Victor Zhidchenko

METHODS FOR LIFECYCLE SUPPORT OF HYDRAULICALLY ACTUATED MOBILE WORKING MACHINES USING IoT AND DIGITAL TWIN CONCEPTS

ACTA UNIVERSITATIS LAPPEENRANTAENSIS 872

Victor Zhidchenko

METHODS FOR LIFECYCLE SUPPORT OF HYDRAULICALLY ACTUATED MOBILE WORKING MACHINES USING IoT AND DIGITAL TWIN CONCEPTS

Dissertation for the degree of Doctor of Science (Technology) to be presented with due permission for public examination and criticism in the Auditorium 1318 at Lappeenranta-Lahti University of Technology LUT, Lappeenranta, Finland on the 25th of October, 2019, at noon.

Acta Universitatis
Lappeenrantaensis 872

Supervisors Professor Heikki Handroos
LUT School of Energy Systems
Lappeenranta-Lahti University of Technology LUT
Finland

Professor Alexander Kovartsev
Department of Software Systems
Samara National Research University
Russian Federation

Reviewers Professor Asko Ellman
Automation Technology and Mechanical Engineering
Faculty of Engineering and Natural Sciences
Tampere University
Finland

Assistant professor Tatiana Minav
Automation Technology and Mechanical Engineering
Faculty of Engineering and Natural Sciences
Tampere University
Finland

Opponents Professor Asko Ellman
Automation Technology and Mechanical Engineering
Faculty of Engineering and Natural Sciences
Tampere University
Finland

Assistant professor Tatiana Minav
Automation Technology and Mechanical Engineering
Faculty of Engineering and Natural Sciences
Tampere University
Finland

ISBN 978-952-335-426-5
ISBN 978-952-335-427-2 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491

Lappeenranta-Lahti University of Technology LUT
LUT University Press 2019

Abstract

Victor Zhidchenko
Methods for Lifecycle Support of Hydraulically Actuated Mobile Working Machines Using IoT and Digital Twin Concepts
Lappeenranta 2019
126 pages
Acta Universitatis Lappeenrantaensis 872
Diss. Lappeenranta-Lahti University of Technology LUT
ISBN 978-952-335-426-5, ISBN 978-952-335-427-2 (PDF), ISSN-L 1456-4491, ISSN 1456-4491

The Internet of Things and Digital Twins are popular but quite distant concepts nowadays. Many problems arise along the efforts of combining these concepts. This research work focuses on these problems and considers the possible solutions on the example cases of two applications for lifecycle support of hydraulically actuated mobile working machines: remote surveillance and fatigue life estimation.

The main contributions of this dissertation include: determination of the main problems of running physics-based digital twins in the Internet of Things environment; the methods for remote surveillance and fatigue life estimation of hydraulically actuated mobile working machines based on simulation model of the machine; the structure for software systems implementing the developed methods; the techniques that facilitate the implementation of digital twins in the Internet of Things environment.

Keywords: simulation, dynamics, hydraulics, fatigue, Internet of Things, Digital Twin

Реферат

Жидченко Виктор Викторович
Методы сопровождения жизненного цикла машин с гидравлическим приводом с использованием технологий Интернета вещей и цифровых близнецов
Лаппеенранта 2019
126 страниц
Acta Universitatis Lappeenrantaensis 872
Diss. Lappeenranta-Lahti University of Technology LUT
ISBN 978-952-335-426-5, ISBN 978-952-335-427-2 (PDF), ISSN-L 1456-4491, ISSN 1456-4491

Активно развивающиеся в настоящее время концепции Интернета вещей и Цифровых близнецов остаются обособленными друг от друга. Их интеграция сопряжена с рядом трудностей, преодолению которых посвящена настоящая работа. Рассмотрены две области применения, интегрирующие указанные концепции в задаче сопровождения жизненного цикла машин с гидравлическим приводом: наблюдение за удаленными объектами и оценка циклической долговечности машин.
Положения, выносимые на защиту: идентификация основных проблем реализации цифровых близнецов, построенных на основе структурных математических моделей, в инфраструктуре Интернета вещей; методы наблюдения за удаленными объектами и оценки циклической долговечности машин, основанные на применении математических моделей объектов; структура автоматизированной информационной системы, предназначенной для реализации предлагаемых методов; методики реализации цифровых близнецов в инфраструктуре Интернета вещей.

Ключевые слова: математическое моделирование, динамика, гидравлика, усталость материала, Интернет вещей, Цифровые близнецы

Acknowledgements

This work was carried out in Laboratory of Intelligent Machines of Department of Mechanical Engineering at Lappeenranta-Lahti University of Technology LUT, Finland and in Department of Software Systems at Samara National Research University, Russian Federation, between 2017 and 2019.

I would like to express my thanks to all the people who inspired me to do this work and supported me during the years while I was working on this thesis.

I would like to thank my supervisor, Professor Heikki Handroos, for his willingness to create collaboration between the Finnish and Russian universities and an opportunity for me to make a research work resulted in this thesis. His experience, knowledge, and good judgment provided the right mix of freedom, support, and timely advice.

I would like to express my gratitude and appreciation to my teacher and also a supervisor of this thesis, Professor Alexander Nikolaevich Kovartsev from Samara National Research University. He gave me a bright example of what a real scientist should be, and he has been inspiring me for research activities during many years.

I thank the dissertation reviewers, Assistant Professor Tatiana Minav and Professor Asko Ellman from Tampere University, for their time, their effort and expertise that they contributed to reviewing, and for their valuable comments that helped me to improve the manuscript.

I would like to express my gratitude to all my teachers from Samara National Research University whom I was happy to know and learn from. It is your merit that this multidisciplinary research was able to be performed. I thank my colleagues from Software Department of Samara National Research University for their support and particularly for their remote help with administrative tasks, while I was in Finland. I would like to thank the administration of Samara National Research University for the financial support that helped this research work to be carried out, and especially Dean of Faculty of Information Technology Eduard Ivanovich Kolomiets for his active participation in establishing the collaboration between the two universities and for his support of my activities.

Thanks to all the administrators and the staff members of LUT University for creating and maintaining that marvellous environment for research and study. I would also like to appreciate the personnel of the Doctoral School of LUT University for guidance and support. Sari Damsten was the first person my family and I met when we arrived in Lappeenranta. Her kindness has become a trademark of Lappeenranta and LUT University for us.

I am grateful to Professor Aki Mikkola. One of his earlier works was used as a reference in the current research. I would like to thank Grzegorz Orzechowski for the valuable discussions and comments given by him on the early stage of the research.

I wish to express special thanks to the researchers of the Laboratory of Intelligent Machines for sharing their quiet and productive working environment. Warm thanks go to Hamid Roozbahani who introduced the LUT University to me setting the beginning of the entire story, and who supported me during my stay in Finland.

I would also like to express my appreciation to the administration and all the citizens of Lappeenranta for the warm welcome and comfortable accommodation of our family.

Most of all, I wish to thank my family. My parents, my wife Ekaterina and my sons Iaroslav and Elisei, your love, your care and your support give me the power for all my achievements. Thank you and sorry for all the time I was not with you, while working on this dissertation. I dedicate it to you.

Victor Zhidchenko
September 2019
Lappeenranta, Finland

To my family

Contents

Nomenclature

In the present work, variables and constants are denoted using *slanted style*, vectors are denoted using **bold regular style**, and abbreviations are denoted using regular style.

Latin alphabet

A	rotation matrix	–
A	area	m^2
B	effective bulk modulus	Pa
Cv	semi-empirical coefficient for the volume flow calculation	$(m^7/kg)^{1/2}$
D	the number of quantization levels	–
d	the quantization step	–
F	force magnitude	N
F	force vector	N
f	frequency	Hz
g	acceleration due to gravity	m/s^2
i	index	–
j	index	–
k_p	flow-pressure coefficient of a pump	m^4s/kg
L	the scale of the measured parameter	–
M	mass matrix	–
m	mass	kg
N	number of objects	–
n	number of objects	–
p	pressure	Pa
Q	volume flow	m^3/s
T	transformation matrix	–
T	time	s
V	volume	m^3
v	velocity magnitude	m/s
v	velocity vector	m/s
W	matrix for the text vectorization in the corpus of documents	–
$\mathbf{W}_{\mathbf{norm}}$	normalized matrix for the text vectorization in the corpus of documents	–
W	network bandwidth	bit/s
w	occurrence frequency of a word	–
$\hat{w}$	normalized value of the occurrence frequency of a word	–
X	vector of coordinates	–
x	distance	m

Greek alphabet

α	angle	rad
β	angle	rad
γ	normalized root mean square error	–

Δ	change of a parameter	–
θ	angle of the crane boom	rad
λ	telemetry parameter	–
π	3.14159265358979323846264338327950 29	–
ρ	mass density	kg/m^3
σ	stress	Pa
τ	time constant	s
χ	the percentage of data loss	%
Ψ	root mean square error	–
ω	angular velocity	rad/s

Dimensionless numbers

C_d	discharge coefficient

Superscripts

T	transpose of a matrix

Subscripts

b	bits
c	constraint
cl	cluster
cond	condition
cyl	cylinder
dtype	data type
dl	discretization with linear polinomial interpolation
dq	discretization with quadratic polinomial interpolation
docs	documents
e	external
eq	equivalent
F	force
fr	frequency
h	header
idf	inverse document frequency
L	link
M	mass
max	maximum
md	measurement data
min	minimum
norm	normalized
p	pump
PDU	Protocol Data Unit
q	quantization

ref	reference
S	supply
s	signal
se	sensor
t	time
tfidf	term frequency - inverse document frequency
Tr	transmission
ts	time step
vars	variables

Abbreviations

2D	two dimensional
3D	three dimensional
3G	third generation of mobile technologies
4G	fourth generation of mobile technologies
5G	fifth generation of mobile technologies
ADC	Analog-to-Digital Converter
CAD	Computer-Aided Design
CoAP	Constrained Application Protocol
CSV	Comma-Separated Values
FEA	Finite-Element Analysis
FEM	Finite-Element Modelling
HCF	High Cycle Fatigue
HLTA	Hierarchical Latent Tree Analysis
HTD	Hierarchical Topic Detection
IIoT	Industrial Internet of Things
INEF	Iterative Newton-Euler Formulation
IoT	Internet of Things
IP	Internet Protocol
LAN	Local Area Network
LCF	Low Cycle Fatigue
LDA	Latent Dirichlet Allocation
LTE	Long-Term Evolution
MQTT	Message Queuing Telemetry Transport
MTU	Maximum Transmission Unit
PC	Personal Computer
PDU	Protocol Data Unit
RNEA	Recursive Newton-Euler Algorithm
SBC	Single Board Computer
SCADA	Supervisory Control and Data Acquisition
SD	Secure Digital
SMA	Simple Moving Average
TCP	Transmission Control Protocol
TLS	Transport Layer Security

UDP	User Datagram Protocol
VPN	Virtual Private Network
XML	Extensible Markup Language

1 Introduction

The evolution of network technologies has led to the opportunity of connecting many devices to the global network and transmitting data with high speed. The number of connected devices can be much bigger than the number of people on Earth. Available data transfer rates allow transmitting large amounts of data in real-time. On this basis the term "Internet of Things" (IoT) has emerged representing the interconnected system of communicating devices. Each device is not a computer but a component of an intelligent system that can contribute as well as consume the data or services leveraging the system and at the same time gaining the benefits unavailable in standalone operation.

At the time of writing, the concept of "Internet of Things" is rapidly developing. There are more than seven billion “things” connected to the Internet and it is expected that by 2025 there will have been more than 20 billion of them. Despite the large number of successful projects in this field, there are several bottlenecks preventing the wide adoption of the concept. The main obstacle is the popularity of centralized data processing facilities implemented as "cloud systems". Gathering and processing the data centrally in large data-centers imposes constraints on computing resources available for each device. For this reason, existing solutions focus on generalized data analysis when the data obtained from many devices are aggregated and their statistics are calculated and analyzed. The term "Big Data" has emerged to describe such kind of processing. The data coming from a single device usually lack complex analysis due to constraints on computational resources. It is saved to the database or simple analysis is applied to it, for example, validating the data by checking the bounds violation.

Another developing area nowadays is Digital Twin. That is a creation of simulation models for different objects and running simulations in real time using the data coming from the objects. These data can represent the measurements made by the sensors located on the object or can be generated by the object itself. In the current work we use the term “Digital Twin” to describe the aforementioned concept and the term “digital twins” to describe the instances of simulation models that are used together with real objects within the concept.

Digital twins allow to better understand the object state, or to control the object more efficiently. The concept has been successfully utilized for years in manufacturing or technological processes control implemented in SCADA systems. With the emergence of IoT the concept gained new popularity as it became much easier to collect sensor data from many kinds of objects.

Digital twins use rather complex simulations that usually do not fit the cloud-based environments. They can be implemented in the cloud for large and expensive objects like a manufacturing plant, a shopping mall or a wind turbine. In these cases, the cost and responsibility of the simulated object allow dedicating much computational power for running simulations. In case of less expensive but numerous and rather complex

objects like mobile working machines the computational resources dedicated per simulated machine should be minimized.

This dissertation addresses the issue of running physics-based digital twins in IoT environment by consideration of two example applications for lifecycle support of hydraulically actuated mobile working machines: remote surveillance and fatigue life estimation.

1.1 Background

Simulation models have been traditionally utilized during a design phase of the product lifecycle. In the manufacturing process of mobile working machines, the modelling reduces significantly the development time by minimizing the need for building and testing the prototypes of the machines being designed. A special case is real-time simulators based on the detailed simulation models of the machines. They allow not only simulating the behaviour of a future machine in different working conditions but also provide an ability to "feel" this behaviour by sitting in the cabin of the machine being designed (Baharudin, 2016). Augmenting the simulation model with physical objects like a seat and control devices used in the virtual environment of real-time simulators has led to the wider use of the term "digital twin", since such use cases join together the real and the virtual worlds.

The models used in the digital twins applied in the design phase require high accuracy of simulation. Their performance is of less significance. When the performance is the issue, for example, in real-time simulators, the powerful computing resources are used. These resources are usually unavailable in other phases of the machine lifecycle, especially in the operation and maintenance phase. It is usually economically unreasonable to maintain a powerful computing resource for each individual mobile working machine while trying to reduce the maintenance and operational costs of the machine using its digital twin.

Running simulations using moderate computing resources concurrently with the machine operation can utilize several approaches. In the tasks of controlling various processes on board the machine the simulation model is usually simplified to the level that presumes an accuracy needed for solving the control tasks but is simple enough to perform calculations in real time (Ellis, 2012). Another way is to obtain simple dependences between the different parameters of a machine. These dependences are used then in the programs created for on-board controllers. An example of such a use case can be found in Khodadadi Sadabadi and Shahbakhti, 2016. The similar approach of finding correlations between several parameters of the machine to estimate its condition on the basis of observable data is used in the on-line monitoring solutions. An ability of statistical methods to find correlations in data streams not taking into account the physical dependencies between the data has led to the popularity of the so called "data-driven" approach (Erikstad, 2017). The most of the currently available on-line machine monitoring systems utilize this approach. They analyse the sensor data being

gathered from the machines in real-time and find the boundary crossings, the trends and the correlation violations. Another approach uses Reduced-Order Modelling to minimize the computational complexity of mathematical models (Lucia et al., 2004). This approach is used by ANSYS company, which holds a leading position in engineering simulation during several decades, to produce fast digital twins of different objects. An example considering the application of the approach in the medical area can be found in Groth et al., 2018.

In the case of creating digital twins for the operation phase of the lifecycle of mobile working machines the most flexible approach is to create custom models taking into account the purpose of the model creation. This approach can also be the most time consuming as it is more reasonable to reuse the models created during the design phase. However, depending on the final purpose of the model, it can be easier to create a new model than try to reduce the computational complexity of the initial high accuracy model. The common tasks during the operation phase of mobile working machines are monitoring the condition and performance of a machine and solving the maintenance problems.

Two aspects of the monitoring task can be considered. First, it as an estimation of machine condition. This task involves gathering the condition sensor data such as location, speed, fuel level, engine status, error codes, and others. These data provide an overview of the machine state and also can be used to solve such tasks as workload estimation in terms of on/off hours or the distance travelled. At the time of writing, there are many existing IoT solutions that provide this kind of monitoring as part of the fleet management systems. An overview of the tasks being solved by the fleet management systems in the maritime transport sector can be found in Lazakis et al., 2016. The description of several fleet management systems used in the mining industry is provided in Moradi Afrapoli, 2018. In Rögnvaldsson et al., 2018, an example of the monitoring system for the city bus fleet is considered.

The second aspect of monitoring task is an ability to watch remotely what the machine actually does. In the current research work this task is referred to as remote surveillance. Traditionally, video streaming has been used for remote surveillance on objects of different type. In applications related to mobile working machines the video surveillance began to be actively applied for the remote monitoring of construction sites in the first decade of the XXI century. With the advancements in computer vision and machine learning techniques it has become possible to derive new information from the video data automatically (Ding et al., 2018; Fujitake and Yoshimi, 2017). Nonetheless, video streams produce large volumes of data, require high-speed network connections and large data storage resources. Video surveillance is reasonable in the responsible applications such as public safety and security tasks but may be too resource consuming for the applications related to mobile working machines. The quality of video data depends on the lighting conditions, the presence of dust, mist and other obstacles which makes difficult to implement video surveillance in off-road conditions where all mentioned factors are usual. For these reasons, the remote surveillance based on digital

twins can be more suitable in many applications related to mobile working machines. For example, the machines working in a quarry or in a mine, where such common conditions as the darkness, rain, and snow make video surveillance inefficient can be monitored using kinematic or dynamic simulation models of the machines.

The kinematic models are suitable in the cases when a machine has large and relatively slow-moving mechanical structure. Example applications can be seen in monitoring tower and gantry cranes (Zhong at al., 2014; Fang, 2016).

Using the dynamic models, it is possible to more accurately reproduce movements performed with accelerations and with complex trajectories. What is more important, dynamic models provide new information which is unavailable in kinematic models – the information about the forces acting in the machine. This information makes additional analysis possible. For example, the productivity of the machine can be more accurately estimated, the reason of a failure can be more reliably identified, and predictions can be made on the wear of the machine components. These are the tasks related to the maintenance of the machine.

Several maintenance strategies exist. They have been developed concurrently with the evolution of technology (Prytz, 2014). The oldest and widely used one is the reactive (also referred to as corrective) maintenance strategy. It allows the failures to occur, then the faults are detected and fixed. In some application areas this strategy is very costly or unacceptable. In these areas the traditional approach was to use the preventive maintenance. The equipment is periodically inspected, and the remedial actions are taken on it in order to prevent the possible failures. The drawback of this strategy is that it usually does not account for the actual state of the equipment. It can result in extra costs caused by redundant inspection or by the replacement of parts that do not need replacement yet. The failures may also occur between the scheduled maintenance intervals. To eliminate these drawbacks, the predictive maintenance strategy has emerged. It assumes the periodic or continuous monitoring of the actual condition of equipment and scheduling the maintenance activities in accordance with the prediction of its future state. With the development of information and communication technologies and data analytics, the prescriptive maintenance strategy has become possible and has attracted attention of the research community (Diez-Olivan et al., 2019). This strategy assumes the utilization of the data processing techniques that allow suggesting the actions to be made in order to optimize both the equipment reliability and the maintenance costs (Bokrantz et al., 2017).

