Technology.

Espoo, 01.3.2019
Supervisor: Professor Kuosmanen Petri
Advisor: M.Sc Sorvali Ville

Aalto University
School of Engineering

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi

Diplomityön tiivistelmä

Author Hiep Hoang Nguyen

Title of thesis Remote monitoring and failure trend analysis of guiding elements and diverter pulleys in passenger elevators

Degree programme Master of Science in Mechanical Engineering

Major/minor Product Development **Code** ENG25

Thesis supervisor Professor Petri Kuosmanen

Thesis advisor(s) M.Sc. Ville Sorvali

Date 01.March.2019 **Number of pages** 45 **Language** English

Abstract

Accelerated urbanization has lead to the rising height of buildings and demand for intensive high performance of elevators in recent years. Consequently, condition monitoring has become a highly desirable capability as the complexity of elevator systems increased. The goal of this study is to develop a monitoring method for elevator components which are subjected to mechanical degradation and failures. The method is capable of indicating the current health condition, predicting future failure as well as detecting emerging issues during operation.

Studies of the fundamental principle of elements of condition monitoring such as measurement and measuring equipment, remaining useful life models laid the foundation for new method developing. Moreover, there were reviews of the implementation of health management systems in aerospace and marine industry. A prototype was built from the inductive sensor and open sources embedded system. The device has been installed in two different elevators for data acquisition. Basic data visualization and analysis models were employed for current health state assessment and failure trend prediction.

The results include validation of the condition monitoring method and prediction of time-to-failure. Arithmetic means of displacement data determined operating condition whereas the linear regression model was used to predict failure event. Moreover, while suggesting the potential usefulness of the method for system condition assessment, the analysis of the data also exposed challenges inconsistency of the measuring method, data filtering technique as well as large data size requirement.

Keywords Condition monitoring, elevator, displacement measurement, data analysis, regression, RUL, python

Preface

This thesis only reaches its end thank to the immense support by the kindest people:
Firstly, I would like to thank Ville Sorvali, without whom this thesis work would not be even possible. Ville's invaluable instructions, hints, and patience have always been imperative to the entire work.

Secondly, I would like to show my grand gratitude towards Professor Petri Kuosmanen, whose kind oversee have played major role to the quality of the work. The important part of the work would not be done without kind timely field support by Jani Uskali with the elevators and installation. Great appreciation also goes to Jouko Kinnari and Matti Mustonen of KONE Maintenance Development for extremely useful initial information and the reviews at closing stages. I am entirely grateful to Mika Alvesalo for the approval and administrative support. Also my deepest appreciation goes to members of Hoisting Mechanics Team -Miika, Tapio, Sami, Niko and, Ilari. Their coverages make this work to its end.

My grand gratitude also goes to Professor Kalevi Ekman, Electronic shop staffs – Lassi and Teemu at Design Factory whose facility and supplies present in every grain of the prototype hardware. I would like to thank my good friend Duc Pham for the kind tips with the sensors at the beginning, also big thanks go to Kristian and Timo for their battery expertise.

Other honorable supporters to the preceding and during work are Kaj and Harri from Etteplan, Tien, Hien and Tan, Jannina, the inspirational Aaltoes fellows, my cousins in the Nederlands and dear Heidi. I could not thank you all enough.

The work is also rooted in my previous educations at HAMK University of Applied Sciences and National university HoChiMinh City University of Technology, VietNam.

At the backbone of all my endeavor, perspectives and milestones I have ever acquired is my family - my parents, Aunt Bay, brother Huy, sister Duong and niece Thong, from where I departed and would always come back to.

To those who might find this work interesting and somewhat useful,
For the opportunity, joy and challenge from the work I do not take for granted,
As a reminder of the constant insufficiency of my competences,

Thank you/ Kiitos/ Xin cám ơn

Espoo, 1.3.2019

Hiep Hoang NGuyen

Table of Contents

1 Introduction

One of the greatest advancements in high-rise buildings in the last twenty years has occurred in elevator technology [1]. The elevator plays an irreplaceable role in indoor transport in today's infrastructures. Worldwide, twelve-million elevators perform seven billion trips for over one billion people every day [2]. Such intensive and demanding services make safety, reliability, and efficiency among the most important characteristics of an elevator and its sub-systems. One important metric to validate those characteristics is minimum downtime, which can only be achieved by responsive advanced maintenance operations. In turn, the key to such advanced maintenance is the ability of the system to verify the current health status of the elevator, identify anomalies and predict future failures. Condition monitoring techniques are the enabler for such capabilities.

Condition monitoring is a process of monitoring a system by studying performance parameters whose changes and patterns indicate the system's current health condition as well as impending failures [3]. In recent years, a number of studies have been conducted at various levels of detail and stages of technological readiness on condition monitoring methods. However, the application of these methods to specific industries and areas of use, such as elevator systems, remains challenging, particularly in regard to mechanical components. The present study attempts to address this need by developing a method for the condition monitoring of the traditional mechanical elements in an elevator system. Even though the operating principle and performance of these components are well defined, newly added complexity and higher performance criteria require the up-to-date features and capabilities. The study aims to develop a monitoring method whose data can be retrieved and accessed remotely. It is based on the analysis of the constituents of monitoring methods, such as measurement, embedded systems, and diagnostic and prognostic algorithms. Additionally, a brief review of the methods implemented in highly specific pioneer industries such as Aerospace and Marine is also presented in an attempt to widen the perspective and approach. The evaluation of the method includes the building, and testing of the monitoring prototype in two different elevators.

The thesis report is divided into seven chapters, beginning with the present chapter, which has introduced the background and motivation of the study and its potential challenges, objectives and scope. The literature review is located in Chapter 2. It begins by introducing the terminology of the field and its uses. The chapter then describes the formulation of a condition monitoring method, including specifications and architecture design. Next, the chapter reviews the constituent elements of measuring instruments and diagnostics and prognostics methods. The chapter ends with a review of the applications and perceived values of condition monitoring in two industries. Chapter 3 is about the development process of a condition monitoring method for a selected elevator component. The method is developed based on the component operational profile, condition data acquisition and analysis. Failure predicting model, as the major part of prognostics, is formulated here. Chapter 4 presents the design and function of the prototype that was built to validate the method. The selections of hardware and software specifications were explained. The chapter also outlines the calibration and setting procedures for running the device. Chapter 5 reports the diagnostics, which are data visualization and current state verification results. For prognostics, future failure prediction in the form of Remaining useful life in time-unit. Chapter 6 starts with the results evaluation followed by the discussion of limitations of the method as well as ideas for improvement in further development. The Conclusion summarizes the work by comparing the obtained results against the objectives stated at the beginning. Verdicts of the work to conclude the report.

1.1 Background

Industrial products and equipment are subject to deterioration regardless of how well they are designed and built. Maintenance is to mitigate the effects of such inevitable degradation while prolonging the service life of the assets. Traditional maintenance methods, such as breakdown maintenance, which occurs after the failure of the system or preventive maintenance, which sets a specific time interval, not knowing the health status of the asset, are considered to be ineffective both in terms of time and cost [4]. Especially in today's systems and equipment that are more complex while safety, quality, and reliability requirements are increased. Too short preset maintenance interval leads to extensive services whereas, too long interval causes breakdowns and emergency operations.

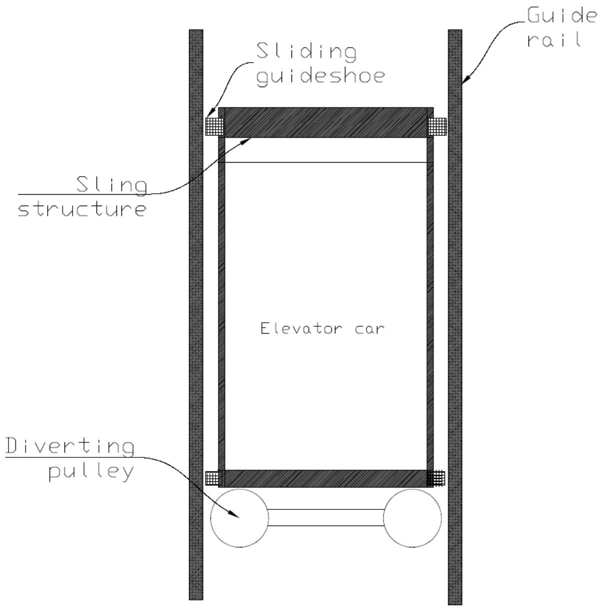


Figure 1. Simplified elevator system indicating subjected components for condition monitoring

For the elevator system, which is the vitally important mean of transportation in every building and infrastructure, maintenance operations demand high reliability and effectiveness. The typical design lifetime of elevators is relatively long, from 15 to 30 years. Additionally, failed elevator equipment or components might lead to passenger entrapment or accidents resulted in injury or death [5]. Consequently, components which are critical to elevator performance and safety, such as guiding element and diverting pulley, are subjected to most advanced maintenance techniques.

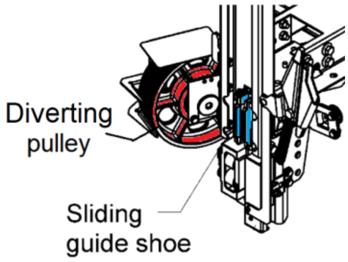


Figure 2. Location of diverting pulley and sliding guide shoe on elevator car sling

Located on car structure, guiding elements have sliding-based (or in other cases rolling-based) contact to its operational counterpart, which is the guide rail (Figure 1). Their primary tasks are to guide the elevator movement along guide rails system in the elevator shaft. Guiding elements are subject to direct friction, surface-based contact with the guide rail nose. The component is capable of force absorbing and damping. Depending on the condition, maintenance activity for the sliding-based guide can be the replacement of the pad(s) or the entire sliding assembly. It is imperative to the elevator movement, in term of smoothness and comfort, to have sliding guide components operate in acceptable condition. As a part of elevator car sling assembly, diverter pulleys, on the other hand, bear loads of the elevator car and its passengers while enabling traction from hoisting forces that actuate the car movement. The bearing assembly of the pulley makes sure the whole component operates at its intended functionality, which is the rotation of the pulley which actuates the elevator car movements. Contrary to being widely known in the industry as the most common failure in mechanical equipment [6], complete failure of bearings in diverting pulley assembly in an elevator, however, are rare according to the Company's record. Nevertheless, maintaining the robust performance of the bearing is highly desirable. Maintenance of the component usually includes complete replacement of the pulley assembly due to its essential duty and location.

One of the methods known to reduce safety hazards is to monitor and predict the elevator health condition. Successful employment of the capability would also reduce maintenance costs and operation downtime [7]. In today's market, it is perceived as a competitive advantage [8] for the manufacturers. Even though condition monitoring and health management is the widely studied and well-documented field over recent years, there is little consensus among methods and applications [9] which makes it challenging for new researchers to evaluate and derive from. Advanced mathematics and physics models are often encountered in published studies indicating much specific and complex in nature of the formulation of a reliable monitoring method. Additionally, although the basic mathematical or technical principles remain the intact, rapid development pace of the computer-aided related-tools such as machine learning algorithms or new connectivity and power consumption standards make the work prone to be outdated and out-paced.

1.2 The aim of the work

The thesis aims to design, build and test a reliable and robust condition monitoring method that is capable of verifying the current health state as well as impending failures of the subject. Due to the fact that elevator subsystems and components are diverse in functioning principles yet close in physical proximity and location, it is expected that the method(s) developed in the work would also be useful for similar use cases.

1.3 Scope and limitations

Primary target components are sliding guide shoes on elevator carsling and secondary target are bearings in diverting pulley which is also a subcomponent of carsling. Failure mechanism and root causes, degradation rate associated with performance indicators or monitoring parameters of those components should be studied and identified at early phases of the method development.

The process that forms the predictive maintenance plan consists of condition monitoring, fault detection, fault diagnostics, failure trend analysis (or fault prognostics) and decision support [10]. In prognostics, reliable prediction of Remaining useful life (RUL) is the output. There are two main principles for modeling RUL estimation, model-based and data-based approaches. The model-based method builds physical failure or degradation models such as crack, wear whereas data-based models utilize operational data from sensors to identify the status and RUL of the equipment. The thesis focuses on the data-based model, which is considered to be preferable and cost-effective [11].

It is also specified that at least one of the presented methods would be tested and analyzed via a functioning prototype. Full test and validation of the results of these method(s) are, however not in the scope of the work. Deployment and integration of the method to elevator system are also outside of the scope of the work.

1.4 Research methods

Methods used in the work, listed in Table 1-1, are based on the practical expected outcome of having a specific monitoring technique and its functional prototype. It is the combination of industry experts from the company and academic efforts via various means of communication and arrangements from meetings to laboratory works and field tests. Self-studying from diverse sources such as world-wide-web researches, books, and web-based education does contribute significantly to the success as well as the extent of the work.

Table 1-1. Research and study methods

Method	Topic	Goal	Resources	Sources
Meeting	Industry expertise	Condition monitoring existing practices Selected component design and operations	Company experts	Company materials Meeting notes
	Academic meeting	Academic practices of the thesis work and writings Academic reviews	Advisor	
Literature research	Component degradation and failure Diagnostics and Prognostic in Product health management Condition monitoring and measurement methods Remaining useful life modeling	Condition monitoring methods Prognostics/ failure trends analysis	Books, articles, papers & dissertations	World wide web
Self-educating	Electronics and embed systems	Condition monitoring device		
	Python programming	Data analysis and visualization using Basic machine learning		Web-bases education

Field test and laboratory works	Prototype constructions Data acquisition	Prototype test Data acquisition	Company experts Test elevator	Field test and notes

In order to achieve the goals, the work studies the fundamentals of a condition monitoring, its contribution in Health management system, as well as the best practices of the method in other industries. A prototype was built and deployed in a suitable elevator system. In addition to method validation, the development of the prototype and obtained results provide useful data and insights for similar design cases.

2 Literature review

Recent advancements in sensing technology and computational power have enabled real-time analysis which has set the stage for adaptive autonomous system management [12]. The key to such an advanced system is the ability to verify the current health state, identify and isolate failure sources as well as to forecast the impending events while the system is still in service. Through various development stages from theoretical research to readily commercialized package, the chapter presents studies of the capability, its constructing elements, and applications.

It is important to the work to have a consistent understanding as well as implication of the terms used in the report. Especially as widely studied as condition monitoring and health management field in which different terms might have similar interpretations and meanings. Official definitions according to European Standard EN13306 [13] are:

Table 2-1. Terminology

TERMINOLOGY	DEFINITION
MAINTENANCE	Combination of all technical, administrative and managerial actions during the life cycle of an item intended to retain it in or restore it to, a state in which it can perform the required function.
CONDITION MONITORING	activity, performed either manually or automatically, intended to measure at predetermined intervals the characteristics and parameters of the actual state of an item.
PREVENTIVE MAINTENANCE	Maintenance carried out t predetermined intervals or according to prescribed criteria and intended to reduce the probability of failure or the degradation of the function of an item.
CONDITION BASED MAINTENANCE	Preventive maintenance which includes a combination of condition monitoring and/ or inspection and/or testing, analysis and ensuing maintenance actions”. The condition monitoring and/or inspection and/or testing may be scheduled, on request or continuous.
PREDICTIVE MAINTENANCE	Condition based maintenance carried out following a forecast derived from repeated analysis or known characteristics and evaluation of the significant parameters of the degradation of the item
USEFUL LIFE	Time interval from a given instant until the instant when a limiting state is reached. The limiting state may be a function of failure rate, maintenance support requirement, physical condition, economics, age, obsolescence, changes in the user’s requirements or other relevant factors

Failure trend analysis and *prognostics* are used synonymously depending on the context and source of studies. Additionally, based on the expected outcomes within the scope of the work, *Failure trend analysis*, *Prognostics,* and *Remaining useful life* estimation are identical. In a similar manner is the case of the terms *Prognostics and Health Management* (PHM) and *System Health Management* (SHM).

2.1 System Health management

Although the work focuses on condition monitoring method at component-level capabilities, it is of highly beneficial to study how health management system specifications and requirements are formulated.

2.1.1 Requirements

Specifications of a prognostics system, according to [14, 15], are formed as the results of a process that analyzes associating factors of the system from which prognostic specifications are defined. The process is depicted in

Figure 3.

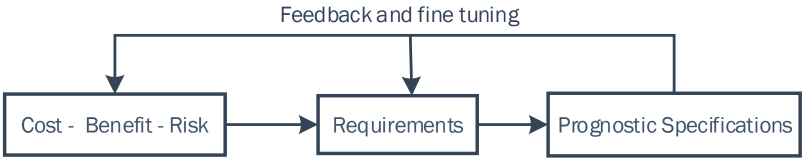


Figure 3. Prognostic specifications (re-drawn from [15])

As a starting point, Cost – Benefit – Risk analysis identifies the cost function, resources, and time constraints. Return on investment, and Cost saving estimation are two main objects in this analysis. System properties such as maintenance policy, uncertainty boundaries regarding input against output are also taken into consideration. The inputs for the Requirements stage include the specific goal set, schedules, budgets, and responsibility. This stage also defines design requirements in details, at all levels of implementation as well as the system behaviors, and interfaces for other systems. Advanced methods, and tools such as Requirement prioritization Analytic Hierarchy Process, Critical to quality tree, and Quality function deployment are needed at this stage. A major part of Prognostic specifications is the Uncertainty management methods that considerably contribute to the reliability, and accuracy of the system outputs. The other major part is the Prognostic performance attributes which are rates, and coverage of the output accuracy [16].

2.1.2 Architecture

At the center of the concept of a system, the architecture determines most of subsequent product development activities including the system integration, and deployment [17]. After requirement and specifications are defined, the design of the architecture can be conducted. It is also to be noted that condition monitoring methodology is an element of this system architectures.

The integrated condition-based maintenance

The study in [16] proposes a complete of a Health management System architecture that involves technical and non-technical aspects, as well as the decision-making process in an organization. In this architecture, the assets which is machinery or vehicle system in this context, possesses the capability to constantly provide information about their health statuses to an Information Systems from which chains of command for planning and actions are initiated. There are suitable hardware, software, and interfaces in place to enable those features.

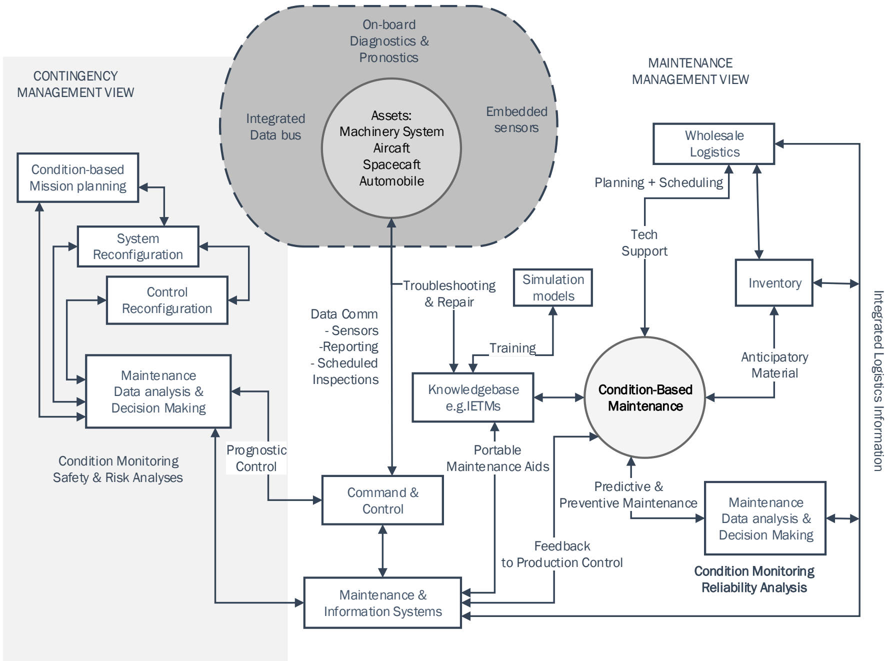


Figure 4. Health Management system architecture (re-drawn from [14, 16, 18])

The architecture regards the operation in two categories: Contingency Management and Maintenance Management. In Contingency management perspective, the goals are to increase safety, and mission (or service) reliability as well as to minimize collateral damage. As a result, service availability is improved, further failures are avoided while consumer confidence and product reputation are maintained. Maintenance management view, on the other hand, seeks to decrease logistics costs, and minimize excessive servicing. The study also proposes the service of Simulation models that are used for training the Knowledgebase from which maintenance information and maintenance activity obtain the data. Command, and Control, and Maintenance, and Information Systems are at the center of the architecture. Condition monitoring activity, which includes data analysis, and decision making, is the preceding process to Condition-based maintenance. Condition monitoring that includes Safety and Risk analysis are done in Contingency management. Although the study does introduce a robust, and thorough architect for Health management systems, there are noticeable concerns regarding the feasibility, and effectiveness of the said system. One is the fragmentations of decision-making units, and input sources. Other is the two-way connections between each of the element in the architecture. This suggests interactive, and adaptive communication which might cause delay, and inconsistency in execution.