The Digital Twin concept contributes to the development of the predictive and prescriptive maintenance by introducing the models capable of estimating the unobserved parameters of the equipment and predicting their future behaviour. The current study considers the estimation of fatigue phenomena in the mechanical structure of a machine.

In the responsible and expensive applications, for example, in the military, aerospace and construction segments, extensive measurement programs and high accuracy simulations are implemented for studying the fatigue phenomena in the produced machines. Viitanen and Siljander, 2019 provides a clear picture of the volume of work performed in this filed in the aviation industry. The similar approach could not be utilized in the case of mobile working machines manufacturing due to economical reasons. A common practice is to estimate the fatigue life on the design phase of the machine lifecycle (Będkowski, 2014). The mechanical structure of the machines is usually designed with high degree of redundancy in order to provide the fatigue life of the machine components that is much longer than the lifetime of the machine. Nonetheless, the failures caused by the fatigue phenomena happen with mobile working machines (Rakin et al., 2013; Danicic et al., 2014; Richard et al., 2008). Manufacturers face the need for the global market competitiveness, customization of products according to user requirements, compliance with international standards concurrently with the reduction of costs and time to market. This leads to the shortening of the design phase, reduction of weight and using various materials. Fatigue-related issues become more important in such conditions. One of the measures of preventing fatigue failures could be on-line monitoring and prediction of fatigue phenomena in the operation phase of the machine lifecycle.

The current study presents a method for the fatigue life estimation of hydraulically actuated mobile working machines. The method uses the forces and accelerations calculated by the dynamic model of the machine to produce a load history and a stress history of the components of the machine mechanical structure. The load history is calculated concurrently with the machine operation by obtaining the sensor data within the IoT environment. The calculated stress history is used to estimate the fatigue life of the machine.

For the monitoring and maintenance tasks the models of machine dynamics should be created. A straightforward approach is to use fast dynamic formulations to describe the machine dynamics. The iterative formulations are known to be the fastest of different approaches used in multibody dynamics. They provide O(N) computational complexity which means that the number of computing operations needed to simulate the multibody dynamics depends linearly on the number of bodies in the model.

In (Zhidchenko et al., 2018) and (Malysheva et al., 2018) the Iterative Newton-Euler Formulation (INEF) was tested on the example case of building a digital twin for the hydraulically actuated mobile crane. In (Zhidchenko et al., 2018) the capability of INEF to predict the motion of the crane in real time on the basis of its dynamic model was studied. Two models were created: the dynamic model based on INEF and the simulation model created in commercially available software MATLAB/Simulink. It was found that both models provide the simulation speed that allows predicting the motion of the crane in real-time. The INEF-based model was more than six times faster than its counterpart. It was fast enough to be executed on the low-cost mobile microcomputer Orange Pi 2G-IOT based on ARM platform and compatible with

Raspberry Pi. Further experiments showed that the model created with MATLAB/Simulink was not able to run on Orange Pi in real time since this microcomputer is approximately sixteen times slower than the PC used in (Zhidchenko et al., 2018). In this paper the author was responsible for the design and software implementation of the dynamic model based on INEF. The program has been developed using C programming language that was compiled and run on different hardware platforms. The author was also responsible for making experiments with INEF-based model, gathering and processing the experimental results and for writing the major part of the paper.

In (Malysheva et al., 2018) the dynamic models of the same mobile crane were extended with the model of its hydraulic system. An additional dynamic model of the crane was created with the use of Open Dynamics Engine software library. The three models were tested for the ability to predict the motion of the crane in real time while taking into account the hydraulic system of the crane. The model created in MATLAB/Simulink showed the lowest performance and was able to run in real time using the simplest solver (ode1, Euler's Method) only. The model created with the Open Dynamics Engine showed a similar performance. The INEF-based model was the fastest one and was able to simulate a time interval of the crane motion in nearly a quarter of the period of time being simulated. Further experiments have shown the ability of running INEF-based multibody dynamics simulation together with modelling hydraulics of the crane in real time on the Orange Pi microcomputer system. In this paper the author was responsible for the design and software implementation of the dynamic models using INEF and Open Dynamics Engine. The programs have been developed using C programming language that were compiled and run on different hardware platforms. The author was also responsible for making experiments with the models, gathering and processing the experimental results and for writing a substantial part of the paper. The research work described in the paper was presented by the author at the 2018 Global Fluid Power Society PhD Symposium (GFPS-2018, 18-20 Jul 2018, Samara, Russia).

The works described above have demonstrated a possibility of implementing fast digital twins of the mobile working machines using relatively slow computational resources. This result is important for the cloud-based IoT platforms where the fraction of computational power available per simulated object is small because of the large number of objects processed by the IoT platform. Another target application of this result is implementing digital twins on board the machines using mobile microcomputer platforms like Raspberry Pi.

The current work is dedicated to a broader investigation of problems that arise while running physics-based digital twins in IoT environments and to the search for other applications of the digital twins in the area of mobile working machines utilization. Two methods for lifecycle support of hydraulically actuated mobile working machines have been developed in the current work: the remote surveillance and fatigue life estimation methods. They use the kinematic and dynamic models of the machine to reproduce its

motion and to estimate the fatigue degradation of its mechanical structure using the sensor data received remotely using IoT environment. The methods have been developed on the basis of the results of the aforementioned research papers. The results of current research were presented by the author at the V International Conference on Information Technology and Nanotechnology ITNT-2019 (21-24 May 2019, Samara, Russia) (Zhidchenko et al., 2019a). The presented research work was accepted for publication in the Journal of Physics: Conference Series (Zhidchenko et al., 2019b).

1.2 Research objectives

The present work concentrates on the following research objectives:

- The first objective is to determine the main problems that need to be solved in order to implement physics-based digital twins in the IoT environment and consider the possible solutions;
- The second objective is to develop the methods for lifecycle support of hydraulically actuated mobile working machines targeted at remote surveillance and fatigue life estimation by utilizing the concepts of IoT and Digital Twin.

1.3 Research questions and tasks

1. To study the present state of the Internet of Things concept;
2. To examine the problems of implementing the physics-based digital twins in IoT environment on the basis of example applications for lifecycle support of hydraulically actuated mobile working machines;
3. To define the physics-based models suitable for digital twin implementations in IoT environment;
4. To validate the defined models for digital twin implementation in the IoT environment by implementing the test applications for remote surveillance and fatigue life estimation of hydraulically actuated mobile working machines.

1.4 Research methods

This study uses several research methods. In order to find similar research works, identify the main directions of research in the areas of Internet of Things and Digital Twin, define the most suitable theories and methods to be used in the current research work, a literature study was implemented.

Publications related to the Internet of Things were analysed using automatic text analysis techniques. 958 articles returned as a result of the search query containing the word "IoT" were downloaded from the openly accessible, moderated repository for scholarly articles arXiv.org. The content of these articles was then vectorized using the tf-idf method, word stemming and a vocabulary of stop-words consisting of the most

common English dictionary words and the words common for the scientific articles (the stems "figur", "fig", "vol", "cid"). These data were clusterized using the K-means algorithm. The resulted clusters were then manually analyzed by reading all or several papers from them in order to discover the topic of the cluster.

Since this work considers the use of digital twins, which are the simulation models, the modelling and simulation were extensively utilized throughout the study. The created models that describe kinematics, dynamics and fatigue degradation of a hydraulically actuated mobile working machine are based on the well-established and most commonly used theories and methods. Iterative Newton-Euler Formulation was chosen for modelling dynamics as a result of the review of the computational complexity of different dynamic formulations. An approach based on the Stress-Life methodology was implemented for estimating the fatigue of a mechanical structure of a machine. The time series of forces acting in the machine which had been calculated during dynamics simulation were transformed into the load history using the inertia relief technique. The load history was converted to the stress history using the finite-element method and the mesh generated from the CAD drawings of the machine components. The rainflow counting method was utilized to extract stress cycles from the stress history. The von Mises method was used to obtain the scalar values of stress and the cumulative damage was calculated using the Palmgren-Miner rule.

The suitability of Iterative Newton-Euler Formulation for the real-time dynamic simulation was experimentally verified. In order to test the applicability of the proposed approach for the remote surveillance on the mobile working machines in the presence of measurement errors the simulations with artificially introduced different levels of sensor accuracy were implemented. To evaluate the proposed methods the proof of concept software systems were designed, developed and tested during the experiments with the models. The work also contributes to the development of instruments for the remote measurement of mobile working machines parameters by considering the issues related to telemetry data transmission in IoT and Digital Twin applications.

1.5 Contribution of the dissertation

The main contributions of this dissertation include:

1. A review of research publications dedicated to the Internet of Things area was made using the automatic text analysis techniques. The review is implemented over the 958 scientific articles published from 2010 till June 2018 and downloaded from the openly accessible, moderated repository for scholarly articles arXiv.org. Using the clustering techniques, the articles were divided into six clusters containing the main topics of research interest in IoT area, and subtopics in each cluster were identified. The evolution of these subtopics over the time was analyzed. The review has shown that by the middle of 2018 the IoT area had reached its maturity state.

2. The main problems of running physics-based digital twins in the Internet of Things environment were determined and the techniques that facilitate a solution of these problems were proposed. Key tasks that must be performed for gathering and analyzing telemetry data in order to get useful results from the physics-based digital twins in the IoT environment were considered. The techniques facilitating the implementation of each task were proposed. It is shown that the use of the IoT environment for gathering input data for the digital twins does not introduce sufficient errors into the simulation process. The accuracy of simulation results is defined by the accuracy of sensors that measure the telemetry parameters.
3. A method for the remote surveillance on the hydraulically actuated mobile working machines based on the simulation model of the machine has been developed. The method uses the digital twin consisting of the hydraulic model of the actuators and the dynamic model of the machine mechanism to calculate the forces acting in the machine and the accelerations of different machine components. The input data for the method are the values of pressure and position of hydraulic actuators obtained by the sensors. It is shown that iterative dynamic formulations allow implementing the method in real-time on board the machine using the inexpensive mobile microcomputers. The method was tested on the example model of a hydraulically actuated mobile crane. The experimental results have shown that using commercially available pressure and position sensors in the IoT-enabled machines it is possible to calculate the forces acting in the machine with the root mean square error of less than 2%.
4. A method for fatigue life estimation of hydraulically actuated mobile working machines based on the simulation model of the machine has been developed. The method uses the forces and accelerations calculated by the remote surveillance method to produce a load history of the components of the machine mechanical structure. The load history serves as input data for the finite-element analysis that calculates the stress history. Using a cycle counting method and a cumulative damage theory the estimate of fatigue life for the machine components is calculated from the stress history. The method was verified on the model of hydraulically actuated mobile crane and the stress history calculation results have been compared with the results of the reference research implemented earlier. The experimental results have shown that the forces calculated with the remote surveillance method using the sensor data transmitted over the internet and the digital twin can be used to produce a load history. The flexibility of the mechanical structure should be taken into account when modeling the machine dynamics in the digital twin.
5. A structure of the software system implementing the developed methods in the Internet of Things environment has been developed. It uses open standards for transmitting and storing sensor data, as well as the open-source software for the data processing including FEA. The experimental results have shown that the components of the software system that implement dynamics simulation can be run either on board the machine or in the cloud environment. The FEA should be

run in the cloud as it uses much more computational and storage resources. The web-interface should be used to visualize the simulation results. The proof of concept software systems implementing the proposed methods for lifecycle support of hydraulically actuated mobile working machines have been developed and tested.

1.6 Thesis outline

The thesis is divided into several chapters with theoretical contents and original results. The first chapter presents an introduction to the area of integrating the concepts of Internet of Things (IoT) and Digital Twin in applications related to mobile working machines. The background research is outlined, the main problems that are being solved and the main applications in this research and business areas are discussed in it. The objectives of the current research work and the contribution of the dissertation are formulated.

Chapter 2 and chapter 3 give more details about the concepts considered in this thesis. Chapter 2 presents an overview of the research publications related to IoT. To get broader review, the author implemented an automated analysis of more than 900 papers contained in an openly accessible, moderated repository for scholarly articles ArXiv.org. Text clustering techniques were applied to this corpus of documents in order to discover the main topics discussed by the research community in the area of Internet of Things. The methods used for the analysis are described and the results of the analysis are presented.

Digital Twin is a newer concept at the time of writing and there are fewer publications on it. The chapter describes the main topics considered in these publications and the more extended overview of the Digital Twin concept is presented in chapter 3. The application of this concept to the mobile working machines is discussed with the presentation of the main methods used for the simulation of different subsystems of the working machines that were used in this research work.

Chapter 4 gives an overview of fatigue life estimation methods. It presents different methodologies used for fatigue estimation and outlines a set of methods used by the author to implement the proposed approach for the fatigue life estimation of mobile working machines.

Chapter 5 presents the main original results of the current research. It describes two methods for the lifecycle support of mobile working machines in the operation and maintenance phase. The first method is presented in subchapter 5.1. It is the remote surveillance on the machines which uses the observable telemetry parameters to obtain the values of the parameters that could not be directly observed. The method uses the sensor data about the pressure and position of hydraulic actuators of a machine. A kinematic model is used to reproduce the motion of the hydraulically actuated mobile working machine that is calculated from the position data of hydraulic actuators. A

dynamic model of the machine is used to calculate the forces acting in the machine. Along with the motion simulation these data provide a full view of machine operation. The actions that should be taken while implementing the proposed method are presented and comprehensively discussed. Different requirements that should be met at each step, e.g. sensor accuracy, a sampling rate, a data transmission speed, are considered and the estimations of the parameters of a system implementing the method are calculated. The subchapter finishes with the proposed structure of the remote surveillance system for mobile working machines and the algorithm implementing the proposed method.

Subchapter 5.2 describes the method for the fatigue life estimation of hydraulically actuated mobile working machines. The method uses the forces and accelerations calculated by the remote surveillance method to produce a load history and a stress history of the components of the machine mechanical structure. The stress history is used to estimate the fatigue life. The main steps comprising the method are described, and an example application of the method to hydraulically actuated mobile crane is discussed. The opportunities for further development of the proposed approach are also considered.

Chapter 6 presents the experimental results obtained with the designed proof of concept software systems implementing the proposed methods. The experiments were made with the test model of hydraulically actuated mobile crane. First, the model is described, and then the experimental results presenting the performance evaluation of the model are discussed. The test environment comprising the developed proof of concept software systems is presented. The remote surveillance system was tested with the different values of sensor accuracy which were artificially introduced. The dependency of the simulation results on the sensor accuracy is discussed. The fatigue life estimation method was tested by comparing the experimental results with the results of the reference research work that has been implemented earlier and investigated the stress levels in the same mobile crane. The thesis finalizes with the conclusions and suggestions for future work.

2 Internet of Things (IoT) Concept

Internet of Things is the concept of building and utilizing a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies (ITU-T, 2012). The concept extends internet connectivity from the area of computing devices into the world of physical devices and everyday objects providing them with the ability to communicate with each other and be remotely monitored and controlled through the internet. Attempts for building such systems are known from the 1980s but the concept itself was born and started its active research and technological development at the beginning of the twenty-first century (Främling et al., 2003). The driving force of it were the technological advances in computing and networking areas.

In the beginning of 2019, there were more than 7 billion devices connected to the internet that were not ordinary computing or communication devices (computers, notebooks, tablets and smart phones). Different predictions exist at the time of writing about the trends of increasing this number but all of them conclude that by 2025 there will be more than 20 billion connected devices.

In the first half of 2019 the concept of IoT is about twenty years old but there is no wide adoption of it. Despite many existing successful applications, in many countries this concept is known basically as an idea but not as an established and widely used technology.

In order to track the development of the IoT concept, discover the directions of research work, the trends and their evolution over time, the review of research papers was performed in this study with the help of automatic text analysis techniques.

2.1 IoT-related publications review using automatic text analysis techniques

Automatic text analysis techniques provide powerful means for knowledge extraction from a large number of documents related to a particular topic. They can be useful for getting an overview of the topic and its main subtopics. They can also be used to track the trends in the development of the topic over time (topic evolution). Despite the huge amount of data accessible through the Internet, solving such kind of tasks is still difficult with traditional tools like search engines. The problem in these tasks is not about search because the user does not know what to search for, especially when dealing with new topics. Rather the problem is about summarization of thematic contents and representation of data in a generalized form providing a “bird's-eye” view on the topic.

There are two groups of methods that are widely used in the topic extraction tasks. The first group is clustering techniques (Steinbach et al., 2000) that split a set of documents

into subsets of similar ones. The similarity can be considered using different metrics but the most common is the use of similar words in the documents. The subsets can be related to the topics of interest that join the documents together. Another group of methods utilizes the probabilistic approach. Application of this approach to the analysis of texts was a popular research area in Machine Learning in the past decade. The most commonly used method is Latent Dirichlet Allocation (LDA) (Blei et al., 2003; Blei, 2012). It represents documents as random mixtures over latent topics, where each topic is characterized by a distribution over words. LDA assumes that topics should be predefined. In order to apply it for topic detection process, several extensions have been proposed, for example, nested Chinese Restaurant Process (nCRP) (Blei et al., 2010), the hierarchical Pachinko Allocation Model (hPAM) (Li and McCallum, 2006), and the nested Hierarchical Dirichlet Process (nHDP) (Paisley et al., 2015). As topics usually constitute a hierarchical structure, the process of building this structure is known as hierarchical topic detection (HTD). Given a corpus of documents, HTD provides a tree of topics with more general topics at high levels of the tree and more specific topics at low levels. The widely used method of HTD implementing the probabilistic approach is hierarchical latent tree analysis (HLTA) (Liu et al., 2014; Chen et al., 2017). This method builds a tree-structured model consisting of observed and latent variables. Each observed variable in HLTA stands for a word. It is a binary variable representing the presence or absence of the word in a document. The latent variables in HLTA are considered as unobserved attributes of the documents associated with topics. The word variables are at the leaves and the latent variables are at the internal nodes of the tree. All latent variables in HLTA are also assumed to be binary. They partition a document collection into soft clusters of documents that are interpreted as topics. Each topic is described by the words that best distinguish it from other topics. These words usually appear with high probability in the documents belonging to the topic and appear with low probability in the documents belonging to other topics. The feature of HLTA is that it uses the presence of words and word co-occurrence in the topic structure derivation instead of the word frequency. The resulting tree structure can be used to observe different relationships between the documents, but it may also require deep exploration of the structure to discover these relationships. For this reason, clustering techniques were used in this study instead of HLTA or other probabilistic approaches as they provide a straightforward way to group similar documents. The resulting groups were analysed manually to explore the subject common to the documents in the group and to study the research papers on this subject.

In this study automatic text analysis techniques were used in order to observe the evolution of research publications concerning the IoT concept over the last several years. For this purpose, the search query was executed on the site arXiv.org which is an openly accessible, moderated repository for scholarly articles in many fields of science (ARXIV, 2018). The search query consisted of a single word "IoT". The result of the query contained 1058 articles from which 958 for the period from 2010 till June 2018 have been successfully downloaded. The distribution of the number of articles containing the word "IoT" over the years is presented in Figure 2.1. This figure shows

that the number of publications concerning IoT area was doubling per year from 2014 till 2017 and the first half of 2018 confirms this trend.