The simplified condition-based maintenance

In a more simplified presentation compare to that of the studies in [16], and [19] , also a Condition-based maintenance (CBM) architecture, put forth in [20] proposes a unified communication network that connects the seven players. The layers are data acquisition, data processing, condition assessment, diagnostics, prognostics, decision analysis, and presentation.

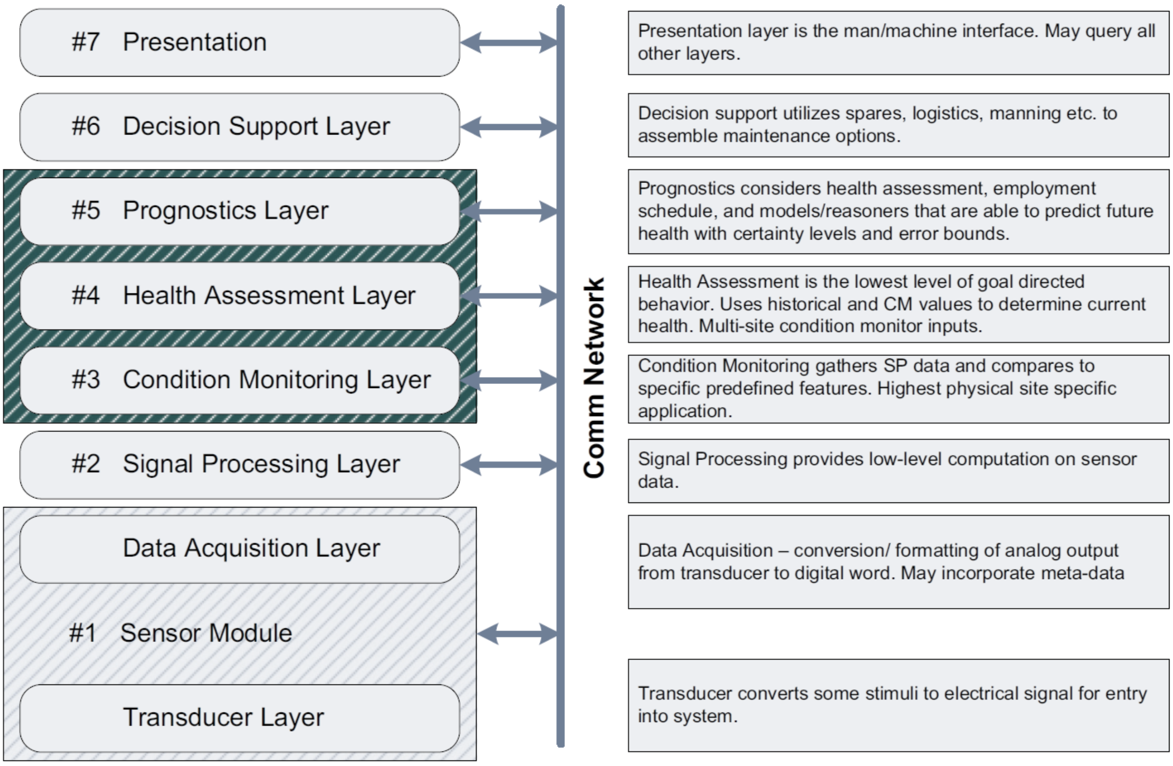


Figure 5. System architecture for Condition-base maintenance [19]

Layer one consists of data acquiring, and signal transmitting. Signal processing in layer two extracts relevant features, and signatures of the data. The core functions of this architecture lay in layer three to five. Condition monitoring makes a comparison between collected data and pre-defined threshold value; the method is expected to be capable of generating alerts in the occurrence of abnormality. Health assessment conducts diagnostics to determine current status whereas prognostics calculate the deterioration rate and estimate Remaining useful life. The architect also mentions the service of historical data in assessing health status as well as predicting future events. Decision support layer takes into consideration all relevant factors such as spare, and logistics, resources, and maintenance operations. At last layer, the presentation of information, and recommendation are displayed via User interfaces and interactions. The study also acknowledges challenges in this architecture. One is the high initial cost, and the other is the complexity of machinery structure as well as the environment in which the system operates. One of the solutions to the challenges is the integration of CBM, and data fusion technique. Data fusion is the process of maximizing the usefulness of the available data by combining data, and knowledge from multiple sources. In an integrated system, data fusion is employed in diagnostics, and prognostics subsystem. Fusing multiple degradation indicators could provide a more reliable result according to the study. The enhanced architecture is depicted in Figure 6.

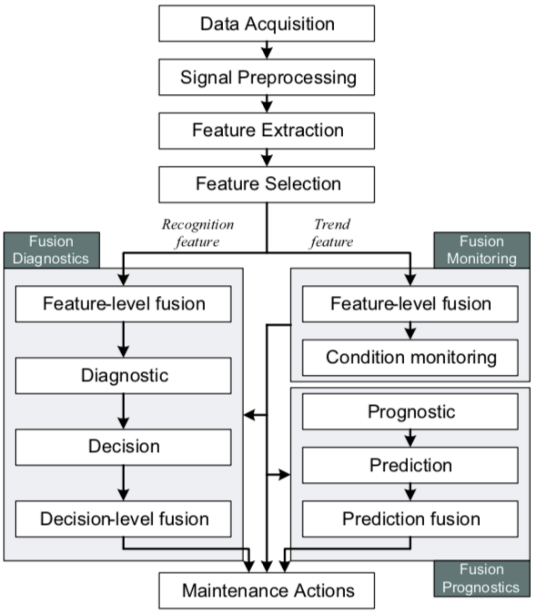


Figure 6. Data fusion integrated CBM [16]

Optimal maintenance actions are believed to be achieved based on the results of recognition of current status, and future trend via integrated processes of monitoring, diagnostics, and prognostics. The fusion technique requires the employment of advanced methods such as neural networks, vectorizations, and statistical learning models, yet its results in the experiment in [16] showed only the approximations to the measured outcome, and that further improvements of accuracy and precision is needed.

2.2 Elements of the condition monitoring method

The section delves into details of the construction of a condition monitoring method. There are two major elements, one is the data acquisition, and the other is the methodologies that are used to utilize the data. In data acquisition, relevant measurement, and measuring device are reviewed. The utilization of data requires diagnostic and prognostic method.

2.2.1 Monitoring instrumentations

Condition data is the primary input of a monitoring method. The acquisition of such data requires suitable selection and deployment of the measuring equipment. The process starts by determining critical components to be monitored. This task requires sufficient analysis, and expertise that involves system, and subsystems decomposition, technical descriptions, and dependability analysis [20]. The output of this process is the physical parameters that are subjected to monitoring actions. The operation profile and degradation rate of the component are also major contributing factors of the process. Data acquisition itself is the consequential task. Sensors or transducers, which convert physical phenomena into an electrical signal [21], are used to record information of parameters. There are various technical decisions to be made in the process such as sensor type, data storing, and data pre-processing. Next phase is data processing and diagnostics in which component condition is assessed, and verified. Features extraction, raw signals conversion, health indicator construction, current are key activities that have the health status verification is the output. Prognostics involves degradation characterization, remaining useful life, and failure trend analysis. Predicting algorithms, and models that can produce quantitative prediction values are the output.

Measurement methods

Measurement is defined as the process of gathering information from physical phenomena, and comparing this information with agreed standards [22]. Physical phenomena are movement, electrical signals, radiant energy, thermal, magnetic or mechanical energy [21]. Information is typically recorded by sensing equipment in the form of an electric signal which then being conditioned,, and stored or displayed.

The accuracy of a measurement is calculated from all accumulated errors. Error sources can be the sensor itself (e.g. thermal drift, linearity errors), or part of the measuring system such as fixturing, ambient temperature change. They can be divided into two groups, ones that occur during measuring, and the others arise afterward (such as transform, and transmission of data). There are 2 groups of measuring errors, systematic errors, and random errors. Systematic errors are incorporated in the measurement output readings. They are consistently on one side of the correct reading (either positive or negative). Two dominant sources of systematic errors are the effects of environmental changes, and instrument setting or functionality imperfections, both happen at the time of measurement. Random errors, also known as precision errors, on the other hand, caused by random unpredictable factors. As far as systematic errors are concerned, measuring equipment calibration is vitally important. Efficient, and effective calibration procedure instructions and records are needed to ensure reliable measurement results [23]

Displacement measurement

There are two types of displacement, translational, and rotational. There are more than 10 types of transducers, and methods can be used to measure displacement[24]. Widely used transducers for small, and medium movements are resistive potentiometer, linear, capacitive sensor, inductive sensor, strain gauge, and a piezoelectric transducer. The main sensor selection criteria are the magnitude of the displacement, and operating environment (including material of the measuring object). At inaccuracy level of ±2%, the piezo accelerometer is the most common type of transducer used.

Vibration measurement

Dynamic response to operational forces is one of the typical outputs of vibration measurement. The results then can be used for optimizing operating conditions or monitoring the health condition of a structure or system [24]. For machinery diagnostics, continuous monitoring of the vibration is to identify the natural frequencies of the structure. Increase in vibration signal indicates part wear, eccentric, faulty bearings or gears, or loose fixtures. At high frequencies, the best measuring parameter is acceleration while displacement, and velocity are the ones in low-frequency motions. The output impedance of most vibration transducers needs signal conditioner to amplify the signals [25]. Transducer, and conditioner also need proper calibration in magnitude, and phase over the measuring frequent range. The conditioned signal then will go through analyzers which....

Vibration transducers can be contacting (e.g., for seismic measurement) or non-contacting (e.g., interferometric, optical or capacitive). Important specifications for selecting vibration transducers are sensitivity, frequency limitations, bandwidth, noise, and amplitude linearity. For contacting transducers, mechanical mounting methods, the attachment position, and cabling are critical for its performance[26].

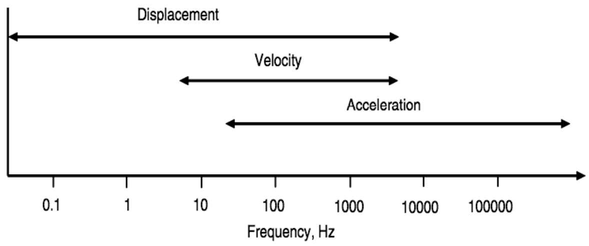


Figure 7.Vibration measurement based on frequency range [27]

As a generic guide, [27] suggests that the parameter which is used for vibration measurement is variable depending on the frequency of the vibration. For low-frequency vibration, displacement or velocity is the most suitable subject whereas it is the acceleration for high-frequency vibration. There is also a comprehensive practical guide for the selecting sensor type based on the application, shown in the following table:

Table 2-2 Area application of transducer types [27]

Application	Transducer type
Motor/pump drives	Velocity or acceleration
Motor/fan drives	Displacement or velocity
Motors connected to gearboxes (rolling element bearings)	Acceleration
Motor with oil film bearings	Displacement
Generators steam turbines	Displacement

Measuring device

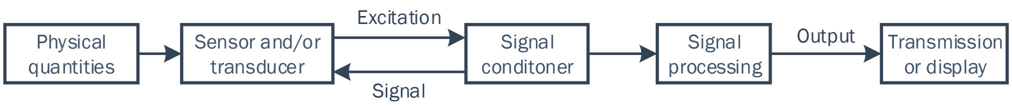


Figure 8. Measuring instrument construction (re-drawn from [22])

According to [22], a typical measuring device possesses some or all of the five stages shown in Figure 8. The behavior of a physical variable is recorded by a sensing unit in the form of electric signal. Depending on the architecture of the device, an electric signal can either be converted to digital for transmission, display or storage. It is also can be noticed that the stage-based construction of the measuring device effectively supports identification, and classification of the source of errors. The design of a measuring device is a process that consists of three major tasks: measurement descriptions, concept generation, and result evaluation. The descriptions enlists the information of device's performance such as range, bandwidth, accuracy, and sensitivity, technical details such as dimensions, power consumption, operation conditions. Additionally, business aspects such as development time, design lifetime, and costs are to be addressed at this phase. Concept generation, as the most creative part of the process, started by weighing factor characterizing the requirements. The architecture of the device specifies internal as well as external elements including installation, and User interfaces. Iterations and design cycles are part of the process. Evaluation involves assessment of the performance and measurement results.

2.2.2 Diagnostics and Prognostics

The key success factor of a condition monitoring method is the accurate diagnostics of the current health status, and the ability to reliably predict future failure events [28]. This section examines the appropriate methods, and tools for those purposes.

Matlab Predictive maintenance toolbox

With Release R2018a Matlab of MathWorks introduces complete toolbox of building Remaining Useful Life models for a Predictive maintenance program. Condition monitoring and prognostics algorithms are implemented in a Predictive maintenance system to analyze collected data then predict the next failures of target asset [29]. Condition monitoring detects and diagnoses faults by using measured data. Condition monitoring algorithm derives condition indicators which represent degradation behavior. Prognostics forecasts when will the asset fail base on diagnostic results. A prognostic algorithm gives an estimation of Remaining Useful Life or time-to-failure by using simulated models or machine learning or in some cases the combination of the two techniques.

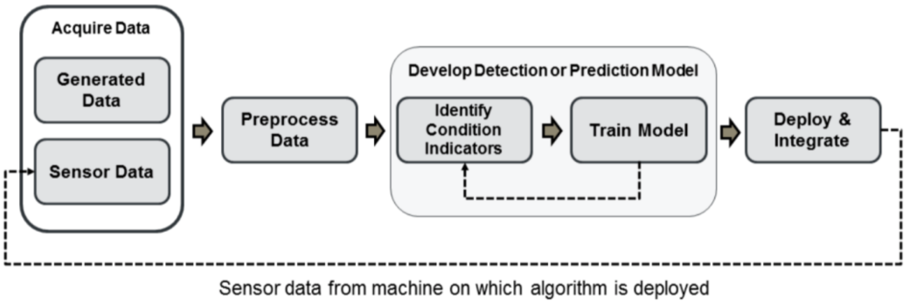


Figure 9 Workflow for algorithm development [29]

The process starts with data acquisition then collected data go through a certain conversion, transformation or basic data cleaning techniques which will make signature behavior or condition indicators to be extracted easily, and accurately. The appropriate statistical analysis combines with asset knowledge help identify condition indicators. This step is iterative which takes into consideration of uncertainty as well as trial and error. Diagnostic and prognostics are then implemented by testing different models and combinations that include also Statistic and Machine learning toolbox. RUL estimation is the result of this step. The last action of the development is the deployment of the algorithm by integrating it into the Information Technology system. As the most important role of prognostics, RUL model estimates remaining useful life based on statistical properties of condition indicator. Remain life unit depends on an observed parameter which can be in distance, volume or time. The Toolbox offers 3 main model families: similarity, degradation, and survival models. Similarities models use historical knowledge to apply to the current similar system. This approach is useful when data of systems available. Built-in functions are ready for all three types of similarities models.

Degradation models extrapolate past behavior to predict future trend. They are most useful when accurate condition indicators and their values are known. Survival method uses a statistical method to model the failure trend, useful when complete asset health management data is not known (for example only asset lifespan is known whereas how it fail is yet to observe). There are built-in functions for every model mentioned in the Toolbox, an example of the syntax is shown in Appendix B.

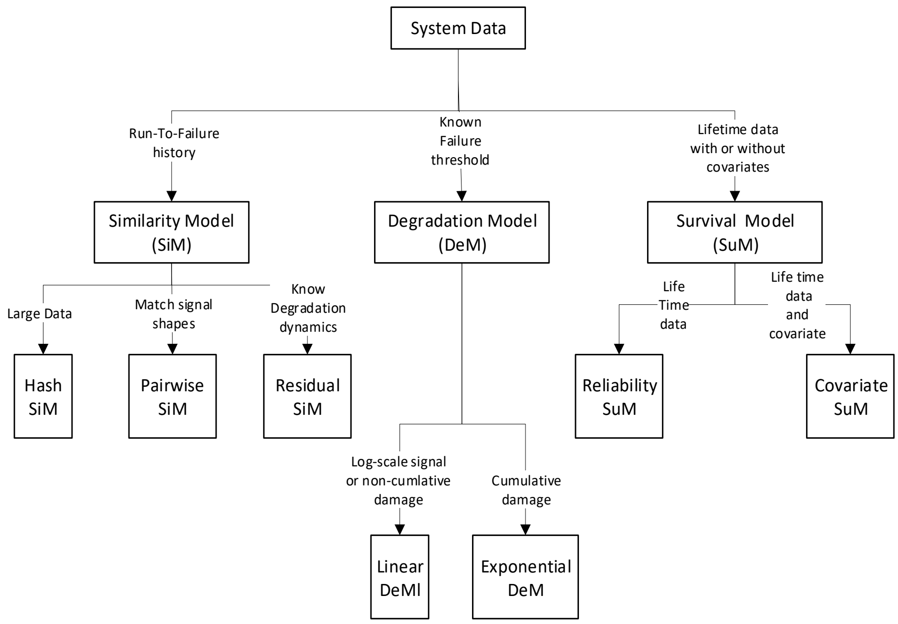


Figure 10 RUL modeling (re-drawn from [30])

The four classifying models

According to ISO-13381 standard, the implementation of the diagnostics and prognostics models following, consists of three levels: current failure mode prognostics, future failure mode prognostics and post-action prognostics preceded by diagnostics process that involves fault detection, isolation, and identification. As the key element of a prognostic process, Remaining useful life (RUL) prediction models are introduced and evaluated by the paper [9].

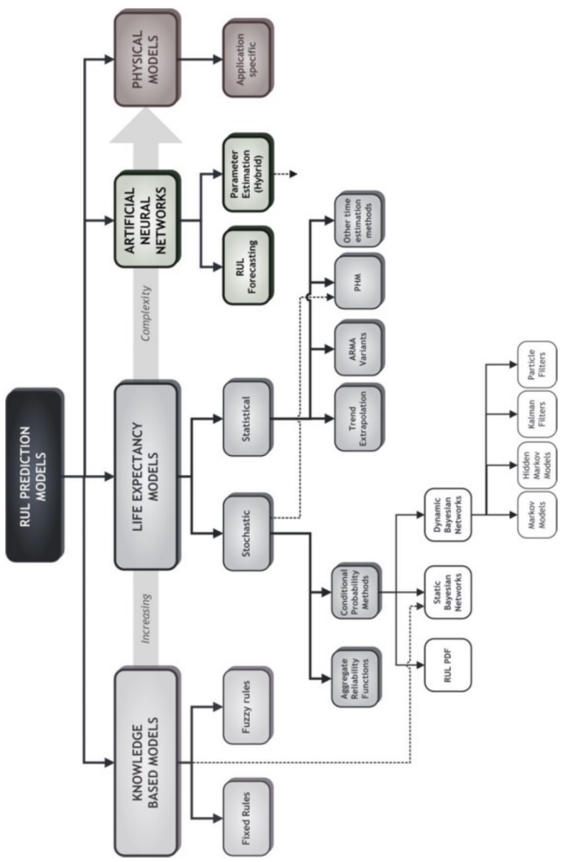


Figure 11 RUL model categories [9]

a) Knowledge-based model
RUL prediction is deduced by assessing the similarity between observed data and historical recorded failures. The model is simple to develop, easy to understand. However, it requires high quality data and a thorough understanding of the system based on historical data. In practice, the model is used to supplement other models' results rather than a standalone solution.
b) Life expectancy model
Life expectancy model calculates RUL from acquired deterioration rate under known operating conditions. The model employs well-established and widely accepted statistical methods such as probability, static Bayesian networks.
c) Artificial Neural Networks
Artificial Neural Networks estimates RUL of the asset directly or indirectly from data without the implication of known failure modes. There are many types of data can be used as input including condition monitoring indicators, operation characteristics and maintenance features (e.g. records of maintenance visits)
d) Physical model
Physical model (also known as behavioral model) uses physical laws to quantitatively characterize behaviors of deterioration processes. The model is application specific

The study also highlighted the effects of business factors on the model outputs, as well as the high the cost of data acquisition, IT infrastructure, and personnel training. An organization should implement a staged approach that is to mature its existing diagnostic programs prior to progress to more advanced models. Trend extrapolation is the most widely used technique since Artificial neural networks are most popular in academic applications.

The imperial knowledge-based

A health state estimation process consists of three subsystems – historical knowledge, diagnostic, and prognostic [31]. In this process, a diagnostic model is based on empirical knowledge of degradation and failure of the asset. Output of this model provides essential information for prognostics. The architecture requires a large amount of historical knowledge about recorded failure modes as well as the asset's service life. Condition monitoring elements, such as sensor data, signal processing method as well as feature extraction from the condition data are linked with historical data. Whereas Prognostics mathematical model of the prognostics combines imperial data and diagnostic results to estimate and predict future events.

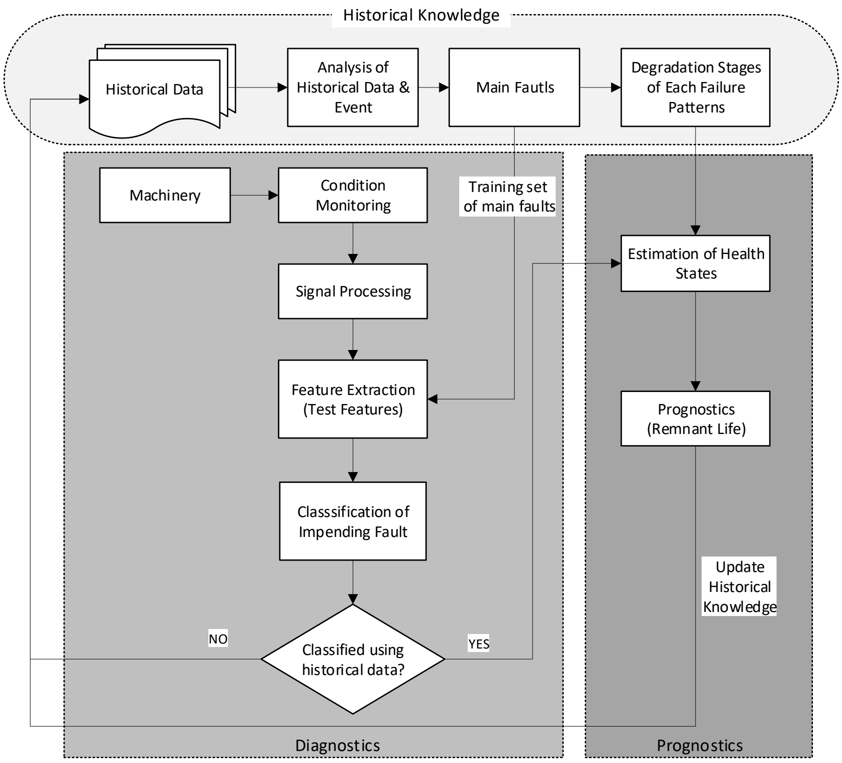


Figure 12 Diagnostic and Prognostic System Based on Health State Estimation (redrawn from [31])

Linear regression model

Linear models are simple and effective in describing the relation between inputs and output, according to [32]. They are also suitable in circumstances where training data is limited, low signal-to-noise ratio. In many instances, prediction results from linear models are more reliable compare to nonlinear one's. of the output predicted values Y that has Input vector X^T:

$$X^T = (X_1, X_2, \ldots, X_p)$$

The linear regression model *f(X)*:

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$$

β_j: coefficients

Regression function:

$$E(Y|X)$$

The output, predicted Y:

$$Y = \beta_0 + \sum_{j=1}^{p} X_j \beta_j + \varepsilon$$

ε: the error

2.3 Condition monitoring in Industries

Considered to be an interdisciplinary field, (structural or system) health management and its implementation requires integration of material science, mechanics, electronics, and computer science. This means a dedicated approach is needed throughout the system lifecycle from design to manufacturing, operation, and maintenance [33]. An effective health management system is capable of detect, isolate, and predict performance and degradation, at a sufficient level of details, without interrupting system operation [34]. It enables system intended performance at minimum cost throughout the entire lifecycle. Main stages of Health management process are data acquisition, diagnostics, prognostics, and health management. In data acquisition and diagnostics, a thorough understanding of the operation and failure modes of system or machine is required. This stage involves the formulation of performance indicators, monitoring parameters as well as the development of method and equipment. Relevant statistical analysis and modeling, advanced predictive algorithm development based on the available data are among the keys activities in prognostics and health management phase.

Challenges

As in any advanced and emerging technology development, Health management encounters numerous challenges, some of which are well addressed with promising results while others remain to be the subjects for future studies. Firstly, it is the sensor selection and localization. Integrity and accuracy of acquired data are the first conditions of any reliability requirement. This requires sensible sensor selections and optimization of their locations in the system. In a complex system, optimization is a substantial challenge that expects complete knowledge of the design and operations of the entire system [35]. Secondly, analysis of collected data requires a various set of techniques including signal transformation, feature extraction based on performance characteristics, data visualization and last but not least noise and error compensation. The third challenge is prognostics that involves Remaining useful life (RUL) prediction model which requires an adequate understanding of failure mechanisms, degradation modes of the asset.

In relation to uncertainty sources in a condition monitoring system, there are three main groups. One is related to prognostic modeling in which oversimplification and/or incorrect selection of indicating parameters lead to inaccurate results. Another potential source is the measuring environment where unaccountable operational events might occur. Lastly, deficiencies of the subject being monitored, for example, geometry, material, production quality should be taken into account.

Benefits

In spite of the mentioned challenges that would result in the increase of time and cost for design and development, benefits from a functional health management system is believed to be significant in a variety of categories:

- Life-cycle: Machine and system which is equipped with health monitoring capability will perform safely and reliably at minimum downtime and unscheduled breakdowns. As a result, operation and maintenance cost is minimized while the intended service is optimized. Higher reliability and performance rate equivalent to a higher perceived value which leads to increases in sales and revenues.
- System design and development: data and information which is used for and provided by prognostics would also help new design cost and performance improvement. A systematic and strategic decision about the life cycle, system features and architecture will also be enabled thanks to complete knowledge built from available data. management strategy... logistics, supply planning, and support.
- Production: Quality control in product and logistics would now have thorough safety, performance -critical criteria from raw material supply to finished products.

2.3.1 Aviation and space

Forerunner in research and implementation of new cutting-edge technology is space and aviation industry. Modern aircraft is equipped with Aircraft health management

(AHM) systems which enable fleet-wide monitoring of onboard systems and components. AHM systems analyze condition data, identify potential failures, and issue service related information that can also be shared to ground stations [36]. The accumulated health and service data are used in predictive analytics and prognostics model that help improve inspection and maintenance in improving operational efficiency and reduction of costs and risks. Being as highly regulated as aircraft operation, maintenance policies are licensed by airworthiness authorities (i.e. European Aviation Safety Agency EASA). Maintenance programs, on the other hand, are built to optimize the technical total cost of service due to two factors: maintenance costs (such as labor, spare parts purchase, logistics) and aircraft downtimes [37]. Due to its complexity, an aircraft needs specialized software for maintenance operations. For each critical component, there is a failure process model that calculates multiple functions such as reliability, maintainability, projects failure process. Entire aircraft model can also be built by combining component's models taking into account reliability theory and pattern serial characteristic (that is a failure of one component causes the grounding of the aircraft). It is also possible to simulate different scenarios characterized by different maintenance policies, failures, reparation and cost parameters.

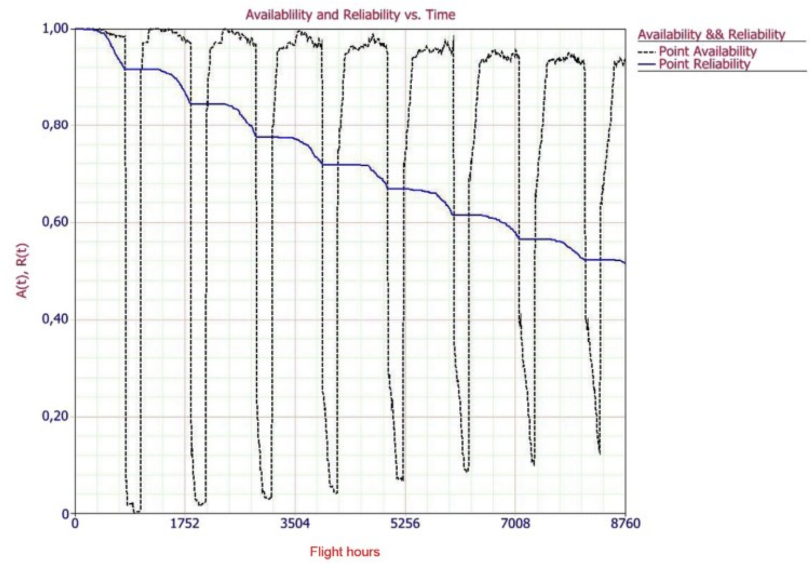


Figure 13. Aircraft maintenance modeling [37]

Compare to the previous implementation, this system reduced 20% of total annual cost of the Airbus A320 family in an Italian carrier, according to the article. Whereas according to [38], the health management system help reduce the cost by 40%, 30% to 10% electrical components, and specifically for Boeing 777 is 50-80% maintenance cost. It is, however, worthwhile to notice that such implementation of lowering overall system life cycle costs would come at the expense of higher design and production costs

Although at even higher level of complexity, health monitoring and management systems on Spacecraft, called FDIR (Fault Detection, Isolation and Recovery) whose role is to ensure the success of spacecraft's mission objectives and constraints [39], has similar operation concept of acquiring and analyzing data from onboard sensors to evaluate the health status. The (American) National Aeronautics and Space Administration (NASA) proposed Integrated vehicle health management (IVHM) and integrated systems health management (ISHM) which can extend the health monitoring concept beyond structures to include the entire vehicle and its missions. Due to the extreme environment in which spacecraft operates, additional model namely Simulated-based systems engineering, which simulates the entire spacecraft lifetime, is applied to the spacecraft detailed model The model is called Airframe digital twin. Full detailed information of the entire life cycle of the vehicle as results of the simulation improve significantly the safety and reliability of the spacecraft[40]. It is worth noticing that many of the spacecraft design and implementations are not applicable to elevator systems (such as radiation exposure, power consumption criticality), however, a successful adaptation of some of the advanced models in the aerospace industry will make a remarkable competitive advantage in the vertical transport industry. Next generation of technology implementation in a vehicle, also proposed at NASA, is the self-aware spacecraft, aircraft which has situational awareness capability of the external environment as well as its internal conditions configurations. There are 3 main subsystems, adaptive mission management which is central to the awareness capability, decision-making under uncertainty and autonomous guidance, navigation, and control. Considered to the be key of a self-aware system, the intelligent subsystem utilizes real-time performance data and prognostics health monitoring to optimize the vehicle performance. The forth mentioned digital twin model plays an important role in facilitating maintenance efficiency and quantified reliability, according to the article.

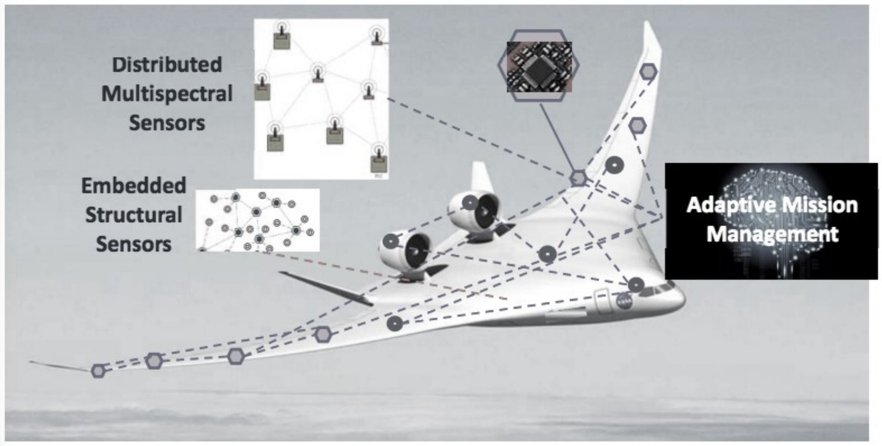


Figure 14. The concept of self-aware vehicle [12]

The employment of systematic selection methods for sensors and actuators has enabled distributed control architectures, replacing existing analog control systems, that powers advanced capabilities such as Thrust Vector control Diagnostic Model. The model is capable of providing qualitative failure effect propagation paths across system physics along with value recommended change information to have the least impact to cost and schedule during system design and development phase [41].

2.3.2 Marine

Another highly specified and regulated industry is maritime whose typical performance contributing factors are varied from crew member expertise, sea weather and climate to fuel cost and delivery schedule, they are illustrated in Figure 15:

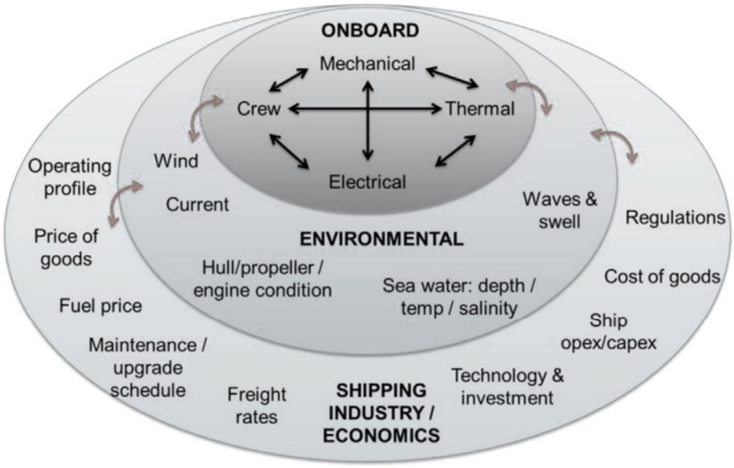


Figure 15. Ship performance contributors [42]

Maritime transport makes 80% by volume and more than 70% by value of the global trade [43] while maintenance accounts for 20% -30% of a ship's operational cost [44], the importance and size of the industry is one of the most significant. Maintenance in the marine industry nowadays is moving from predominant time-based, prescriptive approaches to Condition-based maintenance method. Proposed in the paper is the solution combining Fault Tree Analysis (FTA), FMEA and Artifical Neural Networks (ANN) application to issue maintenance information. FTA, whose basic construction are logic gates and events, is a top-down approach which deduces fault sources from failures. ANN, resembling the data storing and learning of the brain, is the network of parallel distributed processor formed by processing units that have accessible information (experimental knowledge). Being trained to learn from past events then provide functional relationships or forecast future events, ANN is considered to be a highly effective tool for analyzing and diagnosing of nonlinear behavior of complex systems as well as performance evaluating of operators and decision-makers. The paper also introduces a case study in which one of the most important components of a ship, the main engine (MAN B&W 8k90MC-C) is subjected to this monitoring methodology. Total of 39 basic events from various subcomponent groups was studied. Exhaust gas temperatures from each of the eight-cylinder engine is one of the data acquisition targets which then is the input for ANN. The result, shown in Figure 16, is on hourly prediction basic (30 hourly readings to predict the next 10 hourly values). Often being offered as service packages in the maritime industry, monitoring sensor system can have design lifetime up to 30 years and could be easily retrofitted or upgraded [45].

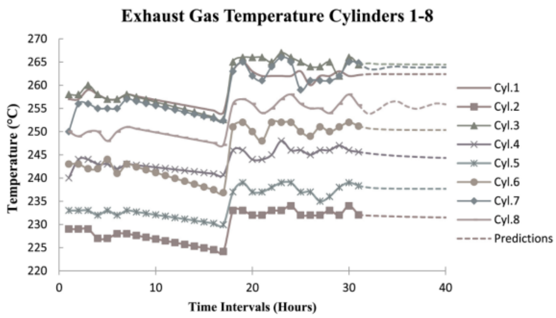


Figure 16. Ship main engine exhaust gas temperature prediction [45]

3 Condition monitoring method for guiding elements

In buildings, a typical elevator system is located in a dedicated shaft in which guide rails are installed in pair, one rail opposite to the other on the shaft wall. The vertical movement of the elevator car is enabled by the guiding elements located on the carsling. There are two typical guiding elements, the sliding guide and roller guide. Sliding guide is mostly used for lower speed, up to 2.5 m/s whereas the roller guide is most suitable for the speed from higher than 2 m/s. The scope of the work is limited to sliding guide.

A sliding guide is designed to perform under certain amount of loads while maintaining good sliding contact and damping. The Sliding guide is critical to the smoothness of the travel of the elevator, characterized as comfort class [46]. In the construction of a sliding guide, there are inserts that are subjected to surface contact and impact [47].These inserts wear and degraded over time and they are the subject of condition monitoring method developed in this work.

3.1 Operation profile of the sliding guide

Sliding guide, referred as sliding guide shoes in the industry, is used in an elevator that has the nominal travel speed up to 2.5 m/s. Available materials for insert (grey part in Figure 17 are polyethylene and polyurethane for lubricated guide rail and Polyamide for non-lubricated rail, according to Company's Product description document SO-07.07.011.

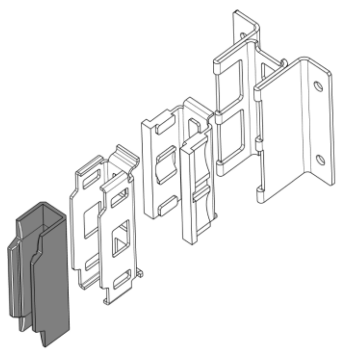


Figure 17. Elements of a sliding guide shoe

The maximum running force applied to the guide shoe components ranges from 320 to 20000 N in one direction. Maximum impact or seismic event forces are typically twice as much as running forces. Also according to the same document, the generic lifetime of the guide shoe component is two years in standard operation (1000 N load, 2 m/s speed, 75 m height travel). Average emitted noise is 67.8 dB, at 2 m/s speed, under 300 N guide force and measure 20 cm away from the contact area. According to Company's Preventive maintenance instruction document (ASG-07.04.034, Appendix J.), allowable gaps between guide rail surfaces to guide shoe pads are 2mm. Maintenance activities are required when 2mm is exceeded. For direction X, a typical solution is sliding insert replacement and shim plates adjustment for direction Y's.

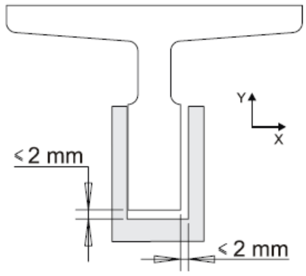


Figure 18. Allowable gaps for sliding guide shoe

3.2 Condition monitoring method

The development process of the condition monitoring method is the analysis of failure modes and root causes in a sliding guide shoe. Depicted in Figure 19, there are two categories one that is the internal degradation of the component due to use, the operation failures and one that is the external elements from the environment in which the component operates, the random failures.

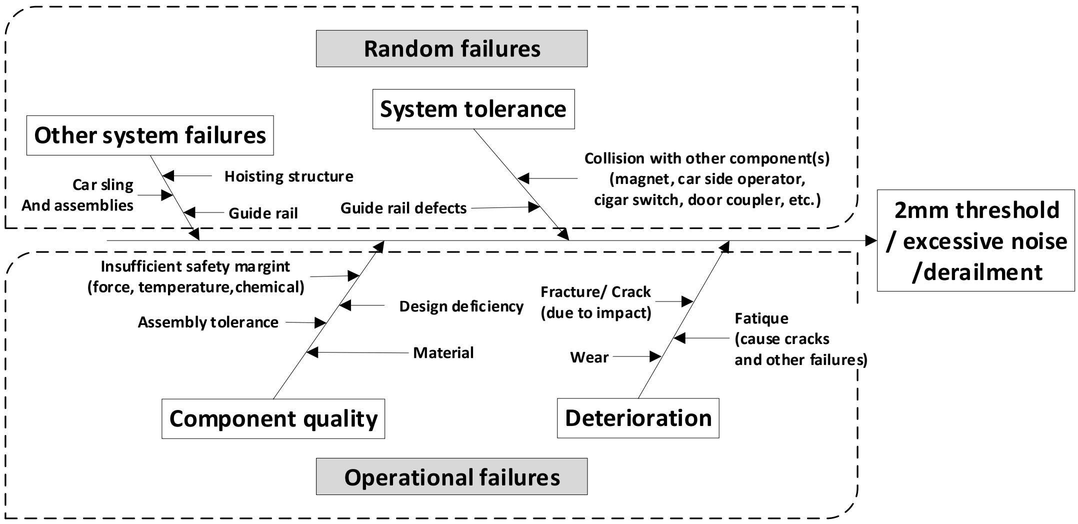


Figure 19. Failure modes of sliding guide shoe

Based on the understanding of the failure modes analysis, the model for health management, consequently the condition monitoring method of the sliding guide is data-based. The condition indicator is the allowable gaps at the threshold of 2 mm. As a result, the health condition is determined solely by the analysis of this gap parameter. illustrated in Figure 20 visualizes the health management architecture in the form of stages of managing the condition as well as the information flow. This architect is designed to be independent with the renewal or replacement that might occur during maintenance of the sliding guide shoes.