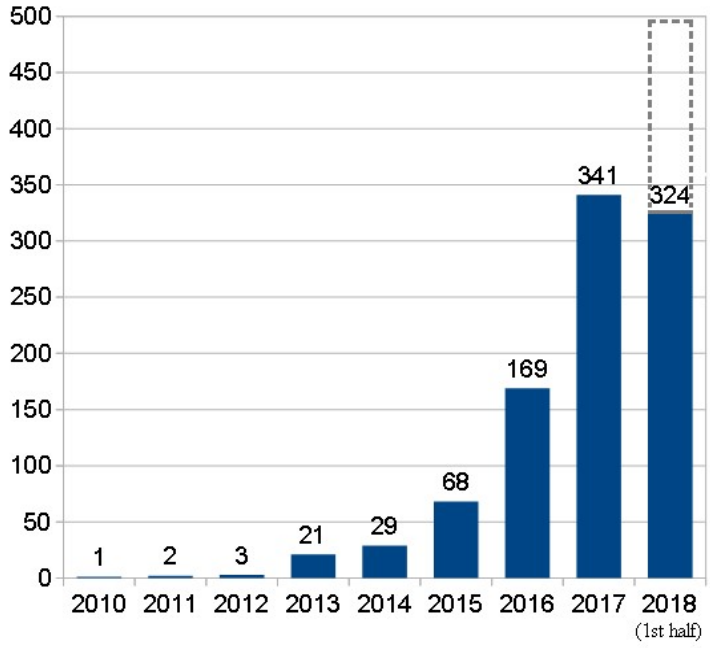

Figure 2.1: Distribution of the articles containing the word "IoT" over the years.

As there was only one article from 2010, this year was excluded from the analysis. The rest of the articles were converted from Portable Document Format (PDF) to plain text format to facilitate their processing.

The goal of the analysis was to find the basic topics of interest in the field of IoT and track their evolution over time. To solve this task the methods of cluster analysis were used. Clustering stands for defining the groups of similar objects given some set of objects. An example of similarity is the Euclidean distance between two points. The points in the group can be defined as similar if the distance between any two of them is smaller than the distance between any of these two points and some other point that does not belong to the group. In order to apply this metric to text data the methods of text vectorization are used. They convert a text into a vector in Euclidean space. The simplest approach is to use the "Bag of Words" ("Bag of n-grams") representation. It considers a text as a set of words and does not take into account the relative position information of the words in the text. In this representation a text is characterized by the word occurrence frequency (that is the number of times each word occurs in a document) but not by the order in which the words appear in the text. With this approach a set of text documents (which is called a "corpus") can be converted to the matrix with one row per document and one column per word:

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N_{words}} \\ w_{21} & w_{22} & & w_{2N_{words}} \\ \vdots & & \ddots & \vdots \\ w_{N_{docs}1} & w_{N_{docs}2} & \cdots & w_{N_{docs}N_{words}} \end{bmatrix} \tag{2.1}$$

Each matrix element w_{ij} in (2.1) is the occurrence frequency of the word j in the document i. If we consider each word as an axis in Euclidean space and the occurrence frequency of this word as a coordinate along this axis, we can represent each text document number i as a vector with the components defined by the row number i in the matrix (2.1). Two documents are considered to be equal if all their coordinates are the same that means that the same words are used in these documents with the same frequency.

In order to reduce the number of different words in the documents, thereby minimizing the dimension of the Euclidean space and increasing the processing speed, several techniques are used. One approach utilizes a dictionary of "stop-words" - widely used words (e.g. "the", "a", "is" in English) that occur in many documents but do not carry meaningful information about the actual contents of the document. These words are eliminated from the corpus prior to processing. Another technique is called "stemming" and represents a process of reducing each word to its invariable form called "stem" by removing morphological affixes. For example, the words "compute", "computer", "computers" and "computing" reduce to the stem "comput". It allows combining different forms of the same word into the common entity.

Further improvement of text vectorization process can be achieved using the tf-idf (term frequency–inverse document frequency) method. This method accounts for the fact that some words are more commonly used than others. It introduces the weight for each word to make rare words more valuable. If a word occurs in every document of a corpus it does not allow distinguishing the documents from each other and hinders clustering. Instead of using the word frequency, tf-idf multiplies it with w_{idf} component, which is computed as follows:

$$w_{idf}(j) = \log\left(\frac{1+N_{docs}}{1+n_{docs}(j)}+1\right) \tag{2.2}$$

where j is the column number of the word in matrix $\mathbf{W}$ from the equation (2.1), N_{docs} is the total number of documents in the corpus, $n_{docs(j)}$ is the number of documents that contain the word corresponding to the column j.

The total weight of the word is calculated as the word frequency multiplied with w_{idf}:

$$w_{tfidf}(i,j) = w_{ij} w_{idf}(j) \tag{2.3}$$

The obtained values of $w_{tfidf}(i,j)$ can be normalized by the Euclidean norm:

$$\hat{w}_{tfidf}(i,j) = \frac{w_{tfidf}(i,j)}{\sqrt{(w_{tfidf}(1,j))^2 + (w_{tfidf}(2,j))^2 + \ldots + (w_{tfidf}(N_{docs},j))^2}} \tag{2.4}$$

This normalization allows balancing the documents of different size. The resulting normalized matrix for a corpus of documents can be represented as follows:

$$\mathbf{W_{norm}} = \begin{bmatrix} \hat{w}_{11} & \hat{w}_{12} & \cdots & \hat{w}_{1N_{words}} \\ \hat{w}_{21} & \hat{w}_{22} & & \hat{w}_{2N_{words}} \\ \vdots & & \ddots & \vdots \\ \hat{w}_{N_{docs}1} & \hat{w}_{N_{docs}2} & \cdots & \hat{w}_{N_{docs}N_{words}} \end{bmatrix} \tag{2.5}$$

where $\hat{w}_{ij} = \hat{w}_{tfidf}(i,j)$.

A lot of clustering algorithms are available (VanderPlas, 2016). One of the widely used algorithms is K-means clustering. With this approach the number of clusters n_{cl} that the data should be divided into must be specified prior to searching them. The K-means algorithm tries to separate data into n_{cl} groups. Each group is described by some mean value which is commonly called the "cluster center" or the "cluster centroid". In general, centroid is not a point from the dataset. It is the arithmetic mean of all the points belonging to the cluster. Each point of the dataset is assigned to the cluster whose centroid is the nearest to the point. The algorithm includes three steps. During the first step n_{cl} centroids are defined for the dataset using some technique. The simplest approach is to choose them randomly, for example as n_{cl} points from the dataset. More sophisticated solutions try to distribute centroids to be distant from each other. Two other steps are executed iteratively. At each iteration the second step assigns each data point to its nearest centroid. At the third step, new centroids are calculated for each group of points by taking the mean value of all of the points assigned to each previous centroid. The distance between the new and the old centroid in each group is calculated and the algorithm repeats the last two steps until this distance is less than a predefined value. In other words, the algorithm repeats until the centroids do not move significantly. The two steps executed iteratively are known as expectation-maximization approach. There are several issues with the algorithm:

1. The number of clusters must be predefined. Depending on the aims of the analysis this number can be adjusted if it is possible to detect the clusters with too dispersed data. An automatic solution of this task is known as silhouette analysis. It is used to determine the separation distance between the resulting clusters. There are alternative clustering methods that can calculate a suitable number of clusters (DBSCAN, mean-shift, affinity propagation) (VanderPlas, 2016).
2. K-means algorithm is highly dependent on the initialization of the centroids. As a result it can converge to the configuration which is not globally optimal. That is why it is common for the algorithm to be run several times with different initializations of centroids.

3. The boundaries between K-means clusters are always linear. The algorithm can be ineffective if the clusters have complicated geometries. One of the possible solutions to this issue is to project the data into a higher dimension where a linear separation is possible.

The clusters that are found with clustering methods can be identified automatically using cluster labelling methods. These methods analyse the common words belonging to the documents in each cluster and try to assign a meaningful label to the cluster that describes its content. This label can be created using the fragments of text cut from the documents. In this work the cluster labelling methods were not used and each cluster was analysed manually.

This study uses the methods described above to analyse the publications on the IoT topic. It was accomplished by using scikit-learn – an open-source set of tools for data mining and data analysis with the Python programming language (Pedregosa et al., 2011). Clustering was performed by KMeans() function from scikit-learn library with initial clusters being assigned with "K-Means++" algorithm (Arthur D. and Vassilvitskii S., 2007). This algorithm initializes the cluster centers to be distant from each other leading to better results than random initialization. The KMeans() function allows defining the number of times the K-means algorithm runs with different centroid seeds. The final results are the best output of consecutive runs in terms of inertia. This number was set to be 10. To provide the input for KMeans() function text vectorization was performed with TfidfVectorizer object of scikit-learn library. This object calculates normalization matrix (1.5) for a corpus of documents. The set of text documents converted from PDF files of the articles described above was used as a corpus. The documents were stemmed before vectorization with the help of SnowballStemmer object provided by the Natural Language Toolkit (nltk) which is a suite of text processing Python libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning (NLTK, 2018). A dictionary of English "stop-words" was utilized to eliminate the words that are frequently used but unimportant for clustering. This dictionary was extended with the words common for scientific writing like "vol", "figure", "fig", "cid". The number of stemmed words used for vectorization was limited to 10000 most frequent stems in the corpus to minimize the vector space and processing time. The TfidfVectorizer object allows defining the upper and lower boundaries for the words frequency in a corpus. These parameters were set to ignore the stems that occurred in more than 95% of the documents comprising the corpus or in less than two documents.

The clustering was performed 4 times with different pre-set number of clusters to form: 9, 12, 20 and 30 clusters. The goal was to define such a number of clusters that provide good separation of topics and at the same time - sufficient size of clusters (in terms of number of documents) in order to be able to track the development of the topic over the time. As the total number of documents in the corpus was around 1000, published during the 5-7 years, to form representative clusters the number of them should be in the range from 10 to 20 with 50-100 documents in each cluster. In the case where the

number of clusters to search for was set to 9 three of them contained 60% of the documents and were supposed to be further refined. The separation to 12 clusters resulted in several well-defined topics but three clusters contained again a half of the documents. With 20 and 30 clusters some of the resulting topics were too detailed and could be joined in a single cluster. The topic of each cluster was defined manually by browsing the content of 5-7 articles from it. The analysis has shown that narrow topics found with large number of clusters could be joined together, but the resulting clusters do not always correspond to the clusters found in experiments with 9 and 12 clusters. As a result, 6 clusters containing main topics of interest were defined. The names of the topics and the total number of documents in each cluster are presented in Figure 2.2.

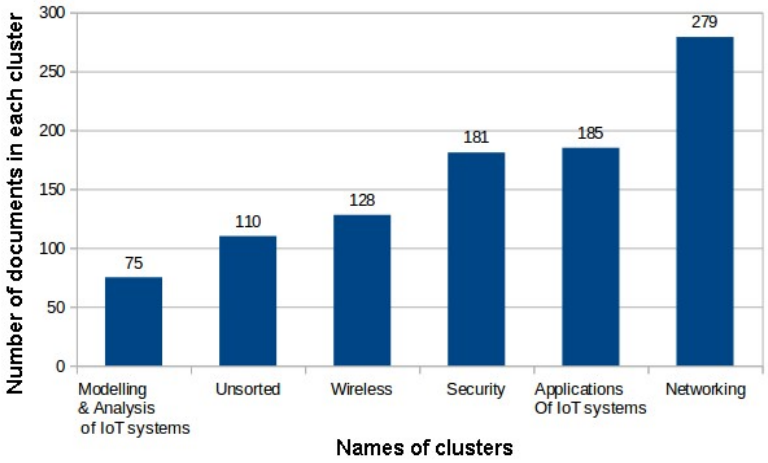

Figure 2.2: Total number of documents in each cluster.

The cluster called "Unsorted" consisted of the documents that were not dedicated to IoT area but contained the word "IoT" and because of that were initially included into the corpus. This cluster also contained the documents that were converted from PDF to plain text incorrectly. Approximately every tenth document in the corpus was placed into this cluster. The distribution of documents among all the clusters is presented in Figure 2.3.

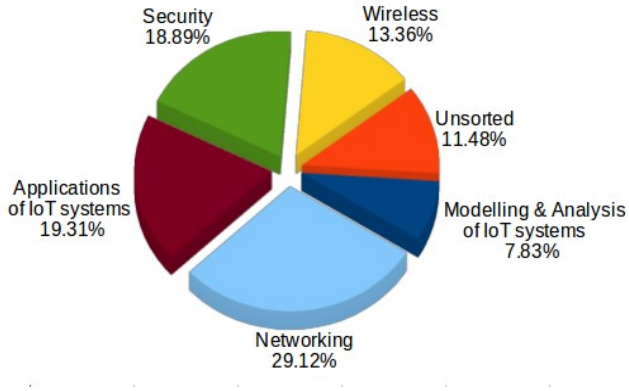

Figure 2.3: Distribution of documents among the clusters.

Figure 2.3 shows that nearly one third of publications in IoT area concerns with different aspects of networking. The wireless technologies could also be related to networking, but they were separated to dedicated cluster as there were several well-defined small clusters concerning different wireless technologies.

Clustering results obtained in the experiments with 20 and 30 clusters allow specification of subtopics in each cluster. Selecting the most frequent stems in each cluster and finding the intersections of the sets of such stems it is possible to define the links between the clusters. The total structure of topics in IoT area discussed in considered corpus of documents could be depicted as presented in Figure 2.4.

The area of each ellipse in Figure 2.4 is proportional to the number of documents in the corresponding cluster. The circular objects combine several clusters into a single topic. They represent a topic itself but not a particular cluster. The ellipses placed close to each other represent close topics as can be seen by the common most frequent stems which are linking these ellipses together. For example, publications concerning power consumption of the wireless network devices can be linked to both the networking and the wireless technologies.

The evolution of topics can be described by the number of documents in each cluster or by the percentage of documents comprising each cluster per year (see Figure 2.5). The percentage of documents dedicated to each topic in the clusters is presented in Figure 2.6. The relative number of documents per cluster remained nearly constant over the years from 2014 till 2018.

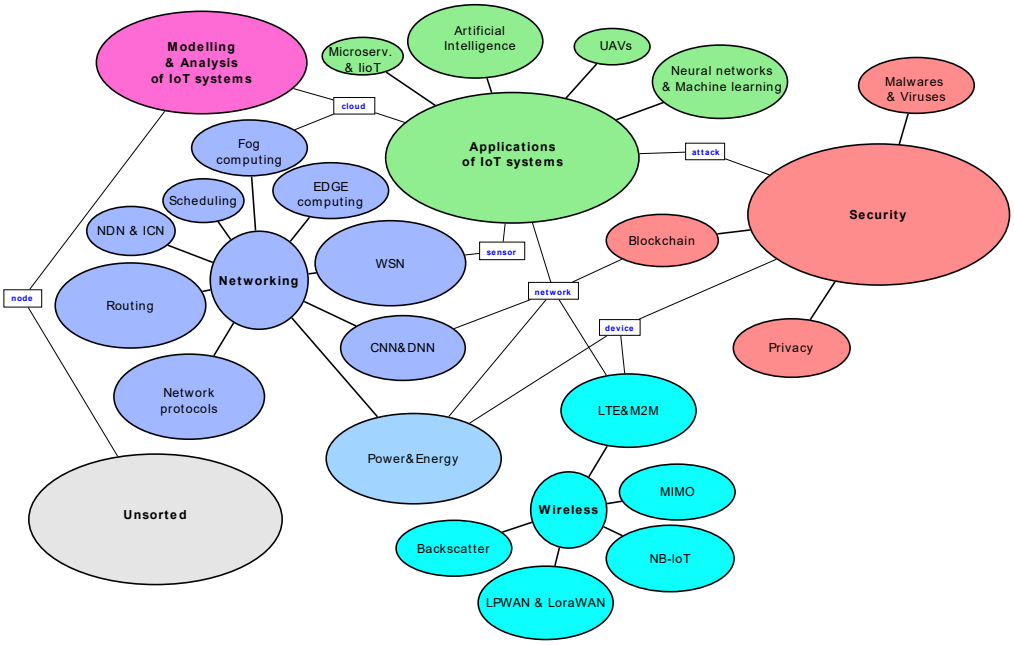

Figure 2.4: The structure of topics in IoT area obtained with clustering techniques.

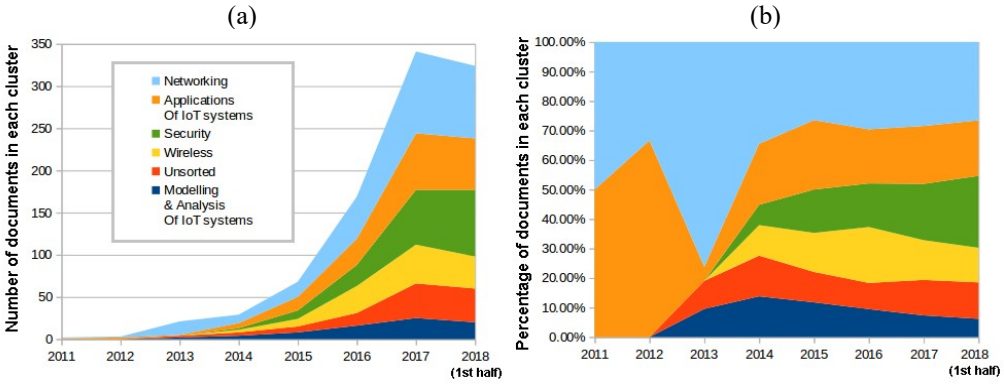

Figure 2.5: The evolution of topics over the years.

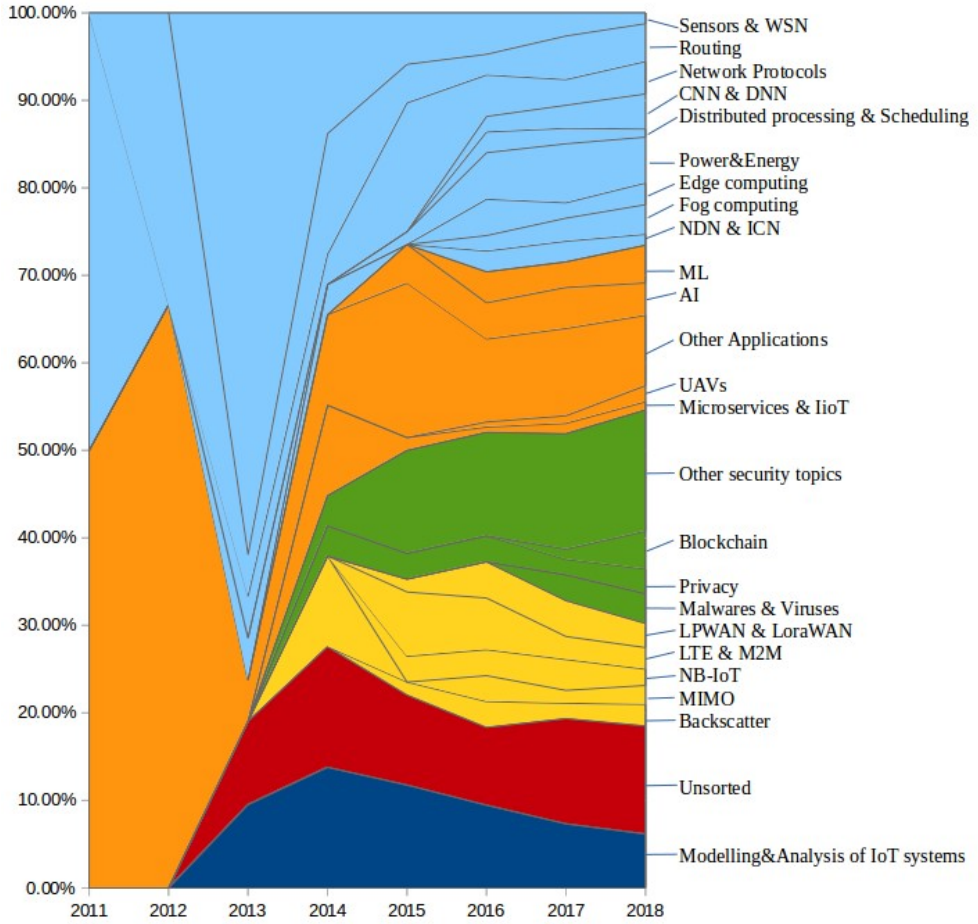

Figure 2.6: The evolution of the percentage of documents per each topic in the clusters over the years.

Two significant trends can be noticed. The first one is the increase of the number of documents concerning security of IoT systems. For example, only in the first half of 2018 there were more publications on this topic than in the whole year of 2017. The second trend is the decrease of relative number of documents dedicated to modelling and analysis of IoT systems. On the early stage of IoT development in 2012-2014 with the lack of well-established technologies a lot of research was targeted to finding the ways of building IoT systems. At the same time, the largest number of research papers was dedicated to networking issues in order to find the appropriate means for connecting a plenty of things to the IoT systems. Five years later when a lot of networking protocols, technologies and software frameworks were developed, the focus of research shifted to the more practical issues like providing security for these systems. It can be stated that in 2017-2018 the IoT area has reached its maturity or has approached it very closely.

Compared with IoT, Digital Twin is a more recent topic at the time of writing. There are only 4 papers dated before 2019 in the results of the search query “digital twin” at arXiv.org. The same query in Web of Science returns 256 research publications dated to the years 2010–2018 in contrast to 28244 publications returned by the query “IoT” for the same period. These 256 publications demonstrate significant growth in the number of published papers each year with 8 papers for 2010–2015, 14 papers for 2016, 56 papers for 2017, and 178 papers for 2018.

At the time of writing there are no results for the search query containing the names of both concepts “IoT” and “Digital Twin” at arXiv.org. Web of Science returns 23 results on the same query dated to the years 2016-2018. About 70% of these papers (16 papers of 23) have been published in the second half of 2018. This fact approves that integration of IoT and Digital Twin concepts is a hot topic in 2018–2019. The subjects covered by the papers are different. 11 papers examine Industrial IoT applications and problems arising in the Industry 4.0 and Smart Manufacturing areas. 3 papers relate to Smart Cities. Other papers consider various applications of digital twins and IoT in healthcare, augmented and virtual reality, smart cars, present general frameworks for the integration of both concepts and demonstrate the work of these frameworks on simple examples. None of the mentioned papers consider the applications of the concepts to mobile working machines. Although such applications have similar properties with other tasks involving the processing of online real-time data, implementation of physics-based digital twins in IoT environment introduces additional problems. The following chapter considers the application of Digital Twin concept to mobile working machines. The problems arising in the implementation of such applications as well as solutions proposed to these problems in the current work are discussed in Chapter 5.

3 Digital Twin concept

3.1 Emergence of the Digital Twin Concept

The Digital Twin concept provides the tool for improving the features of various systems by analyzing the output of the system's model – digital twin. The use of simulation models is not a new idea but the capability of feeding them with the real-time data concurrently with the actual system operation has recently become a very popular approach. This capability arises on the basis of the advantages of networking and computing speed. The wide and natural adoption of the concept is through the Internet of Things. At the time of writing the terminology around the Digital Twin concept is not well established and different sources define the term "Digital Twin" differently (Negri et al., 2017). Many applications of the digital twins have been recently developed (Grieves and Vickers, 2017). They can be divided into the two large areas: analysis of very complex systems (like plants or transportation systems) and real-time analysis of relatively small systems (like a vehicle or a human body). The first area deals with a huge number of processes that take place in a large and complicated model. This approach is very close to the well-known simulation modelling. The second approach uses simpler models but often requires real-time interactions. It is sensitive to the networking speed and computational speed.

3.2 Symbiosis of IoT and Digital Twin Concepts

Digital twin is a simulation model of a real object that exists concurrently with the object, gathers the sensor data from it, and simulates the processes occurring in the object. The simulation results are presented to the user or consumed by the object itself for solving control or optimization tasks. This scenario is well-known and has been utilized for years in telemetry systems in industrial automation, aerospace and military applications. With the expansion of IoT this practice has become available for smaller objects like cars, trucks and other equipment.

Traditional telemetry systems use quite simple post-processing of telemetry data. They utilize the models customized for each particular application or an object that link the telemetry parameters with other characteristics of the object that cannot be observed directly. Current advantages in networking, hardware and numerical methods made it possible to implement a more general approach. IoT environment provides the means for delivering telemetry data from objects of any kind which can be located nearly everywhere on Earth through the Internet. Modern computing capabilities make it possible to run physics-based simulations with real-time speed and obtain the simulation results about different types of processes from complex digital twins.

3.3 Application of Digital Twin Concept to Mobile Working Machines

The following processes could be discovered in a mobile working machine using physics-based simulations in its digital twin:

- Machine *kinematics*: that is a position of each element of the machine that can be calculated and, in some cases, predicted knowing the position of some of these elements. The modelling derives information from position sensors without taking into account the forces acting in the machine. The motion of an empty or fully loaded machine is nearly similar in a kinematic analysis. This is the simplest type of modelling. It consists of geometric calculations that are not time-consuming and are often used in existing solutions.
- Machine *dynamics*: a dynamic model considers the forces that actuate the machine parts. This kind of model distinguishes the loaded motion from the empty one as the forces in both cases differ greatly from each other. This model can be used to calculate many values: the mass of the load, the force applied by the machine to other objects, the work done by the machine, the energy consumed, the stress in the machine's mechanical structure. The position of any part can also be calculated and predicted more precisely. Dynamic modelling provides valuable information for working machines while their job is usually to produce mechanical work. This type of modelling is more complex than kinematics and is usually performed during the design phase using high-performance computers.
- *Thermodynamics* i.e. the process of energy exchange between physical systems as heat and work. This kind of modelling deals with the efficiency, environmental pollution, wear and lifetime of different parts of a machine.
- *Electrodynamics* that links electrical charges and currents acting in the electric system of a machine. With the broader deployment of hybrid and electric vehicles, electrodynamic models will play an important role in their monitoring, predictive maintenance, improvement of their efficiency and productivity.
- *Acoustic phenomena* often represented by vibration of different machine parts. The vibrational analysis is used in modern IIoT solutions for monitoring of industrial equipment. In applications for mobile working machines it can be used for remote diagnostics and condition monitoring.

In the field of mobile working machines kinematics (i.e. position, velocity and acceleration) and dynamics (i.e. forces and torques) of a machine are or great interest. Controlling or monitoring the kinematic properties of machine motion allows solving many tasks, such as predicting machine motion and preventing collisions and overturns,

estimating the productivity, decreasing the time and resources needed for implementing particular work. Calculation of the forces and torques acting in the machine facilitates estimation of its productivity. It allows obtaining the wear and stress that different parts of the machine undergo and helps to solve the predictive maintenance tasks.

Considering the kinematic and dynamic problems in digital twin applications inside the IIoT environment can facilitate the use of other types of models. The reason is that the majority of physics-based models are described by the systems of differential equations. Finding the suitable approaches to solve such systems of equations in IIoT environments for the case of dynamic models this work defines the opportunities for the future implementation of other types of models.

3.4 Simulation of Hydraulically Actuated Working Machines

Fluid power is a common source of energy for mobile working machines. Compared with the other technologies for transmitting mechanical energy (i.e., electric, or pure mechanical systems), fluid power has a clear power to weight ratio advantage (Vacca, 2018). An important advantage of this source of energy is its efficiency in the energy recovery from potential energy which allows reducing the energy consumption of a machine (Minav et al., 2014). Hydraulic actuators like cylinders and motors can be seen in many kinds of machines: cranes, excavators, bulldozers and tractors, loaders and trucks. The motion and the work performed by hydraulically actuated machine can often be fully defined by the motion of hydraulic actuators. For example, in the case of hydraulic crane or excavator the position of hydraulic cylinders defines the position of any part of the machine mechanism except the basis. The position of the basis can be provided by the GPS. The forces acting in hydraulic actuators define the useful work performed by the machine. The position and forces can be calculated from a digital twin of a machine by solving kinematic and dynamic problems.

3.4.1 Machine Kinematics

The operation of mobile working machines is constantly concerned with the location of objects in three-dimensional space. The objects are the parts of machine mechanical structure (booms, arms, links) the load that the machine works with, and other objects in the working environment. These objects can often be described by just two attributes: position and orientation. In order to describe the position and orientation of a body in space, a coordinate system, or frame, is rigidly attached to the object. Describing the position and orientation of this frame with respect to some reference coordinate system, the location of the object can be defined.

Any frame can represent a reference system in which the position and orientation of a body is expressed, so transforming the description of these attributes from one frame to another is a common task.

Kinematics is the science of motion that treats motion without regard to the forces which cause it (Craig, 2005). Within the science of kinematics, position, velocity, acceleration, and all higher order derivatives of the position variables are studied. Hence, defining the kinematics of a machine refers to all the geometrical and time-based properties of the motion.

A mechanical structure of a machine is often considered to be composed of rigid links, which are connected by joints that allow relative motion of neighbouring links. These joints can be instrumented with position sensors, which allow the relative position of neighboring links to be measured. In the case of hinge (or revolute joints), the position in defined by joint angle. The sliding (or prismatic) joints provide the relative translation between links, sometimes called the joint offset. The force or torque applied to some joints can be controlled by actuators.

The number of degrees of freedom that a mechanism possesses is the number of independent position variables that need to be specified in order to define the location of all parts of the mechanism. In the case of mobile working machines, a mechanical structure of a machine is usually an open kinematic chain (for example, in cranes, excavators, loaders, etc.) If each joint position is defined by a single variable, which is the case for commonly used revolute and prismatic joints, the number of joints equals the number of degrees of freedom.

At the free end of the chain of links that makes up a machine mechanical structure is the end-effector (implement). Depending on the type of the machine, the end-effector could be a bucket, a shovel, a grapple, a drill, a harvester head, or another device. The position of the machine mechanism can often be defined by giving a description of the frame, which is attached to the end-effector, relative to the base frame, which is attached to the non-moving base of the machine.

A basic problem in the study of mechanical manipulation is called forward kinematics. This is the static geometrical problem of computing the position and orientation of the end-effector of the machine. Specifically, given a set of joint angles and displacements, the forward kinematic problem is to compute the position and orientation of the end-effector frame relative to the base frame. Solving forward kinematics problem is a common task for discovering the position of a machine from the sensor data obtained from the position sensors of the links.

The problem of inverse kinematics is posed as follows: Having the position and orientation of the end-effector of the machine, calculate all possible sets of joint angles that could be used to reach this given position and orientation. This is a fundamental problem for machine control.

3.4.2 Multibody Dynamics Simulation

Dynamics is a huge field of study dedicated to studying the forces required to produce motion. In order to accelerate a machine mechanism from rest, provide a constant end-effector velocity, and finally decelerate to a stop, a complex set of force and torque functions must be applied by the joint actuators. The exact form of the required functions of actuator force and torque depends on the spatial and temporal attributes of the path taken by the end-effector and on the mass properties of the links and payload, friction in the joints, and so on. One method of controlling a manipulator to follow a desired path involves calculating these actuator torque and force functions by using the dynamic equations of motion of the manipulator. This is a common task in robotics and it constitutes the inverse dynamics problem.

By reformulating the dynamic equations so that acceleration is computed as a function of actuator forces and torques, it is possible to simulate how a mechanism would move under application of a set of actuator torques and forces. This is known as the forward dynamics problem.

Many approaches exist to compose equations linking the forces and accelerations. They are known as dynamic formulations (Haug, 1989; Featherstone, 2008; Shabana, 2010). Existing formulations take their origin in two fundamental approaches known as Newton-Euler formulation and Lagrangian formulation. The first one is based on Newton and Euler laws which relate the motion of the center of gravity of a rigid body with the sum of forces and torques acting on the rigid body. The Newton-Euler formulation might be said to be a "force balance" approach to dynamics. The Lagrangian formulation is an "energy-based" approach to dynamics. The Lagrangian dynamic formulation provides a means of deriving the equations of motion from a scalar function called the Lagrangian, which is defined as the difference between the kinetic and potential energy of a mechanical system. For the same mechanism both formulations give the same equations of motion. The use of particular formulation is defined by the convenience of describing the problem and acquiring necessary arguments. Different formulations have also different computational complexity.

3.4.3 Computational complexity of different dynamic formulations

Computational efficiency is a key point in the field of the dynamics of mechanisms, especially for the robotics community. Many of the most efficient algorithms in dynamics, that are applicable to a wide class of mechanisms, were developed by robotics researchers (Featherstone and Orin, 2000).

A widely used approach to expressing the equations of motion is based on a Lagrangian formulation of the problem (Uicker, 1967; Kahn and Roth, 1971). First algorithms developed using classic Lagrangian dynamics had $O(N^4)$ computational complexity.

The first researchers to develop O(N) algorithms for inverse dynamics for robotics used a Newton-Euler (NE) formulation of the problem. Stepanenko and Vukobratovic developed a recursive NE method for human limb dynamics (Stepanenko and Vukobratovic, 1976), and in Orin et al., 1979, the recursive method was made more efficient by referring forces and moments to local link coordinates for real-time control of a leg of a walking machine. Luh et al., 1980 introduced a very efficient Recursive NE Algorithm (RNEA) by referring most quantities to link coordinates. RNEA is also known as Iterative Newton-Euler Formulation (INEF). The RNEA is the most cited. Hollerbach developed an O(N) recursive Lagrangian formulation (Hollerbach, 1980), but found that it was much less efficient than the RNEA in terms of the number of multiplications and additions/subtractions required in the algorithm. Further gains have been made in efficiency over the years. The results presented in Balafoutis et al., 1988 and He and Goldenberg, 1989 are representatives of those that are up to a factor of 1.7 faster than early implementations of the RNEA (for a 6-DoF robot).

Walker and Orin used the RNEA for inverse dynamics as the basis for efficient algorithms for forward dynamics (Walker and Orin, 1982; Luh et al., 1980). Their Method 3, later named the Composite-Rigid-Body Algorithm (CRBA) in Featherstone, 1987, computed the inertial parameters of composite sets of rigid bodies at the outer end of the manipulator chain. The columns of the inertia matrix were computed efficiently through successive application of inverse dynamics with the joint velocities set to zero, and the joint accelerations set to zero or a unit vector. Since this implies that only one joint is in motion at a time, the inverse dynamics reduces to a simplified analysis of a base set of links in static equilibrium and a composite rigid body in motion at the outer end of the chain. Because of the need to solve a linear system of equations whose size grows with N, the algorithm was O(N^3). For small N, the first-order terms dominated the computation so that the result was more efficient.

The earliest known O(N) algorithm for forward dynamics was developed in Vereshchagin, 1974. This algorithm uses a recursive formula to evaluate the Gibbs-Appel form of the equation of motion and is applicable to unbranched chains with revolute and prismatic joints. The recursive formula was obtained via dynamic programming techniques. This algorithm closely resembles the Articulated-Body Algorithm (ABA), but the paper was way ahead of its time and languished in obscurity for a decade. Later, Armstrong developed an O(N) algorithm for mechanisms with spherical joints (Armstrong, 1979), and then Featherstone developed the ABA (Featherstone, 1983). The first version of this algorithm was applicable to manipulators with single degree-of-freedom joints, but the second included a general joint model and was faster (Featherstone, 1987). In terms of the total number of arithmetic operations required, the ABA was more efficient than the CRBA for $N > 9$. Also, using similar efficient transformations and link coordinates as Featherstone, Brandl et al., 1986 made further improvements on the ABA so that it was roughly comparable to the CRBA for N=6. Further gains have been made in efficiency over the years, with McMillan and Orin, 1995 being representative of those that have reduced the computation (another 15% reduction).

This work uses Iterative Newton-Euler Formulation to calculate machine dynamics in the proposed applications.

3.4.4 Simulation of Hydraulic Systems

The main components of mobile machine hydraulic systems that produce fluid power and convert it to the motion of the machine mechanism are pumps, valves, and actuators. Hydraulic pumps generate power by pressurizing the liquid. The valves, located between the pumps and actuators, control the flow rates and pressure supplied to the actuators. The actuators convert fluid power into the forces applied to the parts of mechanical structure of a machine.

The common types of hydraulic actuators are motors and cylinders. They act as interfaces between the hydraulic circuit and the mechanism. Cylinders are widely used for actuating the links (arms, booms) of heavy equipment. The force produced by a widely-used double-acting cylinder can be described by the following equation:

$$F_{cyl} = p_1 A_1 - p_2 A_2 - F_{fr}(\dot{x}) \tag{3.1}$$

where F_{cyl} is the magnitude of the cylinder force, p_1 and p_2 are the pressures in the cylinder chambers, A_1 and A_2 are the piston side and rod side areas accordingly, F_{fr} is the magnitude of the cylinder friction force which depends on the magnitude of the rod velocity $\dot{x}$. F_{fr} is defined by the friction model being used. Description of different friction models can be found in Keskinen et al. (1993), Andersson et al. (2007).

Hydraulic valves comprise orifices through which the liquid flows. A flow induced by a pressure difference can be described using Bernoulli equation, which assumes the flow to be steady, frictionless and incompressible. Assuming the cross-section of the orifice to be far larger than the cross-section of the port, Bernoulli equation can be transformed into the following form:

$$\begin{aligned} Q &= C_d A \sqrt{\frac{2\Delta p}{\rho}} = C_v \sqrt{\Delta p} \\ C_v &= C_d A \sqrt{\frac{2}{\rho}} \end{aligned} \tag{3.2}$$

where Q is the volume flow, C_d is the discharge coefficient, A is the cross-section area of the valve orifice, ρ is the mass density of the fluid and Δp is the pressure difference between the points, one of which is located before the orifice and the other is located in a sufficient distance after the vena contrata of the orifice. The discharge coefficient takes losses into account and its value is often assumed to be constant in the region of turbulent flow. In practice it is more convenient to consider the C_v coefficient which comprises the parameters that can be obtained using analytical, semi-empirical or empirical approaches that take into account the physical properties of the valve. A

detailed description of semi-empirical models for different types of valves can be found in Handroos (1990), Handroos et al. (1993), Handroos and Halme (1996).

The volume flow of fixed displacement pump can be calculated as a linear function of rotational velocity of the pump shaft:

$$Q_p = \omega_p V_p - Q_L \tag{3.3}$$

where ω_p is the magnitude of the angular velocity of the pump shaft, V_p is the radian volume of the pump, Q_L is the leakage flow. In variable-displacement pump, the radian volume can be continuously adjusted by the displacement controller. Dynamics of variable-displacement pump is usually described by the first-order differential equation:

$$\dot{Q}_p = f(p_{ref}, Q_P, p_p, \tau_p) \tag{3.4}$$

where p_{ref} is the pre-set reference pressure, Q_p is the volume flow of the pump, p_p is the output pressure of the pump, τ_p is the time constant of the pump. For example, the flow supplied by pressure compensated pump can be calculated from the following equation:

$$\dot{Q}_p = \frac{k_p(p_{ref} - p_p) - Q_p}{\tau_p} \tag{3.5}$$

where k_p is the flow-pressure coefficient of the pump.