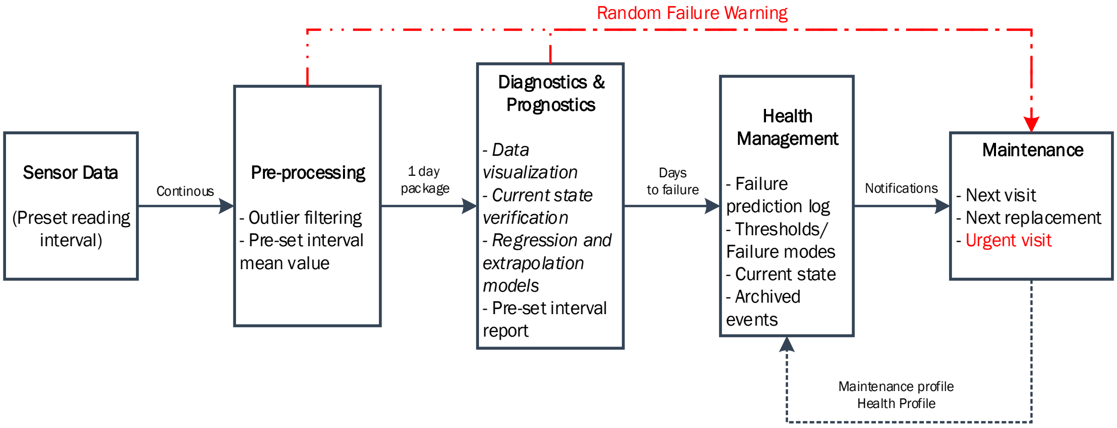


Figure 20. Health management architecture

In this architecture, data recorded by the sensor is processed before feeding to diagnostics and prognostics models whose analysis results are the current state and prediction values. One of the most important roles of the health management stage is the maintenance activities issuing. It also acts as the knowledge center of the component containing operation profile, event log, component archives. This architect also proposes an alarm trigger mechanism that should be employed at early stages in the event of emergency or random failure occurs. The condition monitoring method for sliding guide shoe is developed based on the Health management architecture whose specifications are considered to be the design requirements of the method. Figure 21 depicts the components of the architecture of the method by categorizing them into areas of function and discipline.

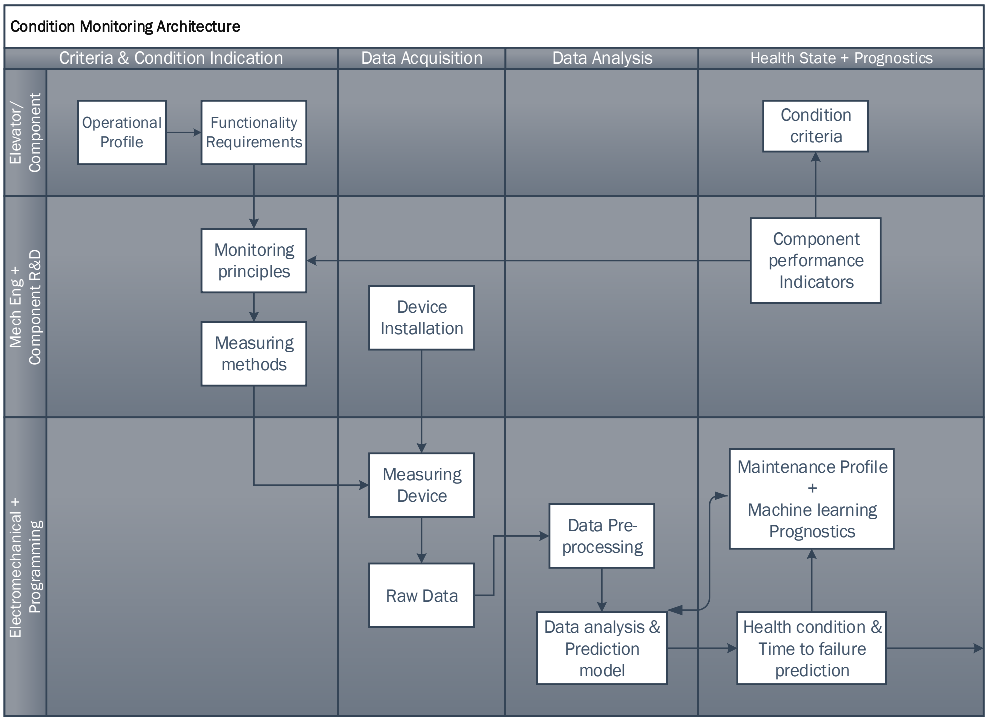


Figure 21. Condition monitoring architecture

The chart greatly supports the information traceability as well as the distribution of key elements and workload. For example, specifications about the component are essential at the beginning and prior to final results whereas computer programming or software plays a significant role in the functionality and performance of the method.

3.3 Failure prediction

At the core of predictive maintenance is RUL prediction whose results a maintenance plan is built upon. As for sliding guide shoe, gap parameter is now identified as the displacement value, is selected for monitoring activity. Statistical model namely linear regression is employed to the collected data to predict the event in which displacement value reaches 2 mm threshold.

Regression analysis is a statistical technique which formulates the relationship between a dependent (target) and one or more independent (predictor) variables. The analysis predicts the time-series model or finds the cause-effect relationship among variables through linear combinations [48]. Being a useful and widely used method [49], Linear regression model: [50] and [51]

$$\hat{y}(\omega, x) = \omega_0 + \omega_1 x_1 + \cdots + \omega_p x_p + \varepsilon \tag{3-1}$$

x_i: predictor
y: target variable
ω_i: coefficients
ω_o: intercept
ε: observation noise/ random error

From proactively identifies fail car parts to predicts the company's annual revenue, regression models are widely useful across industries and applications [52]. There are four steps to formulate the model:

1. Feature engineering and model selection
2. Model training
3. Model finalizing and validating
4. Applying the trained model to new data

In the case of guide shoe and guide rail displacement data, here is the process of building the RUL algorithm based on Linear regression predict model

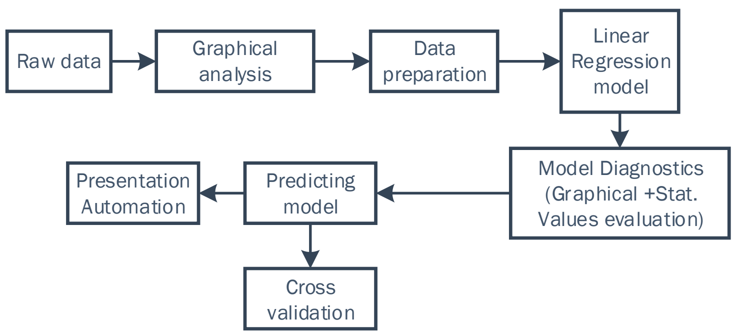


Figure 22. RUL estimation process

Data collected from the sensors will be manually converted into distance values, additional outlier elimination and other data cleaning techniques are recommended before starting analyzing. For simplicity and thanks to reasonably reliable results obtained from laboratory tests and calibration, the signal transformation was not done in the scope of the work. The graphical analysis consists of plotting the entire data set (usually scatter and/or density plot for displacement measurement) in order to initially understand the data's characteristic, generic operation properties, as well as reading abnormality, can so be identified in this step. Based on the understanding of the previous step, filtering and other generalizing methods are applied to prepare the data for the most accurate representing of the data, at the same time minimizing inherited errors. For displacement data set, subsets of maximum and minimum values during a certain timeframe (duration) are considered to be most appropriate representatives for continuous sensor reading.

Scikit-learn library in Python provides ready Linear regression method which then is applied to the prepared data. It is crucially important in this step to visualize the regression line to evaluate the model fitting result. Statistic summary, error values (R-squared, Mean squared, etc.) are computed and to be assessed at this step to determine whether the learned model is sufficiently accurate for predicting new values. There are also data split and trial and error activities during applying and evaluating the regression model. Feeding data set is usually split into training (70%-80%) and testing (30%-20%) result of which will the plotted to the same graph. Without a detailed explanation of statistics, Table 3-1 suggests criteria to quantitatively assessing errors.

Table 3-1. Regression model errors evaluation

Metrics	Criterion
R-squared	Higher the better (>0.7)
MSE (Mean squared error)	Lower the better
p-value	Lower the better (<0.05)

Whereas,

$$\mathbf{MSE} = \frac{\mathbf{RSS}}{\mathbf{N}} \tag{3-2}$$

$$\mathbf{RSS(w)} \triangleq \sum_{\mathbf{i=1}}^{\mathbf{N}} (\mathbf{y_i} - \omega^{\mathbf{T}}\mathbf{x_i})^2 \tag{3-3}$$

RSS: residual sum of squares

y_i: data

$\omega^T x_i$ (equivalent to): predicted value by regression model

The reliable regression model is then used to predict value at threshold using built-in *predict* function. K-fold cross-validation is a widely used method to test the performance model. In this method, data is split into 'k' mutually exclusive random sample portions known as "folds". The first fold is a validation set while the remaining k-1 folds are training ones. The procedure is repeated k times with each fold will be validation test once. Typical values of k are 5 or 10 depending on computational expense and bias-variance tradeoff [53]. Each procedure, also a readily built-in function, computes an accuracy value from which mean accuracy value is computed. The higher average value and lower the variance between values the more reliable Regression model consequently the prediction is [54].

The evaluation of the model is necessary for the final selection and further implementation. [54] suggest two steps of assessing the model, the model performance metrics and the other is cross-validation. The metrics chosen for this work are listed in Table 3-1. One of the sample interpretation could be R^2. Most efficient model maximizes the value of the coefficient of the determinator (R^2). R=0 yields no relationship between the predictor and the target whereas R^2=1 suggests perfect prediction with no error. In cross-validation, it is important to note that, the training and test data must be different and a random split of the data is recommended.

4 A prototype of the monitoring device for sliding guide shoe

The section presents the design and functionality of a prototype based on the monitoring method developed in Section 3. Among the technical requirements is designed for field installation and test at the elevator(s). Moreover, the design and construction of the prototype require a considerable amount of iterative design and development work ranging from electrical, mechanical and embedded system to software tasks.

4.1 Operation principles

Well-defined operating environment and simple monitoring parameter have enabled straightforward working principle and the composition of the prototype. However, robustness in performance and high accuracy in obtained data are well kept in mind during to design process.

4.1.1 Measurement

The distance of the guide rail surface to the probe is the measuring target. The probe is mounted on the bracket which is mounted to elevator carsling structure.

Inductive sensors are widely used for precision measurement of a conductive target's position. Being not affected by material in the gap between probe and target, inductive sensor is ideal for elevator guide rail which may have lubricating oil and debris on the surface. However, inductive sensors are sensitive to the material of the measuring object, therefore proper calibration and testing are essential to the accuracy of the reading [55]. There are 2 sensors which measure 2 directions of car movement (Figure 23) at a resolution of <0,03mm (for fast reading) and <0,03 mmm of repeatability. The output of the is in voltage (0 ... 10 VDC), linear type for value interpretation.

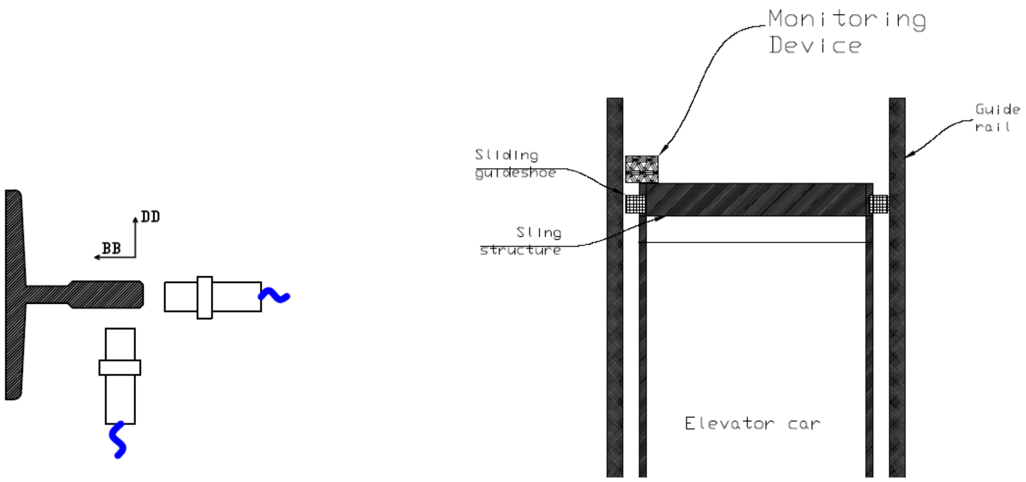


Figure 23. Distance to guide rail measurement

4.1.2 Data acquisition

The output from the sensors was transmitted to the analog input port of the single-board computer Beaglebone Black [56]. The signal is then recorded and stored in the computer memory in predefined and formatted data files. Distance and displacement values are computed manually based on the sensor's linearity principle and preceding laboratory calibrations. Although manual calculations would not be favorable in a large number of data files, this approach allows different interpretation of the data without altering the original values, which is believed to be reasonable at the prototyping phase. Once the method is validated, additional custom program or extending the existing program which also takes into consideration error compensation can be easily implemented.

4.2 The design

Functionality and performance are first elements of selection criteria for components of the prototype. Limited developing time, as well as the competences affected the cost, the build quality and efficiency of the prototype. Improvements made during laboratory and field tests led to different versions of both hardware and software which are fully functional and capable of possible further development at the end of the work.

4.2.1 Architecture

Design to be a stand-alone embedded system, the prototype possesses Beaglebone black as the processing and memory unit, two sensors, one printed circuit board, and one power supply unit. The Beaglebone black (BBB), which is built as a complete computer that runs Linux distribution as an operating system, offers powerful processing power and extensive hardware expansion (up to 92 connection points [57]). BBB runs Python language program, which is considered to be a significant advantage in term of User experience, after suitable library and console being installed. Circuit board distributes appropriate voltage to the sensors (12VDC) and BBB (5VDC) and regulates sensor output signals before transmitting them to BBB's analog pins.
Figure 24 illustrates the system architecture based on data flow sequence, hardware and software distribution of the prototype as well as the monitoring method as a whole with the data analysis phase.

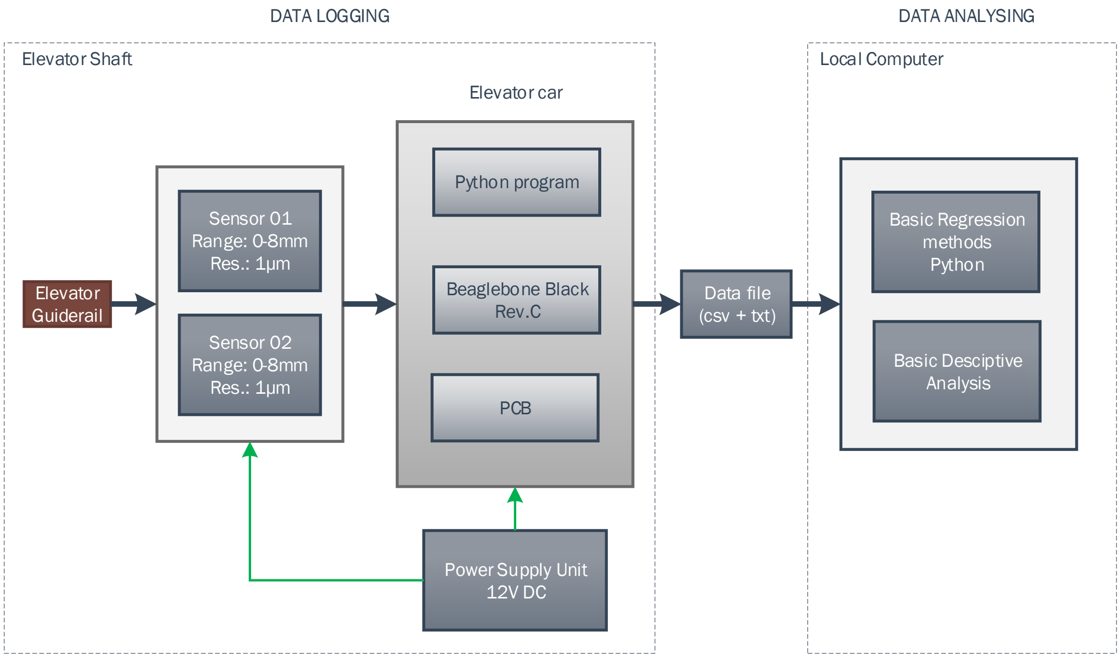


Figure 24. Guide shoe monitoring prototype architecture

For simplicity and minimum development time, User Interface and interaction was minimized to simple power switch and manual plug-in connectors between sensors and the printed circuit board, as well as from the board to personal computer for data transmission (mini-USB port).

4.2.2 Hardware

Inductive sensor

Driveshaft runout measuring, thread detection, valve stroke, and piston positions, thickness measuring, calender roller gap and distance measurement (from 10 µm to 15mm) are among typical applications of inductive sensors. Furthermore, inductive sensor can operate in dirty, hostile environment while maintaining a high accuracy level [55]. Baumer IR18.D08F (BIR) is a high-resolution analog inductive sensor. In addition to key specifications listed in Table 4-1, important features of BIR includes: LED indicator, 3-level teach-in modes, no maintenance or cleaning required [58]

Table 4-1. Baumer IR18.D08F

Specification	
Measuring material	Mild steel, Stainless steel and Aluminium
Measuring distance	0 ... 8 mm
Resolution	<0,03 mm
Sensitivity	1,25 V/mm
Reading	Linearized factor 1
Adjustment	external Teach-in
Supply voltage	12 ... 36 VDC
Output signal	0 ... 10 VDC
Protection class	IP 67

Construction and overall dimensions of BIR are shown in Figure 25. BIR's measuring principle is that it generates a high-frequency electromagnetic field around sensing face. Changes of the field by measuring the object's movement will be detected and converted proportionally into output signals. The all-analog principle allows high measuring speed, excellent repeatability, and low readout noise.

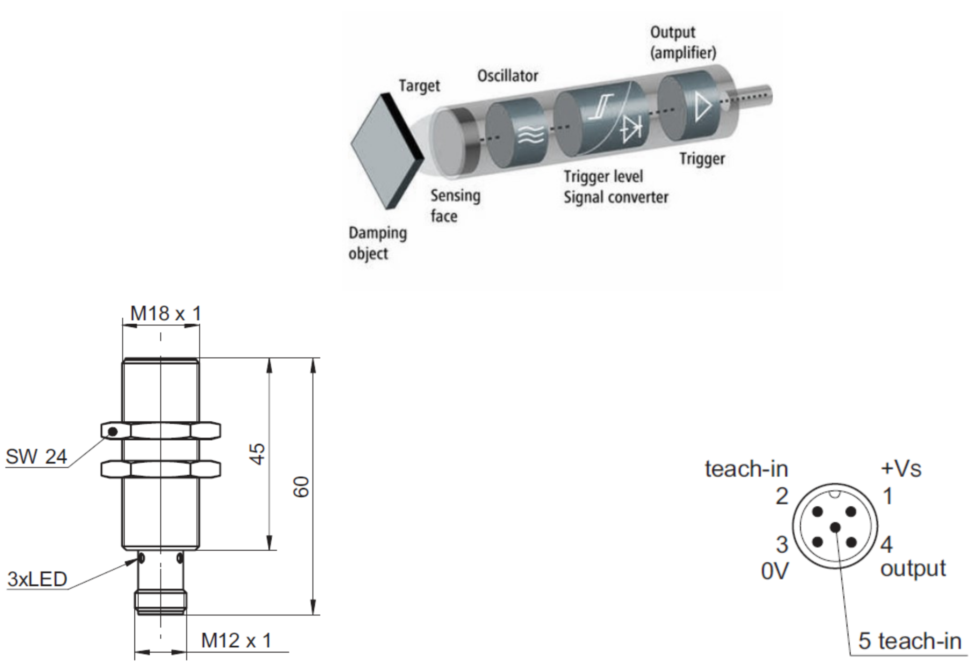


Figure 25. Baumer inductive sensor IR18.D08F (Appendix H and [59])

As important as error is in any measuring equipment, to BIR highly influential sources of errors are: ambient temperature change which causes drifts output (ideal one is 23^0C $\pm$ 5); size and material of measuring object and said to the largest influence on the output is mounting situation with regard to distance between sensing face and target. More details can be found in Appendix H.

Beaglebone black Revision C

Beaglebone black is a full-featured computer at a compact size. Moreover, it runs a Linux-based operating system which has sizable community-supported development platform and open source libraries. These allow rapid prototyping and expandable embedded systems [60] which make BBB ideal for the work. On-board AM335x 1GHz ARM microprocessor can run Linux distributions, Android and even Windows Embedded CE. There is 512MB of DDR3 RAM and 4GB onboard eMMC storage. General purpose input/output offers two 2x23 pin rows coupled with various peripheral interface subsystems enable the BBB to process different forms of output and input, the analog signal is one of them. There is one USB host port is available, and one micro HDMI for monitor screen output. BBB is powered by 5VDC 2A supply (Appendix I).

Printed Circuit Board

The custom printed circuit board (PCB) is designed and built to distribute power supply and regulate sensor signals which are then wired to analog pins on the BBB. Among other electrical components, there are the power switch, fuse, and indicating LED which indicates the sensor reading operation. The PCB is designed and equipped for working with both 12V 5A LiPo battery or direct 220V standard power supply. Details of the design can be found in Appendix F and G.

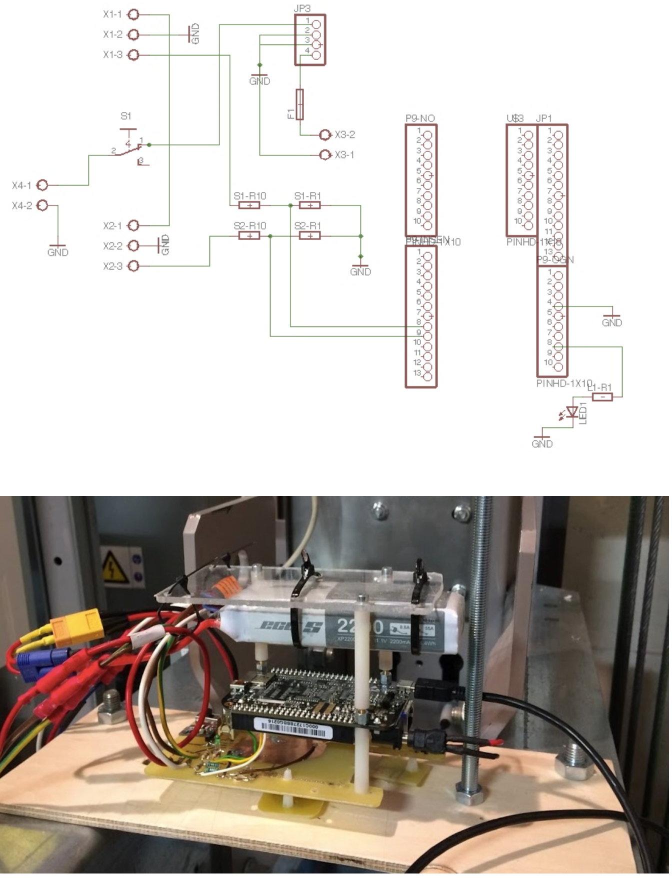


Figure 26. Schematic and prototype

Brackets and fixtures

Additional custom design steel bracket was designed to locate sensors to their operating position. The board and battery (if used) are fixed on carsling top beam. Concerning inheriting error induced by hardware, reliable connection and wiring electric cable was carefully considered during design and installation. Mechanical design of the bracket was to maximize the robustness so that sensor movement is minimized, however, shock or vibration absorbing mechanism were not applied. As a result, under certain conditions (e.g. sudden movement of the car, vibration due to impact) sensor readings are expected to be inaccurate which needs to addressed in data analysis processes.

4.2.3 Software

Operating System

Angstrom 2013-09-24 distribution has been in use throughout the work thanks to its stability and compatibility. Specific instructions can be found from the community website, yet it is to be noted that the process of installing the Operating system encountered compatibility issues, additional third-party programs to be installed (Win32 Disk Manager, PuTTY) as well as trial and errors, extra research efforts to be made.

Programming language

Python was the sole programming language used in the work including programs for the prototype and functions for data visualization, analysis, and machine learning algorithm. Created in 1991 as an interpreted, general-purpose, object-oriented language, Python has steadily grown into a mature widely supported ecosystem with quality packages and libraries for embedded system design as well as data analytics and machine learning applications [61]. Among the key advantages of choosing Python over the remaining languages are its versatility, user ease and friendliness, cross-platform support and processing and memory efficiency which results in n faster performance.

Library

The Adafruit BeagleBone IO Python library named Adafruit_BBIO is installed following instructions at [62]. Adafruit is an American electronics and equipment manufacturer which offers a wide range of modern electronic component with large community supports. Adafruit_BBIO allows (Python) program to get access to GPIO pins on the BBB, among which are the 7 analog pins. There are alternatives to the Adafruit's such as Bonescript library and Cloud9 IDE programming environment, with JavaScript or C languages.

Data acquisition custom programs

Basic embedded programming (in Python) skills have been acquired in order to build the programs which can reliably run the BBB and store sensor data into files (text or spreadsheet format). Descriptive programming practice was pursued in the attempt to have the information obtained from the program to be sufficient for further study and analysis. For example, test location and purpose as well as sensor reading interval, data file length are set as input every time the program started. The filename is the date and time of the creation of the file.

```
20180320_102921.txt - Notepad
File  Edit  Format  View  Help
Date and time of measurement:20180320_102921
Datafile logging interval (seconds): 84600
Reading delay (seconds): 60
Test Location: KoivistoTest specs/ purposes: normal run + grip tap, 60s interval, 23.5h log
Test readings, sensors in position, elevator doesn't run:Sensor00:371.0,    Sensor mk:423.0
----------------------------------------------------
```

	A	B
1	Sensor 00	Sensor mk
2	392	360
3	400	315
4	546	457
5	545	458
6	400	313
7	367	312
8	546	457

Figure 27. Recored data descriptions (top) and original sensor data.

Program file can only run on Command-line basic with the BBB board connected to Windows (or Mac) computer via mini USB port. Full program code is available in Appendix B.

4.3 Configurations and data acquisition

Prior to the deployment of the device, there are two main tasks which need to be attained in order to acquire usable data. Firstly, an appropriate sensor teaching model is set while calibration, as well as associated tolerances and errors, are taken into account. Secondly, suitable reading interval and data package length settings for the program are set so that they can most accurately reflect the operational characteristic and health condition of the monitored guide shoe.

4.3.1 Sensor calibration

Teach-level 3 was set to the two sensors. This mode is factory default mode which can utilize the full measuring range from 0 to 8mm. Calibration of the sensors was done in laboratory condition to determine the ratio between the reading output of the sensor to the displacement of the object:

$$r_s = \frac{\Delta x}{\Delta s} \left(\frac{mm}{mV}\right) \tag{4-1}$$

Where:

r_s: sensor output ratio
Δx: object displacement ($x_2 - x_1$)
Δs: reading difference ($s_2 - s_1$)

Mean value of Δx and Δs in 5 measurements yields r_s= 0,0067568 mm/mV

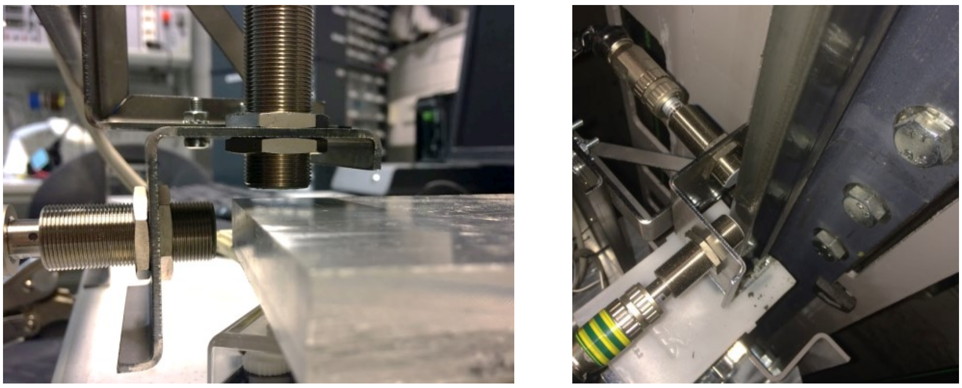


Figure 28. Sensor calibration (left) and elevator installation of the sensors

Calculation of displacement in acquired sensor data used the mentioned r_s. This approach allows system error to be included and compensated by r_s while computing displacement values which then used for data analysis tasks. On the other hand, having the entire monitoring data depends on a single laboratory calibration is the method which needs precision and robustness throughout the process.

4.3.2 Software configuration

Configurations required to initiate the device includes reading interval, which is the time (in seconds) between 2 readings; data file length, which is mostly for safety reason in the even of corruption in software or unexpected shutdown previous data is safe in other saved files, typical file length is 24-hour or 8-hour data. Moreover, descriptive free text input that specifies the data context such as location, test purpose, elevator running state is considered to be helpful in the analysis phase.

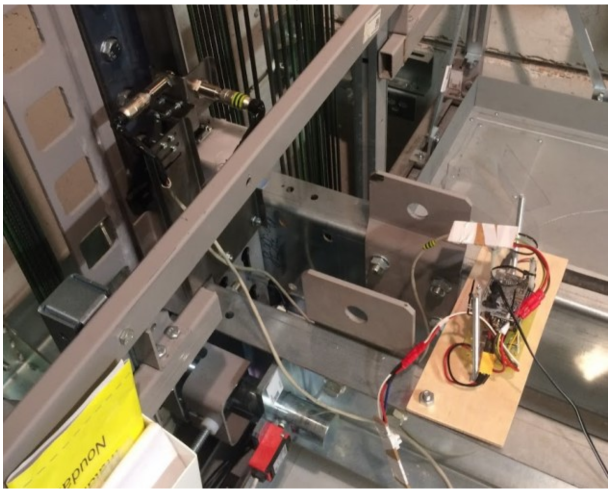


Figure 29. Koivisto elevator installation (sliding guide shoe)

In the tests, the reading interval was set from 20 seconds, 30 seconds or 60 seconds depending on the elevator running state. In a similar manner is the file length, 4 hours, 8 hours or 24 hours. In the course of the work, there are test run (Koivisto) and normal public use (Paasikivi) state of the elevators. It is also important to be noted that while Koivisto has sliding guide shoe which is the primary target of the work, Paasikivi is equipped with roller guide shoe whose operation and monitoring characteristic although are not in the scope of the work, the acquired result is really useful.

4.3.3 Operating procedure

The first step of starting the data recording is the installation of the sensors and locating the device (PCB with BBB) and wiring of the sensors and the supply power. Next step is to connect the device with Personal Computer via mini USB cable. In Windows, run Third-party program named PuTTY to set IP address of the BBB *192.168.7.2* with *root* as username (or run *ssh 192.168.7.2 -l root* in Terminal on MacOS) to get access to BBB command shell. Correct data and time for the BBB can be set using *--set --date="yyyy-mm-dd hh:mm:ss*". Before the data recording program can be started, a terminal multiplexer, as one of the methods to avoid connection timeout, has to be initiated by running *tmux* in the command line. Run device program by typing *python Apollo13.py,* with Apollo13.py is the python program file which is located in the BBB local memory or can be written and saved in a built-in coding environment by running *nano Apollo13.py*. The device running procedure is ended by exiting *tmux* using command input sequence *Ctrl + b* and then *d*. The USB cable can then be detached without further action required. The device is running and sensor data is being recorded which can be confirmed with the indicating LED illuminated at the same interval with the sensor's reading interval.

4.3.4 Real-time reading with ThingSpeak

One of the additional feature tested with the prototype is the real-time pushing sensor signal to the educational version of the Internet of Things (IoT) platform called ThingSpeak by Mathworks. The feature requires a dedicated code set and internet line connected to the 10/100 Ethernet port on BBB board. Due to the fact that candidate elevators are no equipped with internet cable nor WIFI, the capability was not further developed.

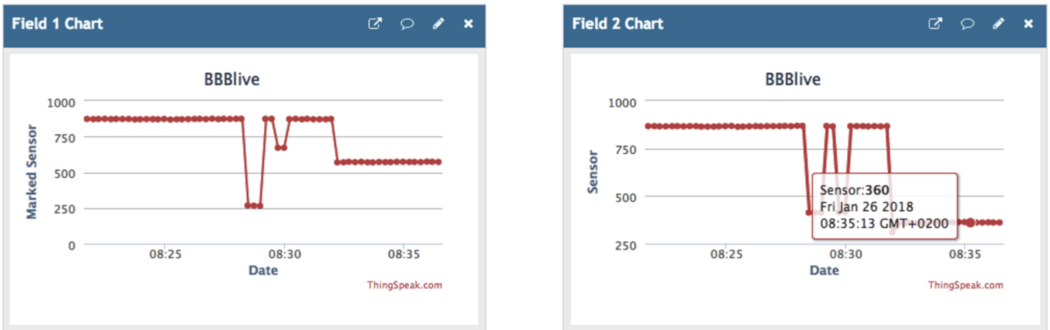


Figure 30. Real-time sensor readings on IoT platform

5 Results

This section presents introductory data visualization techniques and statistical learning models whose results are predictions values and visualization. In addition to validating the method developed in Section 3 and 4, the work also reveals potential issues and limitations which are expected to be useful for future developments. Development tools are Pycharm Edu as the main program while Jupiter notebook was used in cases where it was more productive or responsive.

5.1 Monitoring data acquisition

During the courses of 9 weeks, the prototype was installed in two elevators – Koivisto and Paasikivi located in Hyvinkää. Koivisto is the primary 2-floor travel candidate. It is a test elevator that can be set to run continuously during the test whereas Paasikivi is an in-use elevator for internal personnel. Paasikivi has travel height of 4-floor. Data from the sensor was stored locally in device memory as files. Third-party programs, for instance, WinSCP on Windows computer, was used to extract the files.
The data file is then processed and compiled into to single spreadsheet file format ready for analyzing. In this step, the original sensor reading in voltage was converted into distance value in mm. There were two sets of data that were at sufficient quality for analysis. One set is of 10486 data points, the reading interval was 1 minute at in-use elevator Paasikivi (PAS). The other has 17971 data points with same reading settings at test elevator Koivisto (KOI).

5.2 Premilinary data visualizing and analytics

In addition to advantages mentioned in 4.2.3, Python as a programming language is strongly supported by active scientific computing communities. Consequently, robust Python libraries with detailed instruction are widely available for academic and commercial purposes.

Libraries

Numpy is the fundamental package for efficient data storage and operations. It is a powerful tool for reading and writing array data. Numpy arrays form the foundation of nearly the entire higher-level data science tools.

Pandas provide productive structured data analysis environment. Indexing, labeling, and other frameworks and spreadsheet functionalities are key features of the package.

Matplotlib is the most popular package for plotting and other 2D data visualizing in Python. Its plots allow zooming and panning.

Scikit-learn (SKL) is the best-known package for implementation of machine learning algorithms. SKL provides numerous convenience functions for common preprocessing tasks such as k-fold validation, normalization [50] and [63]

Data exploring

Scatter plot was used to gain an overall understanding of the data. Disruption in the data acquisition process or elevator status can be promptly identified from the graph. Figure 31 suggests a reasonable performance of the prototype, and operational characteristics of the elevators are well presented. The data points are in blue and the residual line is red. The vertical axis is the distance (to recorded static position) in mm. The horizontal axis is time in minute, 0 is at the beginning of the measurement.

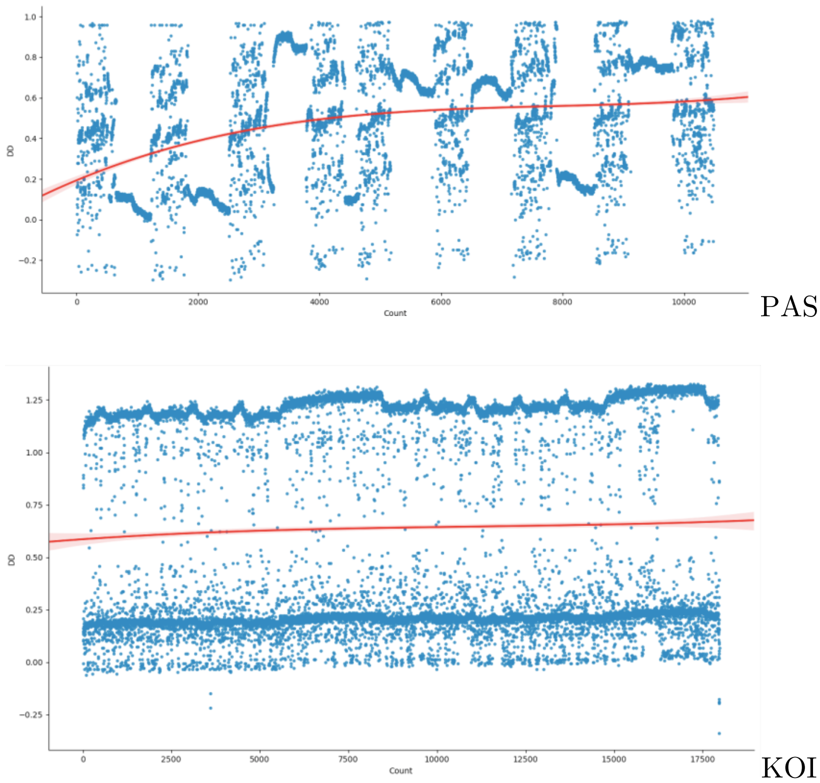


Figure 31. Displacement data from PAS (upper) and KOI elevator

For PAS, as an in-use elevator, continuous sensor reading captured intermitted static values which reflected elevator's free-time every day outside of working hours, whereas for test elevator KOI, which was set to run continuously, is uninterrupted values. The interpretations called for a data filtering mechanism in which statics values should be excluded before going forward. Additionally, maximum values in PAS were fairly stable while minimum values showed shifting patterns which can be understood as large movement occurred at a certain location during the elevator travel (2 floors). Reserved situation recorded in KOI. It also became evident that the sensor settings were relatively sensitive to the movement of the elevator. The code was inspired by [54] and the full version can be found in Appendix D

Data cleaning and line fitting

Since health condition of the object (sliding guide shoe) is determined by worst-case scenarios, which are maximum recorded displacement values during operation (elevator run), data which can be put forth for prediction modeling are maximum and minimum values. Construction of the new data set can be illustrated as follows:

$$original\ set = \begin{bmatrix} a_{01} \\ a_{02} \\ a_{03} \\ \vdots \\ a_{0n} \end{bmatrix} \rightarrow filtered\ set = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix}$$

$$\boldsymbol{a_i = \bar{a}_{imax} - \bar{a}_{imin}} \tag{5-1}$$

Where,

$\bar{a}_{imax}$: mean value of (M) amount of maximum values in every (N) amount of data points in original data

$\bar{a}_{imin}$: mean value of (M) amount of minimum values in every (N) amount of data points in original data

Different values of M and N were calculated in the search for the most appropriate combination. Figure 32 emphasizes the effect of the choices of M to the presentation of the data - 200 versus 20 maximum (and minimum) values in every 1440 data points (which is equivalent to one operation day). It is also worth noticing that for simplification, indices (x value) of the sensor data are omitted, instead the new indices of the mean value is set to be the value of N. Therefore, the smaller N statistically higher the fidelity of the new data set.

$$\boldsymbol{x} = \begin{bmatrix} N \\ 2N \\ 3N \\ \vdots \\ nN \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix} \tag{5-2}$$

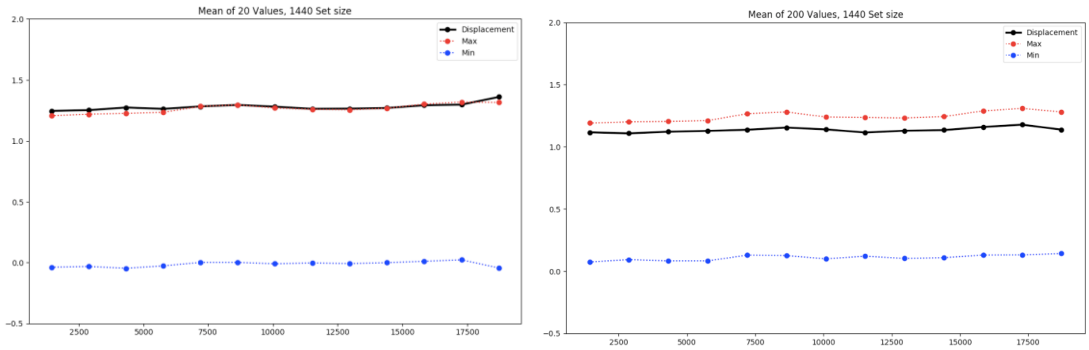


Figure 32. Maximum, minimum and displacement values (KOI)

For the filtered dataset, it is the case that the amount of the available was, to some extent, insufficient. Larger data set would allow the higher value of N or with the same N, there would be a larger sample population which is highly favorable for pattern recognition and curve fitting.