The presented models of the hydraulic elements are interconnected together using the following continuity equations:

$$\dot{p}_i = \frac{B_i}{V_i}\left(\Delta Q_i - \dot{V}_i\right) \tag{3.6}$$

where p_i is the pressure in i-th volume of the system, B_i is the effective bulk modulus of the volume, V_i is the value of the volume, ΔQ_i is the difference between the input and output flows of the volume. This equation is written for every volume between different components of the hydraulic circuit.

There are several approaches of coupling different domains in multiphysics problems. A comprehensive overview can be found in Keyes et al., 2013 and Gomes et al., 2018. The resulting system (or systems) of differential equations or differential-algebraic equations (DAE) is solved using one of the numerical methods (Haug, 1989; Kovartsev, 2011). The problems that need to be solved while implementing the coupling of mechanical and hydraulic sub-domains in the real-time simulations and the use of different hydraulic models in this kind of simulations are discussed in Esqué et al., 2003. In the current study the equations describing the hydraulic system are incorporated into the system of dynamic equations of the machine.

The dynamic model of the machine provides the values of forces acting on its components and the acceleration of each component. These data can be used to estimate

the machine performance and efficiency, or the loads that the machine components undergo with that forces and acceleration. The information about the loading is useful for the maintenance tasks. In this work it is used to estimate the stress in the mechanical structure of the machine and the fatigue life of the machine. The next chapter describes different methodologies that can be used to estimate the fatigue. The chapter concludes with the choice of the methodology utilized in the current work.

4 Methods for Fatigue Life Estimation

Mobile working machines perform their job during many years in different conditions and under various loads. As the economically reasonable lifetime is estimated to be about ten years, in practice many machines work for several decades. During this period their mechanical structure suffers from many factors like tension, compression, torsion and bending as a result of different forces and torques acting on the elements of the structure. The temperature, humidity, and solar radiation which change over time influence the properties of the structure's material and its coating. Corrosion caused by these factors further deteriorates the structure properties. The resulting degradation of the mechanical structure leads to the fact that the structure can fail under the loads that are well below those that would be expected to cause failure on a single application of load. This phenomenon is known as fatigue. The influence of temperature, humidity and corrosion is hard to account for and its numerical evaluation is a subject for continuous research work. The influence of changing loads has been studied for more than a century and is well-known in materials science and engineering. The fatigue failure caused by these factors can be defined as the failure under a repeated or otherwise varying load which never reaches a level sufficient to cause failure in a single application.

In many cases, fatigue involves the initiation and growth of a crack until the crack size reaches a critical value, causing the separation of mechanical component into two or more parts. In some circumstances, other criteria of deterioration can be applied, but the situation described above covers components made of iron, steel or aluminium which are the most common materials. In these crystalline metals the initiation process is caused by slip on crystal planes, controlled by time-varying shear stresses. The start of a fatigue failure is therefore a strictly local process and it depends on the dynamics of the system. The critical factor is the time history of stress or strain values at the exact location where a crack is going to start. The general distribution of these parameters throughout the component is of secondary interest. That is why Finite Element Analysis (FEA) is used in this discipline. It allows an analyst to choose any location within a model and concentrate attention on it, using the intrinsic ability of FEA to bring in dynamic effects.

Three major fatigue methodologies have become established over time. Two of them do not model the crack growth process at all. Instead, they use the concept of similitude to determine the number of cycles to failure. A failure is defined as a predetermined crack length, or loss of stiffness, or separation of a component into parts. The relationship between component life and load level in a test specimen is compared directly with that expected in service (assuming the component is tested under the same conditions). The first of these two methodologies is based on stress and is called Stress-Life (S-N, nominal stress, or total life) approach. The second, and more recent technique, is based on strain. It is called Strain-Life (Local-Stress-Strain, Crack-Initiation, Manson-Coffin or Critical-Location Approach – CLA) approach. The third, and most recently developed method, deals with Crack-Propagation. It relies on the observation that once

cracks become established they have a stable growth period. This is usually described using linear elastic fracture mechanics. It further relies on the assumption that crack growth rates are proportional to the applied stress intensity (a function of crack length, geometry and stress level). Today, almost all fatigue design calculations are covered by one of these three approaches.

Since stress and strain are the governing variables it has been usual to test prototype components in order to obtain the required data needed for fatigue analysis. However, the introduction of FEA techniques has made it possible to perform fatigue calculations long before a prototype exists. Furthermore, an improvement in computing power has made FE based fatigue life calculations a routine task. It is considered now as a mature technology.

4.1 Stress-Life Methodology

This approach is normally used for total life calculation. A stress-time history is estimated for some part of the component considered to be representative. These data are transformed into a stress-time history at the point where a crack is likely to start using parameters derived from the geometry of the component. Traditionally, these parameters have been taken from tables and graphs of stress concentration factors, which often do not cover the exact component geometry being used. One of the most important roles for FEA in this field is to replace and supplement these graphs.

Material data are introduced in the form of tests which apply a cyclic load of constant amplitude and frequency to small smooth specimens till their total separation. These data are plotted as a number of cycles (duration of life) N against some nominal stress S. Comparing the stress-time history at the chosen critical point with this S-N curve allows a life estimate for the component to be made.

In a variation of this method the S-N plots are obtained for whole components. Life estimates in this case are limited to that component. Even minor changes in the geometry of the component produce a new test programme. This is not a very economic procedure for modern development programmes and another major role for FEA is to avoid the need for it.

A special case of component testing concerns welds in metals. Extensive tests have been made that resulted in S-N data published in standards like BS7608. These documents relate the total life of a welded component to the nominal stress in some point which is located at a predefined distance from the weld. These are not basic materials data, but component data, specific to a certain type of weld. The role of FEA in such a case is to provide the nominal, or reference, stress-time history at a place where stresses remain elastic.

The S-N method assumes the structure to be fully elastic, not just in structural terms, but even in local fatigue-related details such as notches. It is therefore only applicable to the

cases when a component undergoes relatively low loads and is designed for long life. Such cases are considered as high cycle fatigue (HCF) problems. In application to FE models, linear elastic stresses from FE analysis can be used directly to calculate fatigue damage.

4.2 Strain-Life Methodology

S-N approach assumes that stresses remain elastic in all parts of the component, even in the location where a crack will start. This implies that all stresses are low, and the component is operated during a long life. The methodology should be confined to lives greater than 10000 applications of load and is probably better confined to lives over 100000. In the case of higher loads, multiplying nominal stress by a concentration factor gives figures greater than the yield stress at the critical location. This leads to yielding and shorter lives. Strain-Life Methodology uses the strain response in the structure for such low cycle fatigue (LCF) problems. Methods that use empirical relationships to predict a strain-life history under reversed yielding were well developed before FEA became a general design tool. Although non-linear FEA could replace these methods in some cases, this is not the general practice. Instead, FEA is used to give better elastic predictions at points near to the critical location. These values are then converted to elastic-plastic predictions at critical locations.

When a strain-time history has been determined for the critical locations, material properties are introduced as the data from tests conducted on small smooth specimens under different ranges of constant strain. These tests are terminated when a small crack appears, and the predicted life is considered as the life to crack initiation. Large organizations can make test specimens by cutting the actual components in order to account for manufacturing effects when obtaining material properties.

4.3 Fracture Mechanics (Crack-Propagation) Methodology

If the crack propagation phase of life is to be considered, crack prediction models are needed for two tasks. The first task is to predict the rate of growth of a crack (e.g. mm per load application). The second task is to predict how long a crack can be before the next peak in the loading history causes catastrophic propagation.

These tasks are solved by Fracture Mechanics, usually Linear Elastic Fracture Mechanics. The controlling factor in both cases is the crack tip stress intensity factor, which depends on the crack length, the nominal stress near the crack tip and a factor, sometimes called the Compliance-Function. The latter factor depends on a component geometry. Expressions for it are hard to derive because of the singularity at the crack tip. The stress at the tip is plastic and in elastic analysis tends to infinity. Tables and graphs for it exist, as with stress intensity factor. The role of FEA in Crack-Propagation is therefore similar in some ways to its role in Stress-Life approach since it replaces existing data banks. FEA has even wider role in Crack-Propagation because Compliance-Function nearly always changes as the crack grows. If this is described by

a simple expression, numerical integration may be good enough. In many practical cases, though, the crack extends into a region with completely different geometry, and the adaptability of FEA becomes essential.

A common assumption in crack life estimation is that a crack of a certain length is present when the component is put into service. The crack initiation life is zero in this case. This assumption is particularly common when dealing with welds.

4.4 Using Finite-Element Modelling in the Stress-Life Approach for Fatigue Life Estimation

A mechanical structure of mobile working machines is usually designed for a long life. Safety and reliability requirements lead to such design of mechanical components that the stress can be considered to be elastic in all acceptable modes of operation. Many components are considered in simulations as rigid bodies. That is why the Stress-Life approach was chosen in this work for fatigue life estimation.

The Stress-Life approach assumes that all stresses in the component, even local ones, stay below the elastic limit at all times. It is the oldest of the three main methods and it is still suitable when the applied stress is nominally within the elastic range of the material and the number of cycles to failure is large. The nominal stress approach is therefore best suited to problems that fall into the category known as high-cycle fatigue. The nominal stress method does not work well in the low-cycle fatigue region where the applied strains have a significant plastic component. In this region, a strain-based methodology must be used.

A stress cycle in the most common form can be represented by Figure 4.1. The stress consists of two components, a static or mean state stress S_m, and an alternating stress amplitude S_a. In an idealised loading condition typical of that found in rotating shafts operating at constant speed and constant load the mean stress is zero. The maximum and minimum stresses in such a case are of equal magnitude but opposite sign, tensile stress being considered positive and compressive stress negative.

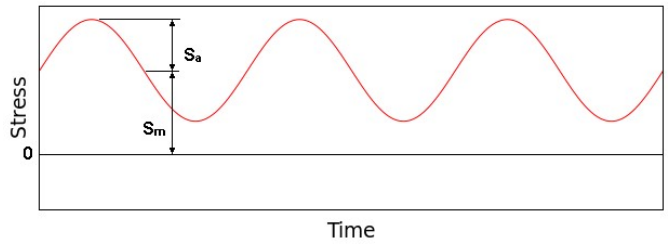

Figure 4.1: Example of a stress cycle

In more general situation the maximum and minimum stresses are not equal in magnitude. They are both tensile and so they define an offset for the cyclic loading. Sometimes it is necessary to consider the stress range, which is the difference between the maximum stress in a cycle, S_{max} and the minimum stress S_{min}. The stress amplitude is one half the stress range and mean stress is the algebraic mean of S_{max} and S_{min}. The stress ratio is S_{min}/S_{max}.

4.4.1 S-N curve

Holes, grooves, fillets and other geometrical features cause high local stresses under load, called stress concentrations. The basic fatigue properties of a material under ideal conditions should be measured using the specimens that are free of these features.

A common type of specimens is a cylinder loaded in an axial tension, free of sudden changes of geometry and with a polished surface at the critical section. A number of identical specimens are tested to total separation and the number of cycles needed is recorded as N. Load is kept constant during the test. For each specimen a nominal stress S is calculated from simple elastic equations and the results are plotted as the un-notched S-N diagram, a basic material property.

Some metals, particularly low alloy steels, have a two-line S-N plot, with flattening of the relationship when N is greater than about 10 million cycles. The line may become horizontal so that no failures occur at higher values of N and the material is said to have a fatigue limit, which is important if infinite life is the aim. A care should be taken because fatigue limit can be sensitive to a variety of effects such as mean stresses and corrosion. The treatment of the fatigue limit has had a lot of attention, and the convention for any particular class of components is dealt with in the software, but it is desirable for the user to know the convention. Welds in ferrous metals, for instance, are assumed not to have a fatigue limit but experience a change of slope at N=107.

Real components, unlike test specimens, must have holes, grooves, changes of section etc. which cause local “hot spots” of high stress. Fatigue will start at these hot spots and life calculations must allow for their effect. It is in making this allowance that FEA methods differ most from traditional ones. In both cases some way must be found of converting applied loading into local stresses at the point where a crack is likely to start. In the traditional approach features causing high local stresses are called “notches” and a key factor in dealing with them is the stress concentration factor. For any given geometry, like a circular hole for instance, this is defined as the ratio of the maximum stress in the region of the notch to the nominal stress remote from the notch.

Closed-form solutions exist for many of the simple common notches, and extensive tables and graphs have been published for the more difficult ones that are of practical importance. The fatigue analyst’s task is to identify a likely site of failure, identify a suitable place remote from that site where nominal stress can be calculated, and find an expression for the stress concentration factor which allows the two to be related. With

FEA it is possible to go directly from applied loads and component geometry to stress histories at likely sites of failure. This completely alters the basis of calculation, although it may still be convenient to retain some of the traditional approaches. FEA, for instance, can be used to calculate stress concentration factor values. The versatility of FEA would still provide an advantage unless the notch is a simple one. Combinations of notches close together would usually favour FEA. In some circumstances it may still be advantageous to use FEA to find a nominal stress remote from the notch and then use a well-tried stress concentration factor figure to convert this to a local stress. It is not always practicable to provide detailed meshes for local features in large models such as of a ship or an aircraft.

A fatigue life depends mainly on the amplitude of stress or strain existing in the component, but it is also influenced by the mean value of stress. Many components carry some form of "dead load" before the working stresses are applied. Some way to account for this factor is needed. Traditional methods use stress as the controlling factor. For a given life the allowable amplitude of fatigue stress gets smaller as the mean stress becomes more tensile and to a lesser extent increases when the mean stress is compressive. The latter effect is of great practical importance. The processes of cold-rolling, shot-peening, etc, are used to deliberately introduce compressive mean stress in surfaces and so improve fatigue resistance.

4.4.2 Variable Amplitude Loading

Material properties tests performed for fatigue analysis assume constant-amplitude loading with or without a mean offset. However, it is more common, especially for mobile working machines, that the loads and the responses vary in magnitude.

The simplest extension of the constant amplitude case is one in which the amplitude of the sine waves changes from time to time. The history in such a case consists of n_1 cycles of amplitude S_1, n_2 cycles of S_2 and so on. Usually, the pattern repeats after a small number of S values, say S_n. The sequence up to S_n is called a "block", and the target is to estimate how many of these "blocks" can be applied before failure occurs. The linear damage rule is generally used which is known as the Palmgren-Miner hypothesis (Palmgren, 1924; Miner, 1945). Having S-N data one can find the number of cycles of S_1 which would cause failure if no other stresses were present. Calling this N_1, the simplest assumption is that n_1 cycles of S_1 use up a fraction n_1/N_1 of the total fatigue life. Doing a similar calculation for all the other stresses and summing all the results gives the total damage fraction for one block. The fatigue life can be estimated then by the number of blocks N_b. Given as an equation this is:

$$\left(\sum_i \frac{n_i}{N_i}\right) N_b = 1.0 \tag{4.1}$$

The limitations of Palmgren-Miner hypothesis are that it is:

- Linear, i.e. it assumes that all cycles of a given magnitude do the same amount of damage, whether they occur early or late in the life;
- Non-interactive (sometimes referred to as sequence effects) i.e. it assumes that the presence of S_2 etc. does not affect the damage caused by S_1;
- Stress-independent i.e. it assumes that the rule governing the damage caused by S_1 is the same as that governing the damage caused by S_2.

These assumptions are known to be faulty. Experimental data indicate that the order in which various stress levels are applied may have a significant influence and also that damage rate at a given stress level may be a function of prior cyclic stress history. For the decreasing stress sequence, that is, if cycles with the level S_1 are applied first, followed by the cycles with the level S_2, and $S_1>S_2$, the sum $\sum n_i/N_i$ is typically less than 1. For the increasing stress sequence this sum is typically greater than 1. Experimental values for the sum (4.1) at the time of failure often range from about 1/4 to about 4, depending on the type of decreasing or increasing cyclic stress amplitudes used. As a result, other cumulative damage theories were proposed, e.g. Marco-Starkey, Gatts, Corten-Dolan, Manson damage theories (Collins, 1993).

If the various cyclic stress amplitudes are mixed in the sequence in a quasi-random way, the experimental sum (4.1) more nearly approaches unity at the time of failure, with values in the range of about 0.6 to 1.6. Since many service applications involve quasi-random fluctuating stresses, the use of Palmgren-Miner linear damage rule is often satisfactory for failure prediction.

4.4.3 Cycle Counting

Many real engineering components experience stress responses which are more complex than this. Consider the sequence shown in Figure 4.2.

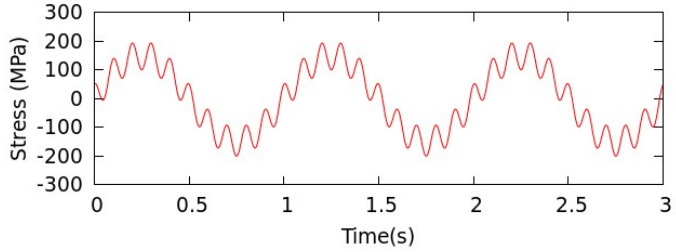

Figure 4.2: Example of an irregular time history

This is a commonly occurring situation where dual mode response is present in a structure. Each cycle transition is equal to approximately 100 MPa, but the overall transition of the peaks is over 400 MPa. The question here is what ‘cycles’ of stress to use? One approach would be to take the stress difference between adjacent peaks and troughs. This would result in many cycles of about 100 MPa.

However, because fatigue behaviour is non-linear, higher stress levels cause much higher fatigue damage. The above approach would therefore underestimate fatigue damage.

An alternative approach is to assume the peak levels as representatives of the stress amplitudes. If we used this amplitude for all cycles, this approach would grossly overestimate damage. Instead, a technique is required which can identify overall trends in the response, whilst also keeping track of intermediate and small response cycles properly.

Rainflow ranges have been widely used for estimating fatigue damage from random signals since Matsuishi and Endo first introduced the concept to the scientific community (Matsuishi and Endo, 1968).

The most common procedure for Rainflow cycle counting is the following:

1. Extract peaks and troughs from the time signal so that all points between adjacent peaks and troughs are discarded.
2. Make the beginning, and ending, of the sequence have the same level. This can be done in a number of ways but the simplest is to add an additional point at the end of the signal to match the beginning.
3. Find the highest peak and reorder the signal so that this becomes the beginning and the end. The beginning and ending of the original signal have to be joined together.
4. Start at the beginning of the sequence and pick consecutive sets of 4 peaks and troughs. Apply a rule that states that if the second segment is shorter (vertically) than the first, and the third is longer than the second, the middle segment can be extracted and recorded as a Rainflow cycle.
5. If no cycle is counted, a check is made on the next set of 4 peaks, i.e. peaks 2 to 5, and so on until a Rainflow cycle is counted. Every time a Rainflow cycle is counted the procedure is started from the beginning of the sequence again.

Eventually, all segments will be counted as cycles and so for every peak in the original sequence there should be a corresponding Rainflow cycle counted. The Palmgren-Miner rule can then be applied to the cycles to calculate the overall damage produced by the given stress history.