Curve-fitting

Following the approach specified in section 3.3, different regression models were applied to the displacement data and the results are plotted in Figure 33.

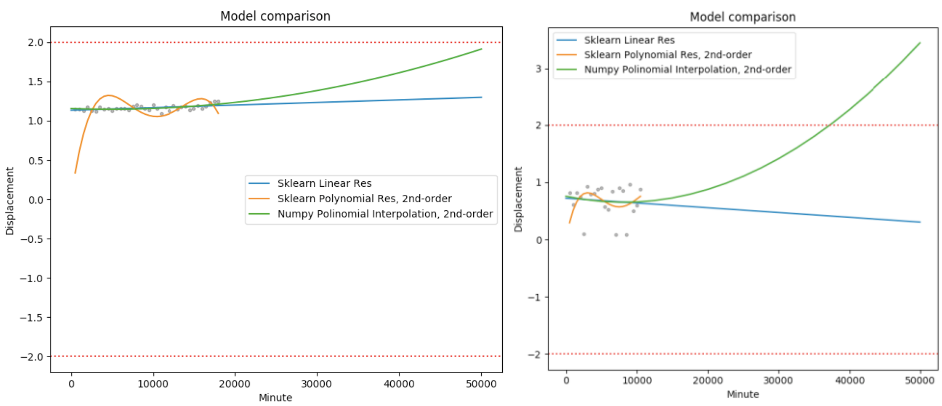
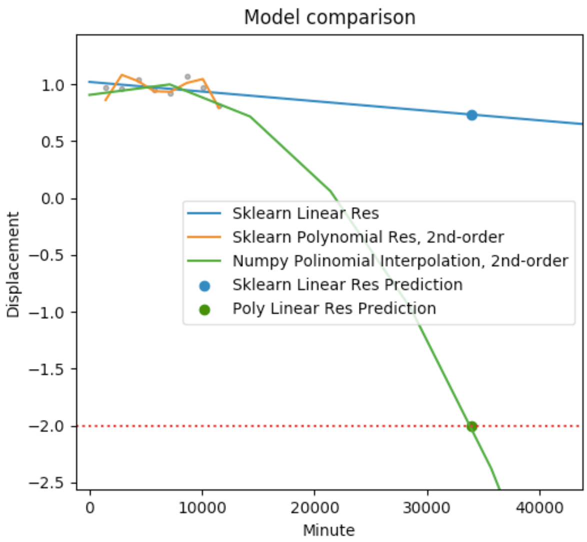


Figure 33. Regression model applied to displacement data sets

Displacement threshold of 2mm is visualized as red dash-lines. The blue line is created by Linear regression model while the orange line is from second-order polynomial regression. Both are Sci-kit learn libraries'. The green line, however, is created by Numpy polynomial regression, also in second-order. The grey dots are collected data while the lines continuously project future values. It is evident that data in PAS is noticeably fluctuating and resulted in a higher coefficient of the curves. This yields different M and N combination would reflect the condition more accurately. Also available at this stage is the prediction of displacement which is believed to be useful for preliminary model evaluation.



SK linear res.: 0.73272263
Numpy poly. inter.: -2.0062

Figure 34. Prediction model comparison

At M=50, N= 1440, x_f = 45000 that can be interpreted as – prediction of displacement value, for the next three-week time, in mean values of (50) maximum displacement recorded per 8-hour-working-day data set (1440), linear model yields 0.7 mm whereas polynomial one calls for the critical condition at allowable values at 2.0 mm. Sensible results are not expected at this stage.

Error and uncertainty consideration

It is evident that systematic error as well as random error did occur in the original data set. However, more test cases and hardware calibrations are required in order to obtain a reliable uncertainty calculation. On the other hand, the effect of the error types on the large data size, that is collected over a prolong period of time, by continuous reading method also need to be examined. For the simplification of the work, therefore, error and uncertainty were not calculated.

5.3 Current condition and health state

The mean of the recorded displacement values $\bar{y}$ indicates the current condition of the subject sliding guide shoe.

$$\bar{y} = \frac{\sum_{i=1}^{n} a_i}{n} \tag{5-3}$$

According to equation (5-1), y has different size and value depending on M and N values. Based on the available data, tt is decided that arithmetic mean recorded in one day (N=1440) from one hundred boundary values (M=100) is the most appropriate estimation that can reliably reflect operation condition at a reasonable amount of data needed for the prediction model.

Table 5-1. Current status (M=100, N=1440)

	PAS		KOI	
	DD	BB	DD	BB
Data size (n data point)	8	-	13	13
Current displacement (mm)	0.826	-	1.204	1.073
Permissible threshold (mm)	2.00			

According to maintenance criteria specified in section 3.1, the analysis concludes the sliding shoe is functioning within the allowable zone and no maintaining activities required.

5.4 Trend and Prediction

Obtained results thus far suggest that it is possible to attain the failure trend from available data. Prediction can also be made as the result of a suitable selection of the predictive model. Two models from two libraries were employed in making predictions. In order to make a quantitative evaluation of the results, Mean squared error (MSE) and Coefficient of determinator (R^2) were computed for each prediction.

5.4.1 Prediction modeling

A Python program was written for computing prediction and its statistical metrics according to different M, N values. Full calculation results can be found in Appendix M, Figure 35 partially reveals the table.

ID	BB	DD	n	w	Last time	Last displacement	k	Linear Predictio	MSE	R2	Poly Prediction	MSE	R2
KOI	0	1	1440	50	18720	1.299459459	50000	1.37326684	0.002982	-6.81703	1.617324641	0.000112	0.707095
KOI	0	1	1440	75	18720	1.258918919	50000	1.34305039	0.002779	-6.30031	1.49444907	0.000141	0.630747
KOI	0	1	2880	100	20160	1.228783784	50000	1.27682995	0.000419	-0.40818	0.786906916	0.000188	0.368618
KOI	0	1	2880	150	20160	1.176756757	50000	1.21587681	0.000593	-0.01274	0.440329111	0.000348	0.40608
KOI	0	1	720	50	18000	1.299459459	50000	1.34533578	0.00491	-3.91061	1.947212833	0.000585	0.414807
KOI	0	1	720	100	18000	1.228783784	50000	1.28029328	0.004813	-4.27071	1.907780026	0.000492	0.460798
PAS	0	1	1440	50	11520	0.81527027	50000	0.59756962	0.041152	-6.8058	-6.312040543	0.003279	0.377942
PAS	0	1	1440	75	11520	0.721891892	50000	0.46929132	0.050284	-8.25118	-6.655323463	0.003172	0.416435

Figure 35. Prediction model comparison (M is w, N is n)

Each set of M and N was applied to both BB and DD in PAS and KOI when it was possible. The prediction value of each model was followed by its corresponding MSE and R^2. The overall interpretation was that the predictions made by polynomial regression were considerably more consistent across data sets and M, N values. On the contrary, the linear regression model performed inconsistently produces a R^2 value ranging from -14,93 to -0,01 while it was 0,03 to 0.99 for polynomials. The contrast between the two models was not as significant in the case of MSE, the least sensible prediction from Polynomial model at M=150, N=2880 is -8,29 whereas for Linear model is 0,10 at M=85 and N=720. x=50000 for all prediction. Table 5-2 presents the best score results following criteria listed in Table 3-1. Current displacement value is in the bracket whereas P is predicted value.

Table 5-2. Best scores prediction model

	Linear Model	Polynomial model
Best score MSE	M=50, N=1440 P=1.135 (1.089)	M=150, N=2880 P= -8.269 (0.826)
Best score R2	M=150, N=2880 P= 0.562 (0.826)	M=150, N=2880 P= -8.269 (0.826)
Best score MSE & R2	M=50, N=1440 P=1.135 (1.089)	M=150, N=2880 P= -8.269 (0.826)

According to said criteria, it is evident that Polynomial regression model at M=150 N=2880 would be the most suitable one. However, those best scores are from only one data set of PAS elevator. When applying the model to a different dataset (DD of KOI), obtained scores are among the lowest. This indicates the inconsistency in the model as well as score-based assessment. One major root cause is believed to be the insufficient amount of data. The other cause would be best scores (maximum R^2 or minimum MSE) might not the most appropriate selectors, rather the model with certain M and N that performs consistently across data sets. Data size can be increased by selecting the smaller value of N and more consistent scores are from the Linear model. Following this approach, first-order Linear regression from Scikit-learn at M=50, N= 1440 is selected.

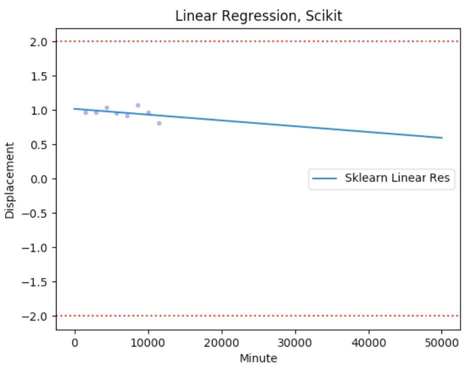
5.4.2 Prediction results

Final results of the condition method are presented in Table 5-1 and Table 5-3 and failure trend are plotted in Figure 36. For Paasikivi elevator (PAS), the sliding guide shoe will reach the end of its service life in 235 days from the last day the prototype recorded the data. It is 171 days in the case of Koivisto elevator (KOI).

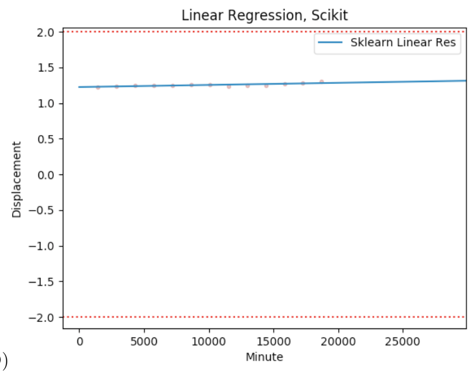
Table 5-3. Predictions by Linear regression model (M=50, N=1440)

	PAS		KOI	
	DD	BB	DD	BB
Present displacement (mm)	0.826	-	1.204	1.073
Time of measurement (min)	18720	-	18720	18720
Time at (2mm) threshold (min)	358000	-	265000	1650000

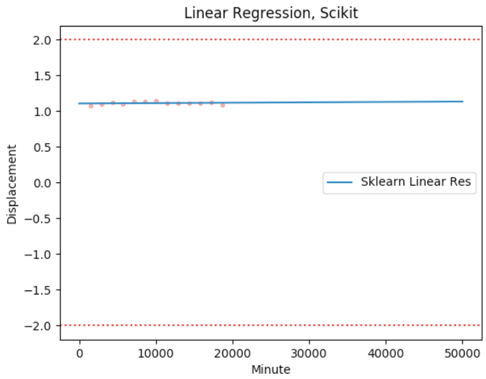
Trend plot and coefficients:



(PAS, DD)



(KOI, DD)



(KOI, BB)

Figure 36. Prediction/ failure trend lines

It is highly possible that in the larger data set, different prediction values and even different performance of the models would occur and result in different predicting values. In that case, the mean of prediction values over a pre-defined period would be the one way to determine the time-to-failure prediction. This also emphasizes the importance of further testing and cross-validating of the model.

6 Discussion

For the condition monitoring method of the sliding guide shoes as well as other potential applications, there are three main areas of developments. Regarding data acquisition, large and high fidelity, precise condition data could be achieved with current hardware. The targets are the precision in sensor calibration, consistent reading accuracy in different conditions such as different guide rail type, travel speed, and operating environment. The work would mostly involve software development and would not be time-consuming thanks to hardware's processing power and established procedures presented in the report. Multi-purpose program(s) could help to compare reading settings, detecting outliners and anomalies while storing condition data.

In order to minimize systematic errors, it is advisable to implement a sensor reset mechanism and procedure into the monitoring device. However, because health status and time-to-failure prediction are solely based on the data from sensors, pre-conditions for sensor resets and post-reset data evaluation are required prior to the data processing for health management purposes.

For data transmission, the ability to access the condition data remotely depends on the elevator shaft (and/ or building) network infrastructure, preferably wireless. Current hardware requires additional module integration in order to gain wireless connection capability.

In a similar manner is data analysis, including diagnostics and prognostics. The well-developed program would shorten the runtime and reduce manual tasks. Prediction model development, on the other hand, would require a much larger amount of condition data as well as the validation process. Even though the current model is linear regression based on data that was available to the work, the model might change with sufficient data size in which proper machine learning procedure can be employed, if chosen.

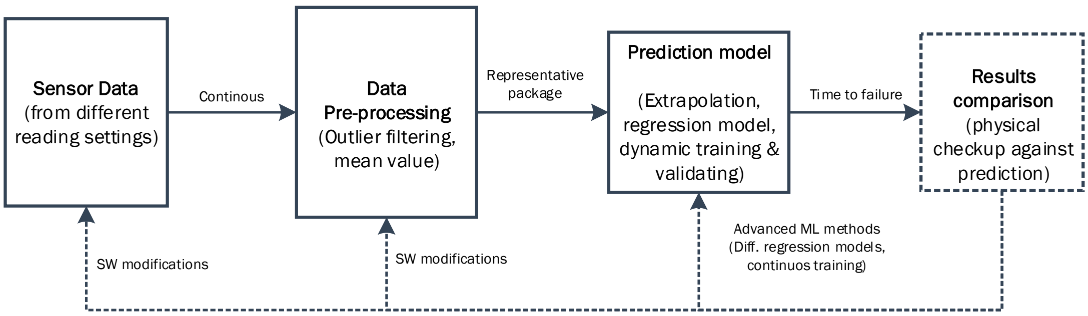


Figure 37. Proposal of prediction model development

Thirdly, failure trend analysis, whose result is the time-to-failure prediction model, would require reasonably extensive work. Figure 37 proposes one approach of the iteration of model development by acquiring then comparing the prediction values with the physical status of the test piece. Prediction values would also need further validation in the various operating environment whereas monitoring parameters and/or model modifications might be needed when unmatches between model output and physical object measurements occur. These tests would also help gain knowledge about the component failures which consequently reduces the time and cost of the design and installation of the component.

Additionally, as mentioned in 5.4.1, appropriate data size (for instance, 100 instead of 13) plays a significant role in the accuracy of the prediction values.

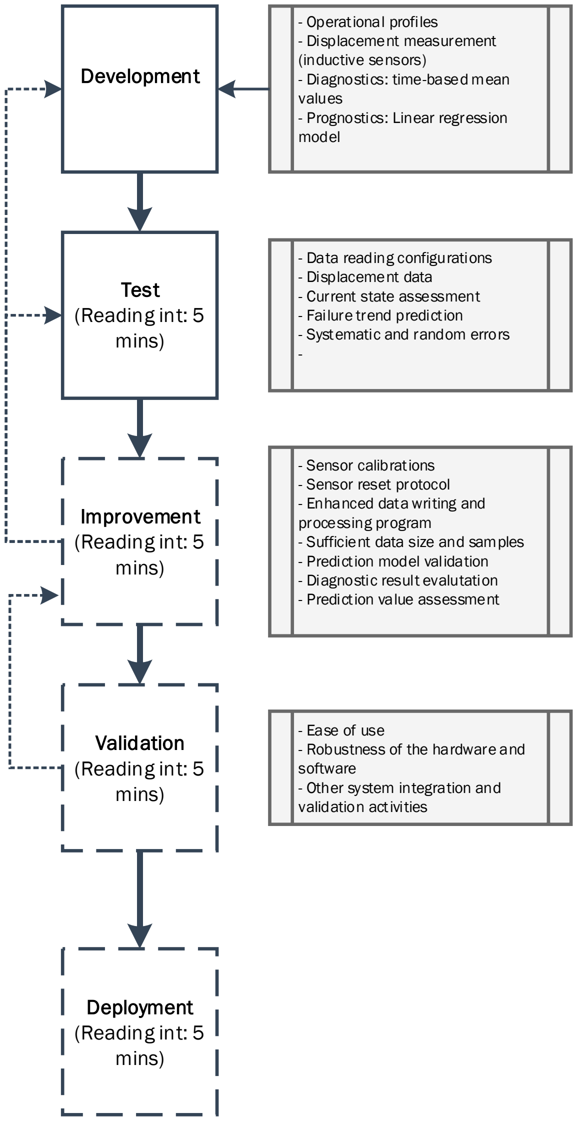


Figure 38. Contents of development phases of the method

Towards the finalization phase of the method development and integration, it is important to recognize the key metrics are robustness and ease of use. They are the dominant factors for a successful implementation of new capability to any system [64]. On the other hand, software for machinery and systems has grown rapidly in length (measured in code-line), for example, software on a high-end automobile has grown from ten to a hundred and fifty million code-line from the year 2010 to 2016 [65]. In the case of the elevator system, at a different rate and magnitude nevertheless, there would be a similar demand for new software and electronic architecture that allows reliable and smooth integration that meets the requirement of increased complexity and interdependencies while maintaining the robustness and ease of use. Standardization, modularization of the interfaces for both hardware and software are among the method that needs to be employed in meeting those requirements.

Summary of the contents of the method development can be visualized in Figure 38 where activities in continuous-line boxes have been performed, the ones which are in dashed-line boxes are development suggestions.

Another aspect to consider in developing a health monitoring system for the elevator is the utilization of data recorded from one or more subsystem monitoring device. The work would help define the benefit-cost ratio or business case study. Article [66] introduced a framework to analyze the Value of Information (VoI) of the condition data in order to assess the effects and benefits of a Health Management system. According to the study the framework requires whole system probabilistic and structural reliability models. As a result, a technical solution for data storage and accessibility is created. As for the data collected by the two inductive sensors in the work, one of the use could be a characterization of the movement of the elevator in term of its relative position to the guide rail during the in-service time.

7 Conclusion

The thesis has aimed to develop a remote condition monitoring method for guiding element and diverting pulley in a passenger elevator. As a result, existing challenges and limitations in the maintenance of the target components should be mitigated. The method development process would involve implementing latest methodology while assessing and validating the method via a prototype. Consequently, future development work would save time and cost based on learnings and findings during the work.

In the attempt to achieve the goals, studies of condition monitoring techniques in various industries (chapter 2) have a laid reliable foundation for the work. The condition monitoring that is developed during the work consists of the condition data acquisition and the analysis of such data (chapter 3). The sliding guide shoe on passenger elevator carsling, considered as the essential component for Ride comfort classification of an elevator, is subjected to testing of the method. Based on the understanding of the requirement of the maintenance of the component, Monitoring parameter is simplified as the effective displacement of the car to the Guide rail during elevator-run. The two inductive sensors, which are controlled by an embedded system built on Beaglebone Black and Angstrom Linux distribution, recorded the data (chapter 4). The analysis of the data, by employing data visualization methods and linear regression models in Python programming language, resulted in the verification of the present operating condition as well as the prediction of time-to-maintenance (chapter 5).

One advantage of the method is the simplicity in reading and in the interpretation of the condition indicator and its data. The acquired results are easy to understand and straight forward for decision-making processes such as maintenance planning and deployment. The other advantage is the generic nature of the displacement data from which other elevator performance monitoring techniques can utilize. On the other hand, the indirect and generic data has led to the sensitivity of the method to the size as well as the integrity of the data. Sufficient large data size and customized data purification techniques are needed. Within the scope of the work, arithmetic mean value of three-week data size has raised a reliability issue to the final result as well as the selection of the underlying prediction model, linear regression of the displacement over time. Additionally, the method prototype did not contain the remote access capability including the remote access to collected data as well as the control of the device. Considerable longer development time and wireless communication infrastructure are required for such capability integration. Although data collected by the method can be utilized for monitoring other performance monitoring metrics of the elevator, the method did not present a framework upon which method for other components can be built.

In conclusion, the success of the development of condition monitoring method for mechanical components in passenger elevator is defined by the accuracy of the method's outcome as well as the economic values created by the implementation of such method. Reliable performance and high fidelity data is the outcome of the robust engineering of hardware and software development based on the relevant understanding of the profile of the subjected component or system. Additionally, the return values of a condition monitoring method not only involves Return on investment index including development and implementation time and costs, but also the ease of deployment and ease of use of the method that ultimately should lead to lowered operation cost and optimized life-time value of the component and system of the elevator.

References

1. Al-Kodmany K. Tall Buildings and Elevators: A Review of Recent Technological Advances. Buildings. 2015 -09-17;5(3):1070-104.

2. MAX - Predictive maintenance solution - thyssenkrupp Elevator [Internet]. [cited May 6, 2018]. Available from: https://max.thyssenkrupp-elevator.com/en/.

3. Tshilidzi Marwala. Condition Monitoring Using Computational Intelligence Methods: Applications in Mechanical and Electrical Systems. 2012th ed. Springer; 2012.

4. Andrew K.S.Jardine, Daming Lin, Dragan Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical Systems and Signal Processing. 2006 /10/01;20(7):1483-510.

5. Hajime Yamashina, Shunsuke Otani. Cost-optimized maintenance of the elevator – single unit case. J of Qual in Maintenance Eng. 2001 March 1,;7(1):49-70.

6. Heng A, Zhang S, Tan ACC, Mathew J. Rotating machinery prognostics: State of the art, challenges and opportunities. Mechanical Systems and Signal Processing. 2009 April 1,;23(3):724-39.

7. Tan AC, Heng A, Mathew J. Utilising Reliability and Condition Monitoring Data for Asset Health Prognosis. Asset Condition, Information Systems and Decision Models. 2012:89-103.

8. Mobley RK. An introduction to predictive maintenance. 2nd ed ed. Butterworth-Heinemann; 2002.

9. Sikorska JZ, Hodkiewicz M, Ma L. Prognostic modelling options for remaining useful life estimation by industry. Mechanical Systems and Signal Processing. 2011 July 1,;25(5):1803-36.

10. T. Benkedjouh, K. Medjaher, N.Zerhouni, S. Rechak. Remaining useful life estimation based on nonlinear feature reduction and support vector regression. Engineering Applications of Artificial Intelligence. 2013 /08/01;26(7):1751-60.

11. Shuai Zheng, Kosta Ristovski, Ahmed Farahat, Chetan Gupta. Long Short-Term Memory Network for Remaining Useful Life estimation. 2017 IEEE International Conference on Prognostics and Health Management. 2017 June.

12. Irene M. Gregory. Self-Aware Vehicles: Mission and Performance Adaptation to System Health Degradation. In: 16th AIAA Aviation Technology, Integration, and Operations Conference.

13. EN 13306 Maintenance - Maintenance terminology, 2001).

14. K. Goebel, M. Daigle, A. Saxena, S. Sankararaman, I. Roychoudhury, J. Celaya. Prognostics: The Science of Making Predictions. CreateSpace Independent Publishing Platform; 2017.

15. Abhinav Saxena, Indranil Roychoudhury, Jose Celaya, Sankalita Saha, Bhaskar Saha, Kai Goebel. Requirements Specification for Prognostics Performance - An Overview. Reston: American Institute of Aeronautics and Astronautics; Jan 1, 2010.

16. Saxena A. Knowledge-based architecture for integrated condition based maintenance of engineering systems [dissertation]. ProQuest Dissertations Publishing; 2007.

17. Ulrich KT, Eppinger SD. Product design and development. 5. ed. ed. New York, NY: McGraw-Hill Irwin; 2012.

18. Atamuradov V, Medjaher K, Dersin P, Lamoureux B, Zerhouni N. Prognostics and health management for maintenance practitioners - Review, implementation and tools evaluation. International Journal of Prognostics and Health Management. 2017 Dec 11,;8:1-31.

19. Niu G, Yang B, Pecht M. Development of an optimized condition-based maintenance system by data fusion and reliability-centered maintenance. Reliability Engineering & System Safety. 2010 July;95(7):786-96.

20. Rafael Gouriveau, Kamal Medjaher, Noureddine Zerhouni. From Prognostics and Health Systems Management to Predictive Maintenance 1: Monitoring and Prognostics (Mechanical Engineering and Solid Mechanics: Reliability of Multiphysical Systems). Wiley-ISTE; 2016.

21. Maurizio Di Paolo Emilio. Data Acquisition Systems: From Fundamentals to Applied Design. Springer; 2013.

22. edited by John G. Webster, Halit Eren. Measurement, instrumentation, and sensors handbook : spatial, mechanical, thermal, and radiation measurement. Second edition ed. Boca Raton: CRC Press, Taylor & Francis Group; 2014.

23. Morris AS, Langari R. Measurement and instrumentation: theory and application. Academic Press; 2012.

24. Kutz M. Handbook of Measurement in Science and Engineering, Volume 1. US: John Wiley & Sons Inc; 2013.

25. Inman DJ, Singh R. Engineering vibration. 4th ed., internat. ed. ed. Boston [u.a.]: Pearson; 2014.

26. 2017 ASHRAE® Handbook - Fundamentals. American Society of Heating, Refrigerating and Air-Conditioning Engineers; 2017.

27. Tavner P. Condition monitoring of rotating electrical machines. 1. publ. ed. London: Inst. of Engineering and Technology; 2008.

28. Reichard KM, Van Dyke M, Maynard K. Application of sensor fusion and signal classification techniques in a distributed machinery condition monitoring system. SPIE; Apr 3, 2000.

29. Designing Algorithms for Condition Monitoring and Predictive Maintenance [Internet]. [cited 25.4.2018]. Available from: https://se.mathworks.com/help/predmaint/gs/designing-algorithms-for-condition-monitoring-and-predictive-maintenance.html.

30. Features - Predictive Maintenance Toolbox [Internet]. [cited Apr 25, 2018]. Available from: https://se.mathworks.com/products/predictive-maintenance/features.html.

31. Kim H, Tan ACC, Mathew J, Kim EYH, Choi B. Machine Prognostics Based on Health State Estimation Using SVM. Asset Condition, Information Systems and Decision Models. 2012:169-86.

32. Hastie T, Friedman JH, Tibshirani R. The elements of statistical learning. 2. ed. ed. New York [u.a.]: Springer; 2009.

33. Tadeusz Stepinski, Tadeusz Uhl, Wieslaw Staszewski. Advanced Structural Damage Detection. GB: John Wiley & Sons Inc; 2013.

34. Nam-Ho Kim. Prognostics and Health Management of Engineering Systems. 1st ed. 2017 ed. DE: Springer Verlag; 2017.

35. Research and technology goals and objectives for Integrated Vehicle Health Management (IVHM). USA: NASA Technical Reports Server; 1992 Oct 10,.

36. Fong Shi, inventor; Boeing Co, assignee. Aircraft maintenance and inspection with data analytics enhancement. patent US20170236075A1. 2016 -02-12.

37. Regattieri A, Giazzi A, Gamberi M, Gamberini R. An innovative method to optimize the maintenance policies in an aircraft: General framework and case study. Journal of Air Transport Management. 2015 May;44-45:8-20.

38. Vandawaker RM, Jacques DR, Freels JK. Impact of Prognostic Uncertainty in System Health Monitoring. nternational Journal of Prognostics and Health Management. 2015;6(2).

39. Tipaldi M, Bruenjes B. Spacecraft health monitoring and management systems. IEEE; May 2014.

40. Ross RW. Integrated vehicle health management in aerospace structures. In: Structural Health Monitoring (SHM) in Aerospace Structures. Elsevier Ltd; 2016. p. 3-31.

41. Sanjay Garg. Propulsion Controls and Diagnostics Research in Support of NASA Aeronautics and Exploration Mission Programs. Hampton: NASA/Langley Research Center; 2011 Jan 1,.

42. Aldous L, Smith T, Bucknall R, Thompson P. Uncertainty analysis in ship performance monitoring. Ocean Engineering. 2015 Dec 1,;110:29-38.

43. UNITED NATIONS CONFERENCE ON TRADE AND DEVELOPMENT. Review of Maritime Transport 2017. UNITED NATIONS; 2018.

44. Lazakis I, Raptodimos Y, Varelas T. Predicting ship machinery system condition through analytical reliability tools and artificial neural networks. Ocean Engineering. 2018 Mar 15,;152:404-15.

45. Norden C, Hribernik K, Ghrairi Z, Thoben K, Fuggini C. New Approaches to Through-life Asset Management in the Maritime Industry. Procedia CIRP. 2013;11:219-24.

46. Wolfgang Klein, inventor; Elevator guide shoe. USA patent US4271932A. 1978 -09-01.

47. Hans Kocher, Hubert Steiner, Stephan Hess, inventors; Inventio AG, assignee. Sliding guide shoe for an elevator. 2016 -11-15.

48. Palmer PB, O'Connell DG. Regression analysis for prediction: understanding the process. Cardiopulmonary physical therapy journal. 2009 Sep;20(3):23.

49. James G, Witten D, Hastie T, Tibshirani R. An Introduction To Statistical Learning. Corrected at 6th printing ed. New York: Springer; 2013.

50. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12(Oct):2825-30.

51. Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining, G Geoffrey Vining. Introduction to Linear Regression Analysis. 5th ed. John Wiley & Sons; 2012.

52. Davy Cielen, Arno D. B. Meysman, Mohamed Ali. Introducing Data Science: Big data, machine learning, and more, using Python tools. 1st ed. Manning Publications; 2016.

53. Raúl Garreta, Gavin Hackeling, Trent Hauck, Guillermo Moncecchi. scikit-learn : Machine Learning Simplified. Packt Publishing; 2017.

54. Andreas C. Müller, Sarah Guido. Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media; 2016.

55. Wilson, Jon S (Ed). Sensor Technology Handbook. Burlington: Elsevier; 2005.

56. Giometti R. BeagleBone Essentials. Packt Publishing; 2015.

57. Hiam A. Learning BeagleBone Python Programming. Packt Publishing; 2015.

58. IR18.D08F-F60.UA1E.7BO [Internet]. [cited Jul 7, 2018]. Available from: https://www.baumer.com/sg/en/p/24571.

59. Functionality of inductive sensors [Internet]. [cited Jul 7, 2018]. Available from: https://www.baumer.com/be/en/a/know-how_function_inductive-sensors.

60. Barrett SF, Kridner J. Bad to the bone. San Rafael, Calif.: Morgan & Claypool Publ; 2013.

61. Sjardin B, Massaron L, Boschetti A. Large Scale Machine Learning with Python. Birmingham, England: Packt Publishing; 2016.

62. Using the Adafruit_BBIO Library [Internet]. [cited Jul 7, 2018]. Available from: https://learn.adafruit.com/setting-up-io-python-library-on-beaglebone-black/using-the-bbio-library.

63. Laura Igual, Santi Seguí. Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications. Springer; 2017.

64. Lindgren E. US Air Force Research Laboratory Perspective on Structural Health Monitoring in Support of Risk Management. 1. 2018 /07/02;4(1).

65. Rethinking car software and electronics architecture | McKinsey & Company [Internet].; 2018 [updated February; cited Jul 24, 2018]. Available from: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/rethinking-car-software-and-electronics-architecture.

66. Straub D, Chatzi E, Bismut E, Courage W, Döhler M, Faber MH, et al. Value of information: A roadmap to quantifying the benefit of structural health monitoring. Aug 2017; Vienna, Austria: ICOSSAR - 12th International Conference on Structural Safety & Reliability; 2017.

APPENDIX

A. Matlab Predictive maintenance Toolbox

Sample from an algorithm syntax in Reference document (30)

Train Linear Degradation Model

Load training data.

```
load('linTrainVectors.mat')
mdl = linearDegradationModel;
fit(mdl,linTrainVectors)
```

Train Reliability Survival Model

```
load('reliabilityData.mat')
mdl = reliabilitySurvivalModel;
fit(mdl,reliabilityData,"hours")
```

Train Hash Similarity Model Using Tabular Data

```
load('hashTrainTables.mat')
mdl = hashSimilarityModel('Method',@(x) [mean(x),std(x),kurtosis(x),median(x)]);
fit(mdl,hashTrainTables,"Time","Condition")
```

Predict RUL Using Covariate Survival Model

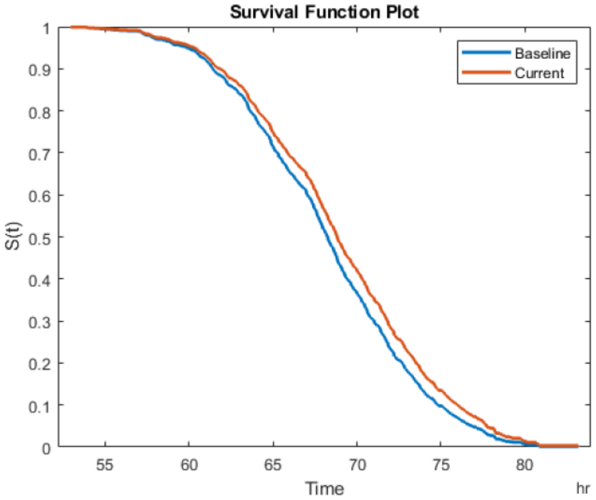
```
load('covariateData.mat')
mdl = covariateSurvivalModel('LifeTimeVariable',"DischargeTime",'LifeTimeUnit',"hours",...
'DataVariables',["Temperature","Load","Manufacturer"],'EncodedVariables',"Manufacturer
fit(mdl,covariateData)
Successful convergence: Norm of gradient less than OPTIONS.TolFun
```

TestBatteryLoad.

```
TestBatteryLoad = 25;
TestAmbientTemperature = 60;
DischargeTime = hours(30);
TestData = timetable(TestBatteryLoad,TestAmbientTemperature,'B','RowTimes',hours(30));
TestData.Properties.VariableNames = {'Temperature','Load','Manufacturer'};
TestData.Properties.DimensionNames{1} = 'DischargeTime';
```

Predict the RUL for the battery.

```
estRUL = predictRUL(mdl,TestData)

estRUL = duration
38.657 hr

plot(mdl,TestData)
```



B. Python code on Beaglebone Black for writing sensors data

Code name: Apollo13.py

```
import Adafruit_BBIO.ADC as ADC
import Adafruit_BBIO.GPIO as GPIO
import time
import sys
import csv
sensor_00 = 'P9_33'
sensor_mk = 'P9_35'
pinLED = 'P9_41'
ADC.setup()
GPIO.setup(pinLED, GPIO.OUT)

print('Sensor reading test..........')+ "\n"
test_rw00 = ADC.read_raw(sensor_00)
test_rwmk = ADC.read_raw(sensor_mk)
print('Sensor 00: %s' %test_rw00) + "\n"
print('Sensor mk: %s'%test_rwmk )+ "\n"

LogTime = int(input("Data file logging interval (seconds):"))
delay = int(raw_input("Reading delay: "))
TestLocation = raw_input("Test location:")
TestSpecs = raw_input("Test scpecs/ purposes: ")
n = int(raw_input("Estimated amount of Data file:"))

def indicatingLED():
   GPIO.output(pinLED, GPIO.HIGH)
   time.sleep(5)
   GPIO.output(pinLED, GPIO.LOW)
   time.sleep(1)

def CSVwrite():
    with open('%s.csv' %log_date_str, 'wb') as csvfile:
        filewriter = csv.writer(csvfile, delimiter=',', quotechar='|',
quoting=csv.QUOTE_MINIMAL)
        filewriter.writerow(['Sensor 00', 'Sensor mk'])
    while time.time() < t_end:
        val_rw_mk = ADC.read_raw(sensor_mk)
        val_rw_00 = ADC.read_raw(sensor_00)
        filewriter.writerow([str(val_rw_00), str(val_rw_mk)])
        indicatingLED()
        time.sleep(delay)

def TestSpecs_txtWrite():
    datafile = open("%s.txt" %log_date_str,"w+")
    datafile.write("Date and time of measurement:%s" % log_date_str+
"\n")
    datafile.write("Datafile logging interval (seconds): %s" %LogTime
+ "\n")
    datafile.write("Reading delay (seconds): %s" %delay + "\n")
    datafile.write("Test Location: %s" %TestLocation + "\n")
    datafile.write("Test specs/ purposes: %s" %TestSpecs + "\n")
    datafile.write("Test readings, sensors in position, elevator
doesn't run:Sensor00:" + str(test_rw00) + ",     ")
    datafile.write("Sensor mk:" + str(test_rwmk) + "\n")
    datafile.write("-------------------------------------------------
-----"+"\n")
    datafile.close()
```

```
def EndingFile():
    log_date_str = time.strftime("%Y%m%d_%H%M%S")
    endfile = open("EndTime.txt","a+")
    endfile.write("Measurement ends at:%s" %log_date_str+ "\n")

print('Sensor reading started...')+ "\n"
endfile = open("EndTime.txt","w+")
for i in xrange(1,n):
  t_end = time.time() + LogTime
  log_date_str = time.strftime("%Y%m%d_%H%M%S")
  TestSpecs_txtWrite()
  CSVwrite()
  EndingFile()

print('Measurement ended !--------------------------------------------
---')
sys.exit(0)
```

C. Python code (runs on Beaglebone black) for real-time data pushing to Thingspeak

```
import Adafruit_BBIO.ADC as ADC
import Adafruit_BBIO.GPIO as GPIO
import httplib, urllib
import time
import sys
pinLED = 'P9_41'

GPIO.setup(pinLED, GPIO.OUT)

def indicatingLED():
   GPIO.output(pinLED, GPIO.HIGH)
   time.sleep(0.3)
   GPIO.output(pinLED, GPIO.LOW)
   time.sleep(0.2)


sensor_00 = 'P9_33'
sensor_mk = 'P9_35'

ADC.setup()
#GPIO.setup("pinLED", GPIO.OUT)


print('Sensor reading started...')+ "\n"
print('Displacement data logging started...') + "\n"

while True:
    indicatingLED()
    val_rw_00 = ADC.read_raw(sensor_00)
    val_rw_mk = ADC.read_raw(sensor_mk)
    params = urllib.urlencode({'field1': val_rw_mk,'field2':
val_rw_00,'key':'I177A25G8ASD2D4G'})
    headers = {"Content-type": "application/x-www-form-urlen-
coded","Accept":"text/plain"}
    conn = httplib.HTTPConnection("api.thingspeak.com:80")
    conn.request("POST", "/update", params, headers)
    res = conn.getresponse()
    print res.status
```

```
print('Measurement ended !-------------------------------------------
---')
datafile.close()
sys.exit(0)
```