4.4.4 Material and Component S-N Curves

In component S-N analysis, including some welding standards, locations of reference stresses are often defined at a fixed distance from a geometrical feature. Sometimes, they are referred to as nominal stresses but more generally they are reference stresses. In this case, it is only the stress at this location that is required to quantify a fatigue life for the component.

Component S-N curves are generated by testing complete components, or pseudo-components, rather than smooth polished specimens of material. These curves can be used to estimate how long the component, as a whole, will last under cyclic loading. The failure location is pre-defined by the component itself during the cyclic testing process. The component S-N approach is very useful in situations where an accurate description of local stress, either elastic or elastic-plastic, is difficult to achieve such as in the case of welded constructions or composite materials.

In contrast, material S-N curves are produced using specimens, such as hour glass specimens, where a uni-axial stress can be calculated from elastic theory and compared directly with a failure life. The specimen is free from local stress raisers and so only the nominal stress level is important. As long as similar stress conditions occur again in another specimen or component, a similar failure life would be expected.

A material S-N curve therefore relates elastic stress, S, to the number of cycles, N, required to cause failure. Such curves can be used for detecting failure locations and estimating lives across an entire finite element model for which appropriate elastic stresses have been calculated. If all other factors are the same, and there is only one load case, the failure locations will correspond to regions of the model exhibiting the highest stresses. Furthermore, the distribution of expected lives can then be usefully represented by a contour plot of life.

4.4.5 Equivalent Stress-Strain Approaches

Traditionally, the approach to the design of components subjected to multi-axial loading is to make the following fundamental assumption: Failure under a multi-axial loading is predicted to occur, according to the theory associated with a particular modulus, if and when the cyclically induced magnitude of that modulus, is sufficiently large that failure would occur in the uni-axial state for an identical magnitude of the same modulus.

The mechanical modulus referred to above, is a measurable quantity such as principal stress, principal shear stress or distortion energy. This philosophy has lead to what is usually referred to as equivalent stress-strain approach. This is where an equivalent stress or strain modulus is calculated under multi-axial loading and then applied to uni-axial data.

These approaches are based on extensions to static yield theories. They assume that lifetimes for fatigue under multi-axial loading can be predicted by substituting combined stress or strain parameters in the uni-axial Stress-Life or Strain-Life equations (i.e., by calculating an equivalent uni-axial stress or strain for a given multi-axial situation). The main stress and strain parameters used are the maximum principal, the maximum shear, and the von Mises or octahedral. The big advantage of this kind of approach is that it enables the large amount of uni-axial fatigue test data available in the literature or in data banks to be applied to multi-axial situations.

The von Mises method is widely used, but all these methods have drawbacks (Bishop and Sherratt, 2000). The von Mises' prediction of yield in terms of the principal stresses is the following:

$$\sigma_0 = \sqrt{\frac{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}{2}} \tag{4.2}$$

The yielding occurs when σ_0 exceeds the monotonic yield stress.

4.5 Fatigue estimation approach used in the current work

This work uses the Stress-Life approach as a well-defined technique for High-Cycle Fatigue which is suitable for mobile working machines. Since the mechanical structure of these machines is subjected to multi-axial loading of variable amplitude, von Mises method is used to obtain the scalar value of stress and Rainflow counting is then applied to calculate the total damage produced by the stress history. These well-known and widely used techniques were selected to examine the applicability and peculiarities of using traditional approaches to fatigue life estimation in an IIoT environment.

The next chapter presents the methods proposed in the current work for the lifecycle support of mobile working machines. The method for remote surveillance gives the information about the motion of the machine and about its loading. The load history provided by this method is used by the fatigue life estimation method to predict the lifetime of the machine components.

5 Mobile working machines lifecycle support methods based on IoT and Digital Twin concepts

At the time of writing the most of existing IoT solutions utilize the data-driven approach in which the field data are gathered to the cloud database and processed using Big Data analytics. The processing is aimed to search for correlations in data that produce new information. This approach does not use at all or uses a limited amount of knowledge about the processes that generate the data and physical laws that link data with each other. Such knowledge is required primarily at the latter steps of data processing for interpretation of the results.

In model-driven approach the field data are used as an input to the simulation model of the object that generates or that is described by the data. The model accounts for data relations governed by the physical laws and thus can provide the results unavailable with data-driven approach. This chapter introduces several methods which are based on model-driven approach and allow building applications for lifecycle support of hydraulically actuated heavy equipment.

5.1 Remote Surveillance

Gathering real-time data from different devices in the IoT environment facilitates the task of monitoring the state of these devices. Real-time monitoring, analysis and control was being used for years in manufacturing industry within the SCADA systems. These systems used local area networks (LAN) to transmit the sensor data to the computational resources located at the manufacturing facilities. The IoT concept has made it possible to extend this technique to the wide variety of objects by replacing the LAN with internet and the local computational facilities with cloud resources.

A large number of monitoring solutions have been created with the development of IoT. In applications designed to monitor heavy equipment they gather geolocation data as well as the data from different sensors like fuel level, rotational speed of motors and pumps, temperature, position, acceleration, voltage, weight and discrete-event sensors like touch sensors, on/off states and error codes read from the engine. Gathering and processing these data provides a lot of information, for example, location of the machine and violation of location constraints, speeding, total distance travelled, location of stops made, engine hours, number of days used, quantity and cost of fuel used, average productivity of the machine and state of its main components. This information can be agnostic to the type of the machine except the estimation of productivity which should account for the type of the work the machine does and the dependence of productivity on the sensor data. Nonetheless, these dependences are quite simple and can be applied for a wide variety of machines. That is why the data-driven approach was the first to be adopted in IoT applications.

More information could be obtained if the sensor data are transferred to the simulation model of the machine. As an example, the physics-based dynamic model can be considered that connects the motion of the machine with the forces acting in it. For hydraulically actuated machines an easy accessible data are available for dynamic simulation. The pressure in hydraulic actuators provides the information about the forces acting in them and the position data give information about the motion of actuators. The pressure gauges are relatively simple to be installed in different parts of a hydraulic system and are widely used in heavy equipment. The digital gauges convert the analogue signal about the pressure to the digital value that can be displayed, stored and transmitted. The modern digital pressure gauges provide the measurement frequency of several kilohertz. As the natural frequency of major hydraulically actuated mechanisms used in heavy equipment is measured in hertz (Kovanen, 2003; Gottvald, 2010), the resolution provided by the digital pressure gauges is enough for accurate estimation of the forces acting in the machine.

The magnitude of the force produced by a hydraulic cylinder is calculated with the equation (3.1) which is repeated here for clarity:

$$F_{cyl} = p_1 A_1 - p_2 A_2 - F_{fr}(\dot{x})$$

p_1 and p_2 are the pressures in the cylinder chambers, A_1 and A_2 are the piston side and rod side areas, F_{fr} is the cylinder friction force which depends on the velocity of the rod. In order to calculate the cylinder force accurately, the rod velocity should be known. It could be measured directly or calculated from the simulation model of the machine.

The equation of motion of a multibody system can be written in the Newton-Euler formulation as follows (Shabana, 2010):

$$\mathbf{M}\ddot{\mathbf{X}} = \mathbf{Q}_e + \mathbf{Q}_c \tag{5.1}$$

where **M** is the system mass matrix, **X** is the vector of system generalized coordinates, $\mathbf{Q_e}$ is the vector of system generalized external forces, $\mathbf{Q_c}$ is the vector of the system generalized constraint forces which can be eliminated if the components of the vector **X** consist of the independent coordinates only. As **M** is in most cases a constant matrix, knowing $\mathbf{Q_e}$ and $\mathbf{Q_c}$ allows to calculate **X** by integrating (5.1) with the use of initial conditions: $\mathbf{Q_e}(0)=\mathbf{Q_{e0}}$, $\mathbf{Q_c}(0)=\mathbf{Q_{c0}}$, $\mathbf{X}(0)=\mathbf{X_0}$, $\dot{\mathbf{X}}(0) = \dot{\mathbf{X}}_0$. The components of the vector $\mathbf{X_0}$ can be found from the generalized coordinates of the elements of machine mechanism in some static unloaded position (for example, a transportation position of the machine). In a static position $\dot{\mathbf{X}}(0) = \ddot{\mathbf{X}}(0) = 0$, $\mathbf{Q_{e0}}+\mathbf{Q_{c0}}=0$. The values of $\mathbf{Q_{e0}}$ and $\mathbf{Q_{c0}}$ can be calculated in the particular static position from **M**, gravity and reaction forces.

In theory, starting from the initial conditions and substituting the $\mathbf{Q_e}$ vector into the equation (5.1) it is possible to simulate the motion of the system. In practice, the measurement or calculation of $\mathbf{Q_e}$ can not be performed with an infinite accuracy. Numerical solution of differential equation introduces an additional error which

increases over time. The use of discrete computational system adds more errors through discretization. Even the calculation of $\mathbf{Q}_e$ vector in discrete moments in time becomes more complicated for hydraulically actuated machines. As the force produced by hydraulic actuator depends on the velocity, the value of velocity should be measured together with the values of pressure. If the velocity can not be measured directly, its approximation depends on the discretization time step. This step has the lower bound which depends on the performance of the measurement device and in case of digital measurements on the performance of analog-to-digital converter (ADC). A small time step also increases the amount of data generated per time interval and the requirements for storage and processing facilities. If the measurement data should be transmitted over the communication network, the small time step imposes high bandwidth requirements.

In case of IoT applications the network bandwidth is often a key factor. With the increase of the number of internet-connected devices the traffic volumes also grow. To mitigate this issue the EDGE-computing paradigm has evolved that moves computational facilities to the edge of the network. The goal is to reduce the traffic volumes and network delays. At the time of writing the EDGE-computing is in its initial development state and existing EDGE-computing devices have limited performance. They are primarily targeted to relatively simple pre-processing of data gathered from a large number of geographically concentrated devices (like bounds checking) but not to computationally intensive simulations. Another issue is a poor network connectivity or the lack of internet access. The network coverage is uneven. In highly populated areas there are commonly several types of connection (wired, Wi-Fi, mobile) available together in one place. The density of wireless networks coverage in such places is often so high that it introduces delays and degradation of signal quality. In the areas located far from highly populated regions deployment and maintenance of powerful network resources is not economically reasonable which limits the type and bandwidth of network connection. In such places like mines and quarries, in the forest, in mountains or in the sea the satellite connection is often the only option. This type of connectivity is imposed to large network delays by the finite speed of signal transmission from the Earth to the satellite and back.

All these factors influence the way the physics-based simulations are used in particular application. For the case of remote surveillance on hydraulically actuated heavy equipment a model-based solution should consider the following topics:

1. The type of simulation model and numerical method for that model;
2. The discretization time step for the model;
3. The type of sensor data to be measured and the sampling rate;
4. The place where the calculations should be performed (locally on the machine, in the EDGE or in the Cloud);
5. The type and the amount of data to be transferred over the network;
6. The communication technology and protocol to be used;
7. The amount of data to be stored and the type of data storage;
8. The user interface and the way of results representation to the user.

We consider these topics in the following chapters. An example of remote surveillance system implementation is described in Chapter 6.3.1.

5.1.1 Simulation models and numerical methods suitable for remote surveillance

The main goal of remote surveillance is to provide the full picture of machine operation. The more information can be obtained about the machine, the better. For the large number of tasks, the position of a machine is more important than the forces acting in it. The use of a kinematic model that calculates the position of any part of the machine given the measured data about the position of some other parts is sufficient for such tasks. The kinematic model does not take into account the forces acting in a machine but it can reduce the number of sensors required for gaining the full information about the machine position. For the machines that have large mechanical structure and perform relatively slow and simple movements, like tower cranes or gantry cranes, a kinematic model provides the way to predict crane's motion, prevent collisions and gather crane usage statistics. It can even be used to estimate the machine productivity if the machine performs repetitive work with constant loads. If the knowledge about the forces acting in the machine is required, the dynamic model should be used. Position information augmented with the forces provides the full picture of machine state. Machine performance can be estimated for any kind of machines as well as such important parameters like stress and wear of different parts of the machine.

Among many available dynamic formulations, the ones that have lowest computational complexity should be used. Reducing the simulation time is important on several reasons. It allows processing larger number of models in a given period of time which is essential for the cloud-based systems. It also provides the possibility of performing real-time simulations with computing devices that have low performance. Such devices that are usually low-energy and low-cost can be installed on board a machine and make local calculations. As a result, the amount of traffic to the cloud is reduced as well as the network bandwidth requirements. On the other hand, the response time of the model is improved. Fast simulation models can also be processed faster than real-time and can be used for predictive tasks such as control optimisation or prevention of the operator's mistakes (Zhidchenko et al., 2018).

Several dynamic formulations exist with near to O(N) complexity as described in the Chapter 3. As many machines have a tree or a chain mechanical structure consisting of several links interconnected with different types of joints, the Iterative Newton-Euler Formulation is suitable in many cases.

5.1.2 Choosing the discretization time step for the simulation model

Hydraulic circuits of heavy equipment are known to introduce numerical stiffness into the system as they deal with fluid volumes of different orders of magnitude. The integration time step for such systems should be $10^{-4}..10^{-3}$ s depending on the hydraulic

circuit of the system (Piche and Ellman, 1994; Eman et al., 2008). The most accurate simulation could be achieved if the boundary conditions data are provided to the model with the same time step. In practice, the sensor data can be gathered in larger intervals and the values for the moments in time that lay between the measurements can be obtained through interpolation. The rate of acquiring the sensor data depends on many factors such as the sensor type, the analog-to-digital converter performance, the network bandwidth and the communication protocol.

5.1.3 The type of sensor data and the sampling rate

The data about the pressure and position of hydraulic cylinders provide an ability to calculate the cylinder forces according to (3.1). For hydraulically actuated equipment the cylinder forces produce torques in the joints of machine's mechanical structure. Knowing the forces and torques the machine's dynamics can be calculated using some dynamics formulation, e.g. Iterative Newton-Euler Formulation. Thus, the pressure and position of hydraulic cylinders together with inertia properties of the machine (that are often assumed to be constant) give the full information about the motion of the machine. For load-handling machines like cranes, loaders and excavators the inertia properties depend on the mass of the load. This mass can be measured directly using the load sensor which is present in many modern machines. For the case when load sensor is absent the mass of the load can be estimated. An example of a method for such estimation can be found in.

A typical pressure measurement scheme consists of a pressure transducer and analog-to-digital converter. Modern pressure transducers have the frequency response, i.e. the highest frequency that the sensor will measure without distortion or attenuation, in the range $N \cdot 10^2$ .. $N \cdot 10^5$ Hz. The exact value depends on the type and the cost of the sensor. The sensors used in mobile working machines have the frequency response $N \cdot 10^2$ .. $N \cdot 10^3$ Hz hence they can be used to measure the changing pressures with the time step 10^{-4} .. 10^{-3} s.

According to sampling theorem (Kotelnikov, 1933; Whittaker, 1935; Nyquist, 1928; Shannon, 1949) the maximum discretization time step suitable for restoring of analog signal should be one half of the period of the highest frequency of the signal. Thus, the smallest time step for discretization (sampling interval) of the pressure sensor signal should be $5 \cdot 10^{-5}$ s. As a common value for the performance of modern analog-to-digital converters is 10^6 conversions per second, this performance is enough for converting the pressure values to digital form.

Sampling theorem assumes that the highest frequency of the signal is strictly limited which is not the case in reality. An error introduced by the process of analog signal restoration from the discretized form is known as discretization error. In practice, instead of using an ideal filter required by the sampling theorem for a signal restoration, a polynomial interpolation is used in order to calculate the values of the signal between

the discrete measurements. Normalized root mean square error introduced by polynomial interpolation can be estimated as follows (Nazarov et al., 2007):

$$\begin{aligned} \gamma_{dl} &\approx \frac{\pi}{4} \cdot \frac{f_s^2}{f_o^2}, \textit{ for linear polynomial interpolation} \\ \gamma_{dq} &\approx 2.5 \cdot \frac{f_s^3}{f_o^3}, \textit{ for quadratic polynomial interpolation} \end{aligned} \tag{5.2}$$

where f_s is the highest frequency of the signal and f_0 is the sampling rate.

Eigen frequency of mechanical-hydraulic systems used in many hydraulically actuated mobile working machines is usually several Hertz (Sørensen, 2016). Thus, an upper limit for the f_s can be defined as 10^2 Hz. Taking the value of the sampling rate of modern analog-to-digital converters to be 10^6 Hz, the normalized root mean square error for quadratic polynomial interpolation can be estimated to be $\gamma_{dq} \approx 2.5 \cdot 10^{-12}$.

Total measurement error introduced by the process of obtaining sensor data from a working machine remotely through the internet (error of telemetry) depends on several parameters:

- sensor accuracy;
- quantization error;
- discretization error;
- synchronization error;
- data compression error.

An error is a random variable taking the values in some range. The absolute error is the difference between the value obtained through the measurement and the true value of a parameter. The relative error is the absolute error divided by the magnitude of the true value. As the real value is usually unknown, the absolute error can not be calculated exactly. An estimation of maximum error is used instead. Taking into account the probability of errors the maximum error can be a very rough estimation. In some cases, the maximum error can not be observed during the whole time period of measurements. That is why the value of measurement error is commonly estimated by the normalized root mean square error:

$$\gamma = \frac{\Psi}{L} \tag{5.3}$$

where Ψ is the root mean square error, $L=\lambda_{\max}-\lambda_{\min}$ is the scale of the measured parameter. The root mean square error of measurement λ' of the true value λ is the square root of the mean square error:

$$\Psi = \sqrt{E\left((\lambda' - \lambda)^2\right)} \tag{5.4}$$

where $E(\lambda)$ is the mathematical expectation of λ.

The components of the error of telemetry listed above are independent. Thus the total error of telemetry can be calculated as follows:

$$\gamma = \sqrt{\sum \gamma_i^2} \tag{5.5}$$

where γ_i is the *i*-th component of the total error.

A common value of accuracy of commercially available pressure sensors is γ_{se}= 0.05%. It can be shown that this value defines the total error of telemetry as other components of the total error are much lower.

The quantization error can be estimated as follows (Nazarov et al., 2007):

$$\gamma_q \approx \frac{\Psi_q}{L} = \frac{d}{2\sqrt{3}L} = \frac{d}{2\sqrt{3}d(D-1)} = \frac{1}{2\sqrt{3}(D-1)} \tag{5.6}$$

where *D* is the number of quantization levels, *d* is the quantization step. For the 16-bit analog-to-digital converter $D=2^{16}=65536$ and $\gamma_q=4.4\cdot10^{-6}$. For the 24-bit analog-to-digital converter $D=2^{24}=16777216$ and $\gamma_q=1.72\cdot10^{-8}$.

The synchronization error is caused by the discrepancy between the moment in time when a parameter being measured takes the true value and the moment to which the measured value is associated. In case where several parameters are measured to describe the state of the monitored object the maximum synchronization error can be estimated as the discretization error with the time step $n\cdot\Delta t$, where *n* is the number of parameters per measurement and Δt is the time step between the measurements of individual parameters. Taking $\Delta t = 10^{-6}$ s (which corresponds to the sampling rate of analog-to-digital converter 10^6 Hz) and n=14 for the example case of hydraulic mobile crane, the synchronization error can be estimated to be:

$$\gamma_{syn} = \gamma_{dq} \approx 2.5\frac{f_s^3}{f_0^3} = 2.5\cdot(10^2)^3\cdot(14\cdot10^{-6})^3 = 3.5\cdot10^{-11} \tag{5.7}$$

The data compression error is introduced by lossy compression of data during transmission and/or storage. If the compression is not performed at all or the methods of lossless compression are used, the data compression error is absent.

Summarizing different sources of error described above, the total error for the case of measuring 14 parameters using 16-bit analog-to-digital converter can be estimated as follows:

$$\begin{gathered}\gamma \approx \sqrt{\gamma_{se}^2+\gamma_q^2+\gamma_{dq}^2+\gamma_{syn}^2} = \\ ((5\cdot10^{-4})^2+(4.4\cdot10^{-6})^2+(2.5\cdot10^{-12})^2+(3.5\cdot10^{-11})^2)^{0.5} = \\ (25\cdot10^{-8}+19.36\cdot10^{-12}+6.25\cdot10^{-24}+12.25\cdot10^{-22})^{0.5} = \\ (2.50019360\cdot10^{-7})^{0.5}=0.000500019359625= \\ 5\cdot10^{-4}+1.9359625\cdot10^{-8}\end{gathered} \tag{5.8}$$

Expression (5.8) shows that the sensor accuracy is the dominant source of measurement error while the influence of all other sources of error is several orders of magnitude lower. The impact of other sources, especially discretization error, becomes more valuable if the bandwidth and reliability of communication channel is taken into account. This issue is described in more details later in Chapter 5.1.5.

Although the pressure in hydraulic cylinders of mobile working machines can easily be measured and converted to digital form using modern sensor equipment, there are issues with data transmission and storage in real-time IIoT applications.

5.1.4 A balance between the calculations and data transmission in IIoT applications

In order to make the real-time simulation with the time step 10^{-3} .. 10^{-4} s the sensor data should be delivered to computing device with sufficient speed. Despite the considerable advantages in networking technologies made during the last two decades, there are still limitations for network bandwidth and coverage at the time of writing. Mobile working machines often work on remote sites that are away of good network connectivity, for example in the forest, in the field, in the quarry or in the mine. The only option for network connection at these sites is some kind of wireless networking. Mobile operator's network or the satellite connection is the common solution. As at the time of this thesis writing the 5G mobile networks are still in development phase the highest throughput can be obtained using 4G networks. These networks can theoretically provide the data rates up to $N\cdot100$ Mbit/s. The issue is that they are usually designed and deployed for the customer market with the assumption that most users utilize their connection primarily for downloading the data from the internet rather than uploading something. As a result, the network bandwidth is allocated asymmetrically to provide the download speed that is nearly 10 times faster than the upload speed. The data rates provided in Europe are reported to be 10-50Mbit/s for downstream and 4-6Mbit/s for upstream (Safari Khatouni, 2018). In case of IIoT applications for mobile working machines the upload speed is more important for gathering the sensor data from the machine. Working at remote sites introduces additional challenges as the network coverage on these sites is often poor. It is not economically reasonable for a network provider to make a good coverage in a low populated area. As a result, the number of base stations and the network bandwidth in remote locations is usually minimal.

A satellite connection provides even less data rates but larger costs than mobile network. High prices for satellite networking have led to a common approach of using them for low data-rate communication in the locations with no mobile network coverage, e.g. in the mountains or in the sea. A separate problem when using the satellites placed on a geostationary orbit is a network delay of hundreds of milliseconds introduced by the limited electromagnetic waves propagation speed.

With the expansion of IoT the number of connected devices grows rapidly. It imposes additional restrictions for the real-time data transmission via wireless networks. To overcome these issues an approach have been proposed to place computing capabilities as close to the data sources as possible. The concept of EDGE-computing is developing. It assumes that data should be processed where they are gathered and just the results of processing should be transmitted over the network. This concept is contradictory to cloud-computing that collects and processes all the data centrally in the cloud system. The further development of the approach is fog-computing that assumes creation of a large number of small data centers or data processing units that have smaller compute capabilities than cloud systems but are located closer to the data sources. The fog-computing is also considered to be a consumer of data generated by the EDGE and a data provider for the Cloud.

In EDGE computing the networking devices that provide access to the network and are located on the edge of it (giving the name to the concept) should be capable of performing calculations. The economical reasons do not allow to supply these devices with high performance computing. As a result, their computational resources are limited. They are suitable for simple processing of received data, for example, checking the threshold conditions and sending the alarm to the cloud system when a condition is violated. Taking into account a large number of networking devices and their parallel data processing architecture it is possible to perform high-performance calculations in EDGE environment using parallel computing. This approach can only be implemented for the tasks that allow segmentation into independent subtasks which is not always possible. It also requires a lot of efforts in creation both the parallel algorithms and a middleware for running them on several devices at once. At the time of writing this type of EDGE-computing is under development and the most of the EDGE devices available on the market represent ordinary computers equipped with a number of input/output ports and often implemented as rugged devices. For the case of mobile working machines an EDGE device could be an industrial computer installed on board a machine and used for performing simulations.

A promising approach is using low-energy microcomputers which are also called single board computers (SBCs) built on a mobile platform like ARM or Intel Atom. The experimental results presented in Chapter 6.2 demonstrate the ability to use such kind of computers for real-time multibody simulation. More computational performance can be obtained by integrating several microcomputers into a single computing cluster. SBC clusters represent intensive research area (Johnston et al., 2018). There are a lot of implementations, intended for research and educational purposes and used as scale

models of high-performance clusters and cloud systems (Tso et al., 2013; Abrahamsson et al., 2013). The brightest example is 3000 node Raspberry Pi cluster constructed by Los Alamos National Laboratory to be used as a testbed for research tasks related to exascale computing (Lapid, 2018). Other implementations evaluate the possibilities of running different cloud computing tasks on SBCs such as Map-Reduce tasks (Hajji and Tso, 2016; Srinivasan et al., 2018). In 2018-2019, following about 5 years of evolution from experimental design to mass production products, commercially available SBC clusters appeared on the market targeting industrial, education and maker markets. These hardware platforms outpace creation of software solutions capable of efficient execution on them. That is why the main application areas for the first commercially available SBC clusters are education and custom embedded systems. Industrial users have not yet adopted SBC clusters, especially in mobile working machines.

A single microcomputer has substantially lower performance than a common desktop computer. The experiments with dynamic simulation models demonstrate a difference of 4-16 times depending on particular hardware and simulation method being used (Malysheva et al., 2018). In order to get similar performance in different kinds of programs about 10 or more SBCs should be integrated into a single cluster. Taking into account the price of microcomputers, physical housing, power and networking subsystems, the total price of such a cluster becomes comparable with the cost of ordinary computer. The drawback is that computationally efficient use of a cluster requires creation of a parallel counterpart for the simulation program. This fact slows the adoption of SBC clusters as on-board computing systems for mobile working machines.

The main advantage of SBC clusters in comparison with ordinary computers is their energy efficiency (W/Flop) (Ou et al., 2012; Filho et al., 2017). This property is not so important for mobile heavy equipment as the power consumed by a computing device is little relative to other power consumers in a machine. On the other hand, the combination of low power consumption and high computational performance allows to propose another implementation of computer simulations for mobile working machines.

High-performance simulations can be implemented as an autonomous mobile computational service. This solution assumes that high-performance mobile computing cluster is installed on site (e.g. on a construction site, in a mine, in a quarry or in a forest) that is far from the good internet connectivity. The cluster is capable of performing multiphysics simulations, for example, using the methods developed in this work. The cluster is connected by a wireless network to the machines on site. The machines send sensor data to the cluster and it performs the processing. The advantage is that there is no need to install a computing device on each individual machine or the cheap, low power and low memory devices can be used on-board. There is no need for a separate internet connection for each machine. There could also be no internet connection at all which is often the case in a mine. More computationally intensive simulations can be run providing better simulation results. Gathering and processing data from several machines makes it possible to control the on-site job that these

machines should perform. This solution can facilitate the remote-controlled or fully autonomous mining or other jobs. Using blockchain technology, the computations can be ordered as a service and paid by machines themselves eliminating the human intervention into the process. Such application requires customization of existing software or the development of new parallel programs for multiphysics simulation capable of running on SBC cluster.

Local execution of calculations on-board a machine introduces a problem: what data should be stored on the machine itself and what should be transferred to the cloud? The answer depends on the application being used. In remote monitoring solutions the minimum data that should be sent consist of alarm signals that are generated on the machine and must be received by responsible personnel. Other data that could be sent depend on the parameters being monitored such as geographical position, productivity, fuel consumption and so on. The most reasonable approach seems to be storing on-board a machine the history data as long as possible and sending to the cloud in real-time just the most important information. The history represented by the sensor data that was used as input for simulation models provides the most flexible solution. It allows not only to reproduce the results of the simulations but also makes it possible to get new results utilizing other simulation models. These models could provide better accuracy or serve new applications.

If a good network connection is available, the sensor data can be transmitted directly to the cloud computing system. The possibility of sending data in real-time depends on the type of network connection, communication protocol and data size.

5.1.5 The communication technology and protocol to be used

There are different ways to send data over the internet in IIoT applications ranging from low-level transmission over TCP or UDP using custom data formats to utilization of special-purpose IoT protocols. At the time of writing there are two widely used communication protocols for IoT applications: MQTT and CoAP.

MQTT protocol is designed to send textual and binary data over TCP. It is the most famous IoT protocol supported by almost all modern IoT platforms. It is suitable for reliable transmission of slowly generated data like temperature or car traffic volume on a highway. The drawback of using this protocol for sending fast generated real-time sensor data is that it is based on TCP. Connection initiation and termination, reordering of packets, retransmission of lost packets introduces delays into the communication process. For the case of fast-generated real-time sensor data the speed is often more important than the reliability. If some data are lost due to network errors or overload it can be restored by interpolation or even can be totally ignored if the target application allows data loss to the detriment of accuracy. In such applications the use of connection-less protocols like UDP is more suitable.

In UDP the data are split into fragments which are called datagrams. The datagrams are sent independently from each other. They can be received out of the order that they were sent and can be dropped somewhere on their path to destination address. Nonetheless, the low overhead of UDP has made it a basic protocol for real-time data transmission over the internet, e.g. for the voice traffic. In order to use the advantages of UDP the CoAP protocol was developed for IoT applications.

MQTT and CoAP were designed to be lightweight protocols and to add as little overhead to the data being transmitted as possible. However, an attention should be paid when the particular protocol is selected for the real-time sensor data transmission. Consider the simplest case when the data are sent directly through TCP or UDP with some user-defined datatype and the physical layer is represented by Ethernet network. Figure 5.1 shows the structure of a single unit of data being sent over the network which is called protocol data unit (PDU). In the example case it is an Ethernet frame. The PDU has a length limit which depends on the protocol. For Ethernet it is 1518 octets (bytes) when 802.1Q extensions are not used and several octets longer in the opposite case. 1500 bytes are occupied by user data (which is often called payload) and 18 bytes are Ethernet frame header and footer. Ethernet is a physical layer and data link layer protocol. The data of upper layers like IP and UDP is transmitted inside the payload part of the frame decreasing the number of bytes that can be dedicated for payload.

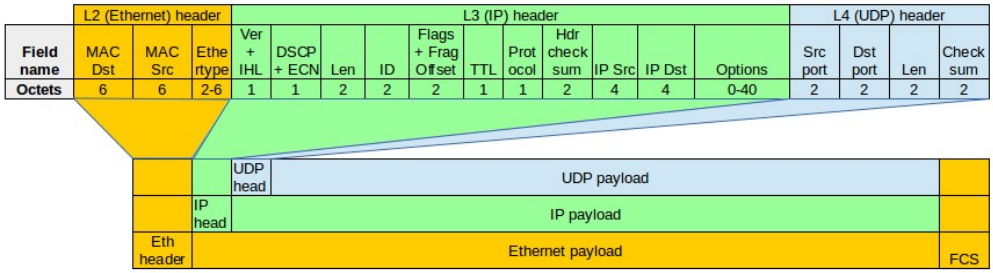

Figure 5.1: Ethernet frame structure

If there are upper layer protocols, their headers further occupy the payload area of PDU and the number of bytes available for actual data decreases. Different network protocols have different PDU sizes, but Ethernet PDU can be used as a basis for estimations concerning data transmission through the internet. The reason is that a majority of internet servers as well as many routers have Ethernet interfaces and Metro Ethernet is a common technology for building metropolitan-size provider networks. The maximum size of PDU (called Maximum Transmission Unit or MTU) in other types of networks (WiFi, 3G/4G) have similar values. Table 5.1 shows the values of network bandwidth, MTU and payload for the types of network access technologies that are commonly used in IIoT applications. It also shows the maximum number of measured parameters that can be transmitted per second using the PDU of maximum size. Each parameter is assumed to be represented by floating point number with single precision. The last column of the table includes the number of measurements that can be sent per second

for the example case of monitoring the hydraulically actuated crane described in Chapter 6.

Column 12 of the Table 5.1 shows that the overhead introduced by the protocol headers increases with the number of protocol layers involved into transmission. For the case of using Virtual Private Network (VPN) connection over TCP with data encryption and IP tunnelling the overhead is approaching 1/10 of total data being transferred. It means that nearly 10% of network bandwidth is wasted for transmission of service data instead of user data.

Although a single PDU can carry several hundreds of floating point parameters (see column 15 of the Table 5.1), the exact capacity depends on the data format selected for parameters representation. Binary format is several times more compact than a textual representation. As an example, consider the largest normal number represented by the commonly used single-precision binary floating-point format *binary32* defined in IEEE 754-2008 standard (IEEE, 2008):

$$0\ 11111110\ 11111111111111111111111_2 = 2^{127} \cdot (2 - 2^{-23}) \approx 3.4028234664 \cdot 10^{38}$$

The decimal format of this number represented as the string "3.4028234664E38" encoded into the two-byte UTF-8 format for textual representation occupies 30 bytes (240 bits) instead of 4 bytes (32 bits) in binary format. Using XML-like markup and meta-data as in widely used MQTT protocol to separate or describe the data being transmitted adds more overhead. It may enhance the interoperability and readability of the data but reduces the transmission bandwidth. The last line of the Table 5.1 shows an extreme case when utilization of textual format reduces more than seven times the number of parameters that can be transmitted inside a single MTU.

Column 18 of the Table 5.1 presents the number of measurements that can be sent per second for the example case of monitoring the hydraulically actuated crane described in Chapter 6. Each measurement includes 14 parameters (a time stamp, pressure and position of the cylinders and a load mass) represented by 32bit floating-point numbers. Although the maximum speed of most modern networking technologies allows transmission of thousands of measurements per second (top three rows of the table), experimental results are less optimistic. The table presents the upload speed obtained in experiments with real 3G and 4G mobile networks in Europe and Malaysia described in (Safari Khatouni, 2018; Abolfazli et al., 2015). These data show that in remote locations that are the common workplace for mobile working machines and where low-speed 3G/4G is the common media for network connection, sending data faster than 1000 measurements per second is not always possible. It means that a real-time transmission of sensor data acquired with the measurement frequency available in modern ADCs is not feasible in many real scenarios. The sensor data should be consumed by a simulation model locally in the machine or the rate of transmission should be less than the rate of measurements. Interpolation should be used in such a case in a simulation model as described in Chapter 5.1.9.

Table 5.1: Parameters of network technologies that are commonly used in IoT

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Networking technology	Protocols used	Bandwidth (bit/s)	MTU	Ethernet	IP	UDP	TCP	IP tunnelling	VPN (TLS)	Total overhead (bytes)	Total overhead (% of MTU)	Payload (bytes)	Bits per parameter	Number of parameters per MTU	Number of parameters per second	Number of parameters per measurement	Number of measurements per second
LAN	Gigabit Ethernet	1.0E+09	1518	18	20	8				46	0.030	1472	32	368	30303030	14	2164502
LAN	100Mbit Ethernet	1.0E+08	1518	18	20	8				46	0.030	1472	32	368	3030303	14	216450
WLAN	IEEE 802.11 Wi-Fi	1.5E+08	2304	18	20	8				46	0.020	2258	32	564	4593912	14	328136
Abstract 1M	1Mbps + IP + UDP	1.0E+06	1518	18	20	8				46	0.030	1472	32	368	30303	14	2164
Abstract 1M	1Mbps + IP + TCP	1.0E+06	1518	18	20		20			58	0.038	1460	32	365	30055	14	2146
Abstract 1M	1Mbps + IP + UDP+VPN	1.0E+06	1518	18	20	8		20	69	135	0.089	1383	32	345	28470	14	2033
4G LTE (MONROE)	25Mbps + IP + UDP+VPN	2.5E+07	1500		20	8		20	69	117	0.078	1383	32	345	720312	14	51450
4G avg (Malaysia)	4Mbps + IP + UDP+VPN	4.0E+06	1500		20	8		20	69	117	0.078	1383	32	345	115250	14	8232
3G avg (MONROE)	2.2Mbps + IP + UDP+VPN	2.2E+06	1500		20	8		20	69	117	0.078	1383	32	345	63387	14	4527
3G min	0.4Mbps + IP + UDP+VPN	4.0E+05	1500		20	8		20	69	117	0.078	1383	32	345	11525	14	823
Worst case	0.4Mbps+IP+TCP+VPN+txt+1E6	4.0E+05	1500		20		20	20	69	129	0.086	1371	240	45	1523	14	108

Accounting for the MTU size is not only important in the sense of network bandwidth utilization. If the volume of user data is larger than the payload part of MTU the data can be split (fragmented) by the network subsystem into several PDUs. This process as well as reassembling the data on the receiving end is performed automatically and does not influence much the data transmission if a reliable network media is used. The drawback is a slight decrease of data rate when PDUs which are smaller than MTU appear in the network flow due to the fragmentation. The fragmentation also adds some extra load to network devices but it is often considered to be negligible as the devices are highly optimized for this common operation. The packets can arrive at the destination address in a different order than they were sent. Some of the packets can be lost because of errors in a data transmission or because of network congestion. Transport protocols designed to maintain reliability, such as TCP, deal with reordering

and retransmission of packets. As a result, these issues influence just the data rate. In case of connection-less protocols such as UDP, the impact can be more harmful. The datagrams in UDP protocol are transmitted independently from each other. Some fragments of a fragmented datagram can reach the destination while the others can be lost. As UDP does not support retransmission, the loss of a single fragment leads to the loss of the entire datagram.

If the probability of a single packet loss is P than the probability of loss for the fragmented datagram consisting of N packets is $N \cdot P$ as the datagram is lost whenever any of its fragments is lost. The fact that IP fragmentation reduces the reliability of Internet communication is well recognized. "UDP Usage Guidelines" defined in RFC8085 does not recommend to use the fragmented datagrams. It states that "an application SHOULD NOT send UDP datagrams that result in IP packets that exceed the Maximum Transmission Unit (MTU) along the path to the destination. Consequently, an application SHOULD either use the path MTU information provided by the IP layer or implement Path MTU Discovery (PMTUD) itself [RFC1191] [RFC1981] [RFC4821] to determine whether the path to a destination will support its desired message size without fragmentation." (RFC8085, 2017).