D. Python code for plotting

Linear regression

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error as mse
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

def pl_regres():
    X= datafile[['Count']]
    Y= datafile['BB']
    lm.fit(X,Y)
    plt.title("Regression Plot")
    sns.regplot(X,Y, data= datafile, marker= '.',line_kws={'color':
'red'})
    sns.residplot(X,Y,label='Predicted value',scatter_kws={"marker":
"+", "color":"grey"} )
    plt.title("Learning model fitting ")
    plt.legend(loc='upper right')
    plt.show()
```

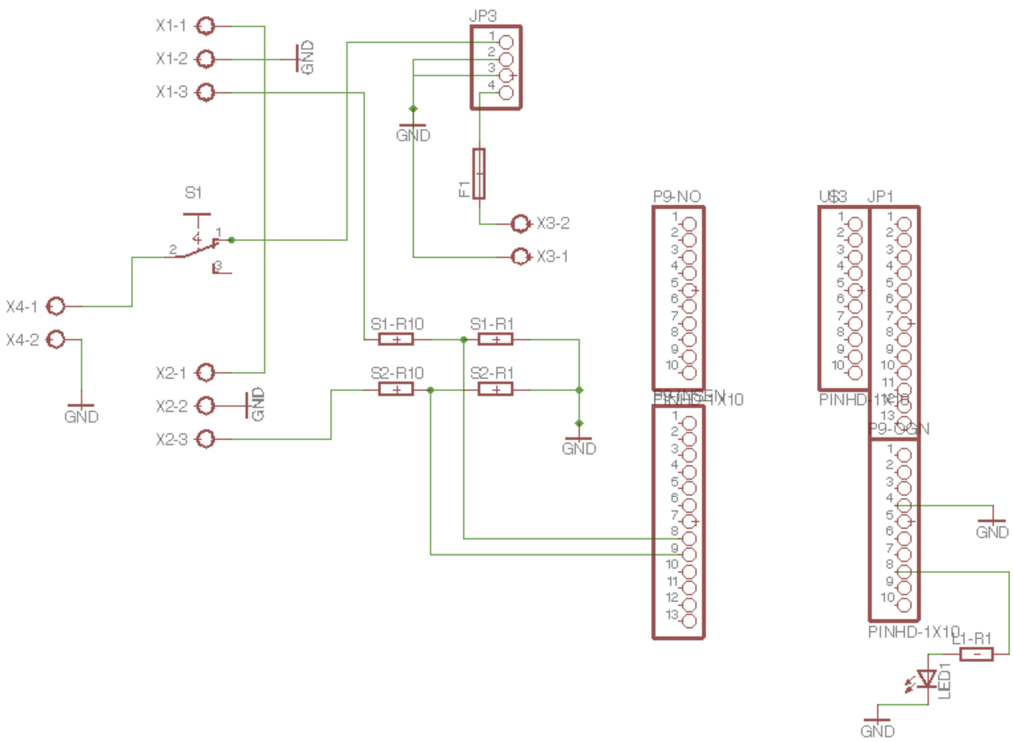
Polyline plotting

E. Python code for Linear regression prediction

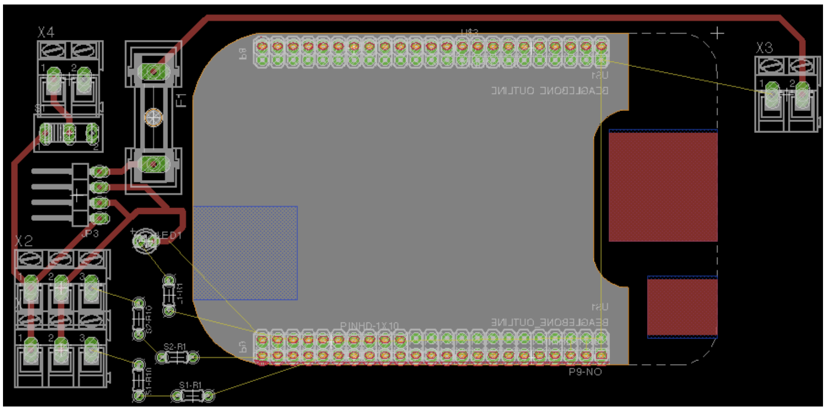
```
def algo_RUL_mean(n,w,k):
    # compute independent variable, count or time in minute
    step= math.ceil(len(yset)/n)
    xi=[]
    for j in range (step):
        xii= n*j+n
        xi.append(xii)
    # compute mean value or dependent variable
    i=0
    maxavg=[]
    minavg=[]
    while i< len(yset):
        yrange= yset[i:i+n]
        subsetmax= yrange[np.argsort(yrange)[-w:]]
        subsetmin= yrange[np.argsort(yrange)[:w]]
        avgma= np.mean(subsetmax)
        avgmi= np.mean(subsetmin)
        maxavg.append(avgma)
        minavg.append(avgmi)
        i+=n
    minavg= np.array(minavg)
    maxavg= np.array(maxavg)
```

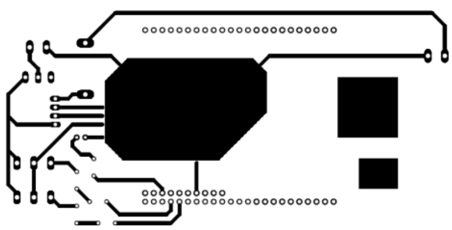
```
    displcmnt_avg= maxavg-minavg


    #Linear regression prediction
    y= np.array(displcmnt_avg).reshape(-1,1)
    x= np.array(xi).reshape(-1,1)
    x_train= x[:-30]
    y_train= y[:-30]
    x_test= x[-30:]
    y_test= y[-30:]

    lm.fit(x_train,y_train)
    #plt.scatter(x,y, color='grey', label='data')
    #plt.scatter(x_train,y_train, marker='.',label='training data')

    #Polynomial line fitting, prediction
    interpol= np.polyfit(xi,displcmnt_avg,2)
    new_dspl_avg= np.polyval(interpol,xi)
    xp= np.linspace(0,50000)
    p= np.poly1d(interpol)
    xp_lin=np.array(xp).reshape(-1,1)

    #print(new_dspl_avg)
    plt.ylim(-0.5,2.5)
    #plt.plot(xi,new_dspl_avg,'g-',linewidth='2', label='Polynomial
Fitting')
    plt.plot(xp, p(xp),linestyle='dotted',color='red', label='Polyno-
mial')
    plt.plot(xi, displcmnt_avg,linestyle= 'solid', linewidth='0.5',
color= 'black', label='Data')
    #plt.plot(xi,maxavg, linestyle= 'solid', color='blue', la-
bel='Max')
    #plt.plot(xi,minavg,linestyle= 'solid', label='Min')

    #SVM fit
    poly_svm.fit(x_train, y_train)
    plt.plot(x_test, poly_svm.predict(x_test), label='Support Vector
Regression')


    plt.plot(xp_lin, svr_lin.fit(x_train,y_train).predict(xp_lin), la-
bel='SVR linear')
    plt.plot(xp_lin, svr_rbf.fit(x_train,y_train).predict(xp_lin), la-
bel='RBF linear')



    plt.plot(x_test,lm.predict(x_test),linestyle=
'dashed',color='blue', label='Linear regression')
    #plt.plot(xp,lm.predict(xp),linestyle= 'dashed',color='blue', la-
bel='Linear regression')

    plt.title('Mean of %s Values, %s Set size' %(w,n))
    plt.axhline(y=2, color='grey',  linestyle='dotted')
    plt.axhline(y=-2, color='orange',  linestyle='dotted')
    plt.xlabel('Minute')
    plt.ylabel('Displacement')
    plt.legend()
    print('Poly prediction x= %s:'%k, np.polyval(interpol,k))
    print('SVR prediction x= %s:'%k, poly_svm.predict(k))
    plt.grid(True)
    plt.show()
```

F. Monitoring prototype schematic



G. Printer curcuit board layout



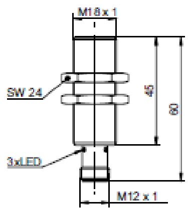


H. Inductive sensor datasheet

Inductive distance measuring sensors

IR18.D08F-11170540

dimension drawing



general data	
mounting type	flush
special type	linearized factor 1
measuring distance Sd	0 ... 8 mm
sensitivity	1,25 V/mm
resolution	< 0,02 mm (stat.) < 0,03 mm (dynam.)
repeat accuracy	< 0,03 mm
adjustment	external Teach-in
teach	1-point analog, 2-point analog, factory reset
linearity error	± 70 µm
temperature drift	± 3 % (Full Scale; S = 0 ... 6 mm) ± 5 % (Full Scale; S = 0 ... 8 mm)

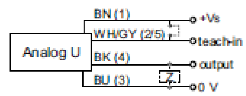
electrical data	
response time (factory characteristic)	< 15 ms
voltage supply range +Vs	12 ... 36 VDC
current consumption max. (no load)	15 mA
output circuit	voltage output
output signal	0 ... 10 VDC
load resistance	> 4000 Ohm
short circuit protection	yes
reverse polarity protection	yes

mechanical data	
type	cylindrical threaded
housing material	brass nickel plated
dimension	18 mm
housing length	60 mm
connection types	connector M12
tightening torque max.	40 Nm

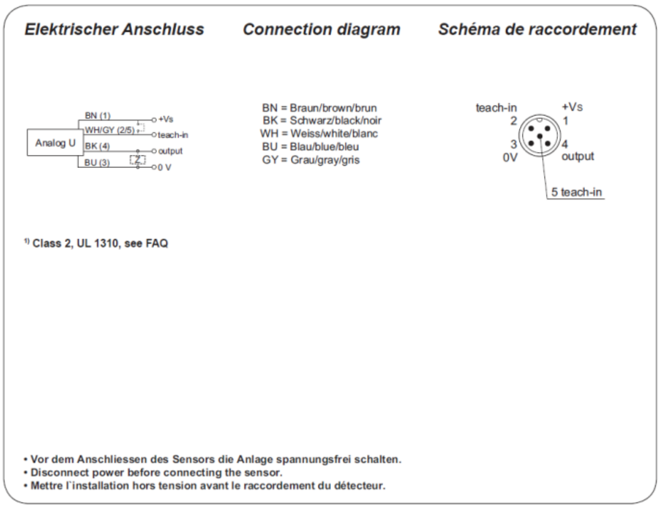
ambient conditions	
operating temperature	-25 ... +75 °C
protection class	IP 67

photo

connection diagram

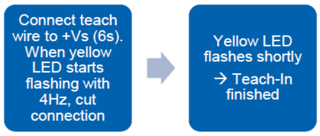


- external Teach-in
- linear analog output
- same measuring distance on all metals (no reduction factor)



Teach Level 3 (Factory Reset)

This will restore the default settings the sensor had when leaving the factory. The default settings are stated in the mounting instructions which are delivered with the sensor.



Teach procedure: Factory reset

If the teach wire is connected to +Vs for more than 10s, the teach procedure is aborted and the sensor keeps its previous settings. This is indicated by an LED flashing frequency of 8Hz.

I. Beaglebone black Revision C specifications

	Feature	
Processor	Sitara AM3358BZCZ100 1GHz, 2000 MIPS	
Graphics Engine	SGX530 3D, 20M Polygons/S	
SDRAM Memory	512MB DDR3L 800MHZ	
Onboard Flash	4GB, 8bit Embedded MMC	
PMIC	TPS65217C PMIC regulator and one additional LDO.	
Debug Support	Optional Onboard 20-pin CTI JTAG, Serial Header	
Power Source	miniUSB USB or DC Jack	5VDC External Via Expansion Header
PCB	3.4" x 2.1"	6 layers
Indicators	1-Power, 2-Ethernet, 4-User Controllable LEDs	
HS USB 2.0 Client Port	Access to USB0, Client mode via miniUSB	
HS USB 2.0 Host Port	Access to USB1, Type A Socket, 500mA LS/FS/HS	
Serial Port	UART0 access via 6 pin 3.3V TTL Header. Header is populated	
Ethernet	10/100, RJ45	
SD/MMC Connector	microSD , 3.3V	
User Input	Reset Button Boot Button Power Button	
Video Out	16b HDMI, 1280x1024 (MAX) 1024x768,1280x720,1440x900 ,1920x1080@24Hz w/EDID Support	
Audio	Via HDMI Interface, Stereo	
Expansion Connectors	Power 5V, 3.3V , VDD_ADC(1.8V) 3.3V I/O on all signals McASP0, SPI1, I2C, GPIO(69 max), LCD, GPMC, MMC1, MMC2, 7 AIN(1.8V MAX), 4 Timers, 4 Serial Ports, CAN0, EHRPWM(0,2),XDMA Interrupt, Power button, Expansion Board ID (Up to 4 can be stacked)	
Weight	1.4 oz (39.68 grams)	
Power	Refer to Section 6.1.7	

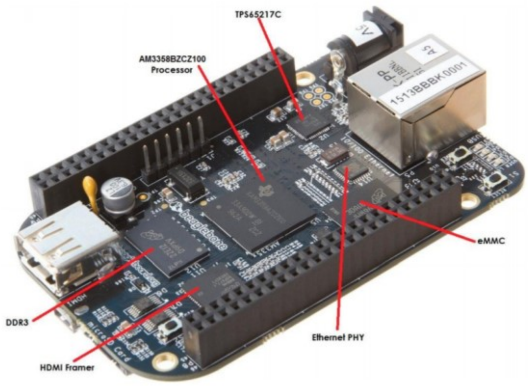
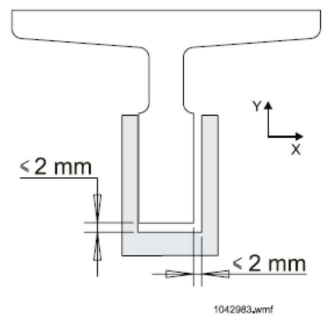


Figure 18. Key Components

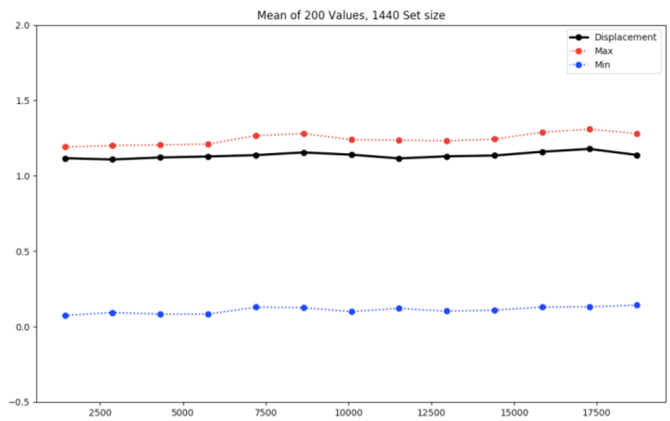
J. Sliding guide shoe maintenance instructions

5.2 Sliding guide shoe

Step	Standard	Action
1	External guide shoe area must be clean.	Clean if dirty.
2	Gaps between guide rail and guide shoe should be maximum 2 mm. Replace the guide shoe when needed. If the gap is over 2 mm in direction X, replace sliding piece according to AS-07.03.005. If the gap is over 2 mm in direction Y (both sides total 4 mm), adjust the gap with shim plates.	
3	There must be sufficient amount of oil in the lubricator.	Check the oil level. Add more oil if needed. See AS-01.01.010 for recommended oils.
4	Guide shoes must not produce noise.	Check the lubricator position. Adjust if needed.
5	Lubricator felt pieces must be in contact with guide rail.	Check at least twice a year. Adjust if needed.

K. Data analysis plots (enlarged)

Figure 30 enlarged



L. Python codes for predictive models and plots

Models

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error as sk_mse
from numpy import array
from sklearn.svm import SVR
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
import csv
from scipy.stats import linregress
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score


def algo_RUL_mean(ele,BB,DD,n,w,k):
    # compute independent variable, count or time in minute
    step= math.ceil(len(yset)/n)
    xi=[]
    for j in range (step):
        xii= n*j+n
        xi.append(xii)
    # compute mean value or dependent variable
    i=0
    maxavg=[]
    minavg=[]
    while i< len(yset):
        yrange= yset[i:i+n]
        subsetmax= yrange[np.argsort(yrange)[-w:]]
        subsetmin= yrange[np.argsort(yrange)[:w]]
        avgma= np.mean(subsetmax)
        avgmi= np.mean(subsetmin)
        maxavg.append(avgma)
        minavg.append(avgmi)
        i+=n
    minavg= np.array(minavg)
    maxavg= np.array(maxavg)
    displcmnt_avg= maxavg-minavg


    # Displacement array/ Formatted data
    y = np.array(displcmnt_avg)  .reshape(-1,1)
    x = np.array(xi)  .reshape(-1,1)

    # training ratio
    ration = math.ceil(0.7 * n)
    x_train = x[:ration]
    y_train = y[:ration]
    x__fit= np.linspace(0,50000,step)
    x_fit= np.array(x__fit)  .reshape(-1,1)

    ## Sklearn Linear regression
    lm.fit(x_train, y_train)
    y_fit= lm.predict(x_fit)
    #Write prediction and stats
    Ylin= lm.predict(k)
    skmse= sk_mse(y_train,y_fit)
    skr2_= r2_score(y_train,y_fit)
```

```
    Intercep= lm.intercept_
    Coef= lm.coef_
    print('SK mse',skmse)
    print('SK r2',skr2_)

    ## Numpy polynomial interpolation
    interpol= np.polyfit(xi,displcmnt_avg,2)
    #model training
    y_nppol= np.polyval(interpol,x_train)
    npmse = np.mean((y_train - y_nppol)**2)
    npr2_ = r2_score(y_train, y_nppol)
    print('NP r2',npr2_)
    print('NP mse',npmse)
    #prediction
    Y_nppol= np.polyval(interpol,k)


    # Last displacement (data recorded)
    ss= displcmnt_avg[-1]
    zz= step*n

    print('SK linear res.:',Ylin)
    print('Numpy poly. inter.:',Y_nppol)
    #print('SK poly.:',Y_skpol)


    return ele, BB, DD, n, w, zz, ss, k, Ylin, skmse, skr2_, Y_nppol,
npmse, npr2_
```

Plotting

```
## Sklearn Linear regression
lm.fit(x_train, y_train)
y_fit= lm.predict(x_fit)
#plotting
plt.plot(x_fit,y_fit,label='Sklearn Linear Res')
plt.scatter(k,lm.predict(k),label='Sklearn Linear Res Prediction')

## Sklearn Polynomial regression
sk_poly= PolynomialFeatures(2, include_bias=False)
x_poly= sk_poly.fit_transform(x_train)
poly_model = make_pipeline(sk_poly,LinearRegression(fit_inter-
cept=False))
poly_model.fit(x_poly,y_train)

x_pred= sk_poly.fit_transform(x_poly)
y_skpol= poly_model.predict(x_poly)
#plotting
plt.plot(x_train, y_skpol,'-', label='Sklearn Polynomial Res, 2nd-or-
der')

## Numpy polynomial interpolation
interpol= np.polyfit(xi,displcmnt_avg,2)
y_nppol= np.polyval(interpol,x_fit)
#plotting
plt.plot(x_fit,y_nppol, label='Numpy Polinomial Interpolation, 2nd-or-
der')
plt.scatter(k,np.polyval(interpol,k),label='Poly Linear Res Predic-
tion', c='green')

## Plot config
```

```
plt.title('Model comparison')
plt.axhline(y=2, color='red',  linestyle='dotted')
plt.axhline(y=-2, color='red', linestyle='dotted')
plt.xlabel('Minute')
plt.ylabel('Displacement')
plt.legend()
plt.show()
```

M.Prediction table

ID	BB	DD	n	w	Last time	Latest Placement	k	Linear Regression	MSE	R2	Polynomial Regression	MSE	R2
KOI	0	**1**	1440	50	18720	1,299	50000	1,37	0,00298	-6,82	1,61732	0,00011	0,70710
KOI	0	**1**	1440	75	18720	1,26	50000	1,343	0,00278	-6,30	1,49445	0,00014	0,63075
KOI	0	**1**	2880	100	20160	1,23	50000	1,277	0,00042	-0,41	0,78691	0,00019	0,36862
KOI	0	**1**	2880	150	20160	1,18	50000	1,216	0,00059	-0,01	0,44033	0,00035	0,40608
KOI	0	**1**	720	50	18000	1,30	50000	1,345	0,00491	-3,91	1,94721	0,00059	0,41481
KOI	0	**1**	720	100	18000	1,23	50000	1,280	0,00481	-4,27	1,90778	0,00049	0,46080
PAS	0	**1**	1440	50	11520	0,82	50000	0,598	0,04115	-6,81	-6,31204	0,00328	0,37794
PAS	0	**1**	1440	75	11520	0,72	50000	0,469	0,05028	-8,25	-6,65532	0,00317	0,41644
PAS	0	**1**	2880	100	11520	0,91	50000	0,662	0,03176	-14,36	-7,68007	0,00006	0,97168
PAS	0	**1**	2880	150	11520	0,83	50000	0,562	0,03646	-14,93	-8,29336	0,00001	0,99533
PAS	0	**1**	720	50	10800	0,82	50000	0,425	0,07055	-0,99	5,30959	0,03432	0,03182
PAS	0	**1**	720	85	10800	0,69	50000	0,145	0,10908	-2,26	3,94893	0,03170	0,05122
KOI	**1**	0	1440	50	18720	1,09	50000	1,135	0,00039	-0,29	0,35793	0,00013	0,57151
KOI	**1**	0	1440	75	18720	1,07	50000	1,119	0,00040	-0,76	0,59205	0,00014	0,40209
KOI	**1**	0	2880	100	20160	1,06	50000	1,062	0,00089	-0,80	0,08376	0,00007	0,86155
KOI	**1**	0	2880	150	20160	1,04	50000	1,043	0,00071	-0,80	0,26583	0,00012	0,70674
KOI	**1**	0	720	50	18000	1,09	50000	1,144	0,00132	-2,91	0,86890	0,00024	0,29869
KOI	**1**	0	720	85	18000	1,07	50000	1,125	0,00147	-3,14	0,98491	0,00026	0,28163