Figure 5.2 presents the network diagram of the test environment built for studying the influence of packet fragmentation on the reliability of data transmission over UDP. The shown connection type is common for IIoT applications. The laptop with the client program was connected to internet through Wi-Fi network created by the Android smartphone and then through the 4G LTE connection. The server computer was connected to the internet through the Metro Ethernet link. The server was running OpenVPN server software providing secure access over UDP with TLS encryption. The client was connected to the VPN and used private IP addresses for communication with the server. The client program was sending UDP datagrams of a fixed size to the server. The server was counting received datagrams and displaying their content.

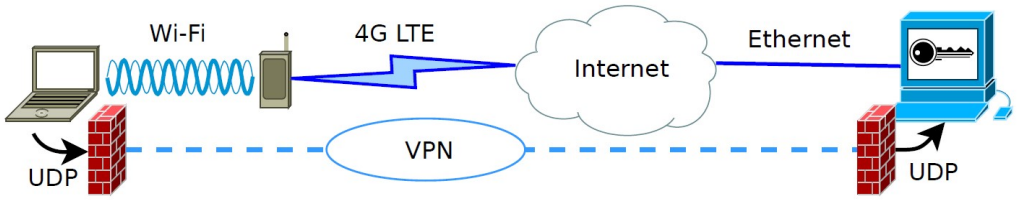

Figure 5.2: Network diagram for UDP datagram fragmentation test

Figure 5.3 presents the results of sending 1000 UDP datagrams over the test connection described above. Two types of datagrams were sent: 1000 bytes and 2000 bytes length. The first type was small enough to fit inside a single frame of link layer protocol and was transmitted in unfragmented form. The second type was larger than MTU and was fragmented by the network subsystem into two IP packets. The network traffic coming through the client and the server interfaces was monitored with packet analyzer software

to be sure that all packets were sent by the client and to observe the fragmentation. The rate of datagram generation was varied from 100 datagrams per second to 20000 datagrams per second simulating the different rate of sensor data acquisition and transmission. Each transmission consisting of 1000 datagrams was performed 6 times in sequences consisting of 5 subsequent transmissions delimited by 5 second intervals. The intervals between the sequences were random and composed of several minutes in order to decrease the influence of network condition fluctuations. The average number of lost packets was calculated for each transmission.

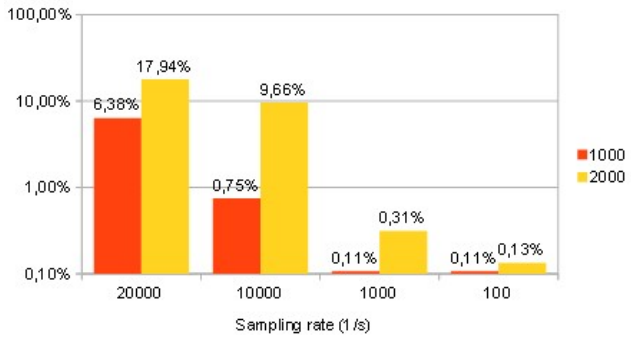

Figure 5.3: Percentage of lost packets for the datagrams of different size

Figure 5.3 shows that the number of lost datagrams is much higher when packet fragmentation is used. In the experiment it was 3 .. 10 times greater than in the situation when a datagram fitted within the MTU. The speed of datagram generation also influences the number of losses. In both cases of fragmented and unfragmented datagrams, an increase in data generation rate increases the number of lost datagrams. Figure 5.4 demonstrates the decrease of actual rate of datagram transmission due to packet losses.

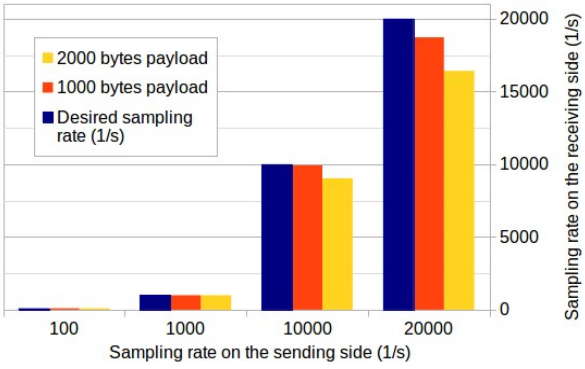

Figure 5.4: Actual rate of datagram transmission due to packet loss

Receiving measurements with lower rate than they are sent at the originating point is equivalent to sending data less frequently. Figure 5.5 presents the discretization error calculated with the equation (5.2) for the cases of using different network technologies and data rates presented in Table 5.1. The border value for the discretization error is defined as $2.25 \cdot 10^{-4}$. This value assumes that the sensor accuracy equals 0.05% and the impact of discretization error to the overall measurement error is less than 10% of the impact of sensor accuracy:

$$\begin{aligned} \gamma &\approx \sqrt{\gamma_{se}^2 + \gamma_{dq}^2} = \\ &= \sqrt{(5 \cdot 10^{-4})^2 + (2.25 \cdot 10^{-4})^2} = \\ &= \sqrt{25 \cdot 10^{-8} + 5.062510^{-8}} = \\ &= 5.48292805 \cdot 10^{-4} = \\ &= 5 \cdot 10^{-4} + 4.8292805 \cdot 10^{-5} \end{aligned} \tag{5.9}$$

Figure 5.5 shows that in order to ensure the discretization error to be less than 10% of the error introduced by the sensor accuracy, the data rate should be more than 2Mbit/s for the example case of hydraulically actuated mobile crane.

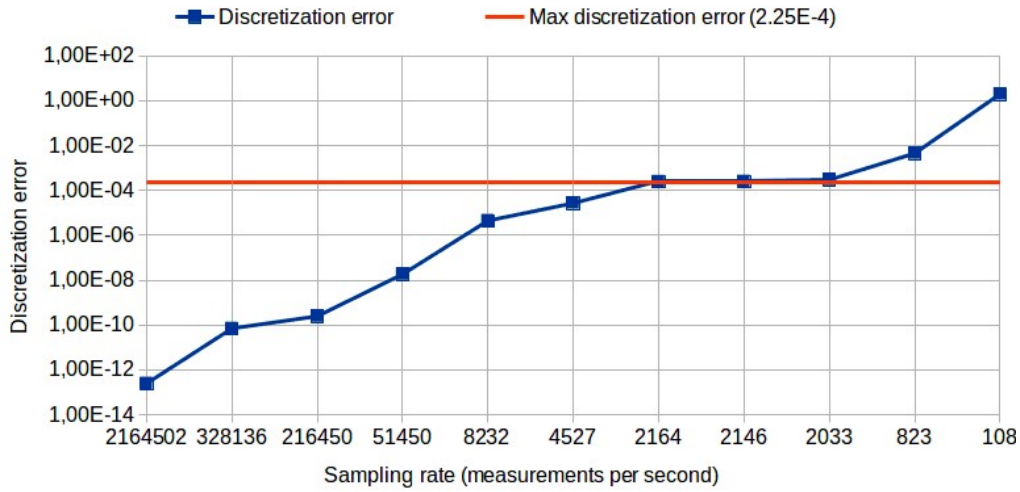

Figure 5.5: Discretization error for the different values of sampling rate

For an arbitrary mobile working machine whose dynamics can be simulated using the data from N_{se} sensors, each data represented by N_b bits, transmitted through the network with the bandwidth W bits/s and the percentage of data loss χ percent, using the PDU with the length of N_{PDU} bits and the header of N_h bits in each PDU, the normalized root mean square error introduced by the data transmission can be estimated with the following expression:

$$\gamma_{Tr} \approx 2.5 \frac{f_s^3}{\left(W\left(N_{PDU} - N_h\right) \frac{\left(1 - \frac{\chi}{100}\right)}{N_{se} N_b N_{PDU}} \right)^3} \tag{5.10}$$

where f_s is the highest frequency among all the signals obtained from the sensors. Expression (5.10) is derived from (5.2) and assumes that quadratic polynomial interpolation is used on the receiving side to restore the intermediate data points. Another assumption is that each data unit in the data flow contains the header of a constant length. It happens when TCP or unfragmented UDP is used on the transport layer of network connection.

As the actual rate of data acquisition at the receiving point is smaller than that at the sending point due to network bandwidth limitations and possible transmission losses, data interpolation should be used to restore the missing measurements. The experiments described in Chapter 6 demonstrate that the interpolation allows decreasing the sampling rate considerably down to 50-100 Hz for the dynamic models of hydraulically actuated mobile working machines.

5.1.6 The amount of data to be stored and the type of data storage

The data obtained during simulation should be saved for later access and analysis. Specific values to be stored, the type and the amount of stored data depend on the application. For remote surveillance application two approaches can be proposed.

In order to minimize network traffic, it is reasonable to make some calculations on-board a machine and send only the results to the cloud instead of sensor data. In the simplest case, these results could be alarm signals indicating, for example, an improper usage of the machine or the productivity estimation data. The sensor data should be stored on-board the machine. If the details of the machine's motion are needed, they can be obtained by accessing remotely the machine and reproducing the dynamics simulation. The amount of data storage that is needed on-board depends on the number of sensors being used, the datatypes that represent the sensor data and also on the use of data compression.

The sensor data can be used in many kinds of simulation models and a lot of information can be obtained about the machine besides its dynamics. Aggregating data from many machines allows for a broader analysis using Big Data methods. Providing this capability assumes storing sensor data from many machines in one place to make them available for analysis. The common approach nowadays is using cloud systems as the place to store and process large amounts of data. In such an approach the sensor data should be transmitted to the cloud. To minimize the impact to the network the communication should be made in periods of inactivity of a machine, e.g. at night, and data compression should be used.

Consider a mobile working machine with a mechanical structure consisting of N_L links which are booms, arms etc. The motion of these links is controlled by N_L actuators. They produce forces or torques acting on the links. An actuator position is usually described by one parameter. For a cylinder it is linear displacement (cylinder stroke) and for a motor it is an angle. If the position of any link can be completely defined by the position of its actuator, which is often the case, the position of links can be described by N_L parameters constituting the independent coordinates of the machine mechanism. Let N_{Fi} parameters should be measured to calculate each force or torque. For a hydraulic cylinder such parameters are the values of pressure in two cylinder chambers and the velocity of rod movement according to (3.1). The mass properties of links in a common case do not change and can be stored as constants. Even if a mechanism allows replacement of some parts, for example, connection of different implements, their characteristics should be stored in advance and used as constant data for the simulation. Let N_M mass parameters be changed at each time step. Usually N_M =1 and it represents the mass of the load, but in a common case there could be several changing mass parameters that should be taken into account. The total number of variables needed at each time step to simulate machine dynamics can be defined by the following expression:

$$N_{ts} = \sum_{i=1}^{N_L} N_{Fi} + N_L + N_M + N_{\mathrm{var}s} \tag{5.11}$$

The first term in (5.11) is the number of parameters needed to be measured in order to calculate all the forces and torques, the second term is the number of linear or angular displacements of actuators, the third term represents mass properties of a mechanism that change over time and should be measured. N_{vars} is the number of other variables used during calculation. It includes constant parameters like inertia properties or geometry data and all variables needed to store intermediate data. The memory for N_{vars} variables is allocated once and is used in all calculations. Thus, the total amount of memory needed to store the measurement data can be defined as follows:

$$N_{md} = \left[\left(N_t + \sum_{i=1}^{N_L} N_{Fi} + N_L + N_M \right) f_0 + \left(\sum_{i=1}^{n_{cond}} N_{cond_i} \right) f_{cond} \right] T N_{dtype} \tag{5.12}$$

where N_t is the number of parameters describing the moment in time at which the measurement is made, f_0 is the sampling rate (the number of measurements performed per second), n_{cond} is the number of additional parameters describing the conditions in which the machine is working (geo-position, temperature, humidity, fuel level, etc.), N_{cond_i} is the number of parameters describing each condition, f_{cond} is the sampling rate of conditions measurement, T is the time interval being simulated expressed in seconds, N_{dtype} is the size of the data type used to represent each measurement expressed in bytes.

The moment in time can be represented by the date detailed down to one second (8 bytes) and a time tick within the second (4 bytes for the measurement frequencies up to 4GHz). It is not reasonable to store the date for each measurement because many

measurements can be performed during a second. A compact way of storing time moments is to assume the date as a condition of a machine and counting the time ticks between the moments when the condition is measured. Using 4 bytes to represent the time tick, such an approach allows acquiring measurements with a sampling rate of 1MHz, performing condition monitoring only once per hour.

The estimation of amount of memory needed for storing measurement data for different types of mobile heavy equipment is presented in the Table 5.2. The data type used to represent measurements is binary32, so N_{dtype} =4 bytes. The time at which each measurement is made is represented by 12 bytes.

Table 5.2: The estimation of amount of memory needed for storing measurement data

Machine type	Number of links, N_L	Number of values for the force calcula-tion, N_{Fi}	Number of chang-ing mass parame-ters, N_M	Sampling rate, f_0 [1/s]	Data volume for 1s [bytes]	Number of machine condition parameters	Conditions sampling rate [1/s]	Data volume for 1 day (24 working hours per day) [bytes]	Memory needed to store the data for 1 month [Gbytes]	Memory needed to store the data for 1 year [Gbytes]
Hydraulic crane, simplest case	4	3	1	100	8000	3	0.017	691217280	20	239
Hydraulic crane, realistic case	4	3	1	100	8000	13	1.000	695692800	20	241
Backhoe loader	4	3	1	100	8000	3	0.017	691217280	20	239
Excavator	4	3	4	100	9200	3	0.017	794897280	23	275

Table 5.2 demonstrates that condition parameters do not add significant volumes of data as they are gathered with less frequency than the measurement data. The maximum volume of data representing one month of work is 23Gb. In many cases this volume can be reduced at least twice, since the working time of most machines is less than 24 hours a day. Such a volume of measurement data can be stored on board a machine using inexpensive single-board computer with an SD card that has storage capacity of 16 .. 64Gb. Thus, the use of SBCs allows storing the history data for several months of operation in a common case. Using more powerful on-board computing device such as an industrial PC, it is possible to store the history data representing several years of operation.

Storing sensor data in the cloud requires high volumes of storage capacity. The fleet of 1000 machines, which is a common case for a single large construction company or an equipment rental company, requires about 300 Tbytes of raw-data storage. A cloud

system serving 100 such companies needs 300 Pbytes of storage for the period of 10 years.

5.1.7 The user interface and the way of results representation to the user

The web-interface is a common type of user interface nowadays, especially for cloud-based systems. Its main advantage is an independence from the client-side hardware and software platform, as well as an ability to access the system from any place that has internet connectivity.

High performance of multibody dynamics simulation makes it possible to reproduce the motion of a machine in real-time as a result of simulations. In order to display the motion performed by a machine during the particular time period, the raw sensor data representing that period of time is read from the database and used as input to the simulation model. The simulation results can be visualized in real time and displayed to the user concurrently with simulations. An example structure of such a system is presented in Chapter 6.3.

If the results of simulations are not only used for displaying the motion of a machine but also for gathering some statistical or accumulated information (the number of working cycles, liftings, loads, the average speed of movement, average/total length or distance, etc.), these results should be stored separately and displayed to the user in appropriate form through the web-interface. Particular ways of storage and presentation of statistical data are not considered in this work.

5.1.8 The structure of remote surveillance system for mobile working machines

A possible structure of remote surveillance system for hydraulically actuated mobile working machines is shown in a simplified form in Figure 5.6. Real-time sensor data containing the pressure and position of hydraulic actuators are transmitted through the internet to the cloud platform where it is stored in a database. A user interacts with the system by means of a web-interface. When the user requests the motion history of a machine for some period of time, the simulation program is started by the cloud platform. It extracts the sensor data corresponding to the requested period of time and runs multibody simulation program to reproduce the motion of the machine. The simulation results are presented to the user in a visual form as a 3D model of the machine.

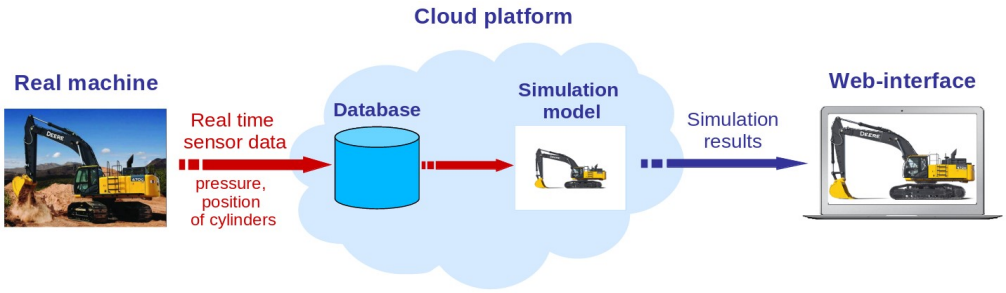

Figure 5.6: Simplified structure of remote surveillance system for hydraulically actuated mobile working machines

The specific set of sensor data, the data format and communication protocol used for transmission, the type of the database and visualization technology for results presentation can vary depending on the type of the machine and particular implementation of the system. The algorithm executed by the simulation program is described in the following chapter.

5.1.9 The algorithm for the motion simulation in remote surveillance system for mobile working machines

The sensor data containing the pressure and the position of hydraulic actuators can be used to calculate the forces acting in a hydraulically actuated mobile working machine and to reproduce the motion of the machine. The simulation algorithm works on sensor data stored in a database and representing some period of time that is of interest for the user. Retrieving the data iteratively for subsequent periods of time, it is possible to watch the motion of a machine continuously. The sensor data consist of samples and each sample contains the pressure and position of all the actuators as well as the time stamp of the particular moment in time when the measurement was made. The simulation algorithm can be presented in the following pseudo-code:

1. Read the first sample that corresponds to the beginning of the time period selected by the user from the database.
2. Initialize the arrays **X** (position of actuators) and **P** (pressure in actuator chambers) with the data obtained in step 1.
3. Initialize the matrices **Xwin** and **Pwin**. The matrix **Xwin** has the number of rows equal to the number of actuators. Each column represents a sample – the values of position of each actuator in some moment in time. The number of columns depends on the numerical methods used for interpolation and differentiation. In this work four columns were used that is enough for natural spline interpolation and differentiation with Newton method. Matrix **Pwin** has the same structure but the number of rows is equal to the sum of the number of chambers in all actuators.

4. Calculate the initial position of the machine using its geometry and obtained data about the position of actuators.
5. *CurrIter* = 0; *DIter* = 0; *dDisp* = <number of iterations between the frames in machine motion visualization>; *Stop*=FALSE;
6. While (*Stop*==FALSE) do {
7. While (*CurrIter – DIter* < *dDisp*) do {
8. if (*FirstStep*) { SimulateNSteps(1) }
9. else {
10. Read the next sample from the database into *TS*, and arrays **lX** and **lP**
11. if (the sample read is the last one) { *Stop*=TRUE; *DIter = – dDisp;*}
12. else {
13. *N* = (*TS-LastTS*)/*dt*; *t=TS;*
14. Shift the columns of matrices **Xwin**, **Pwin** leftwise, replace the last column with vector **lX** and **lP** accordingly.
15. SimulateNSteps(*N*);
16. *LastTS* = *TS*; **P** = **lP**; **X** = **lX**;
17. }}}
18. DisplayMachinePosition(); *DIter = CurrIter;* }

Procedure SimulateNSteps(*N*) calculates *N* time steps:

1. For each *i* from 0 to *N*-1 do {
2. Calculate **X** by natural spline interpolation using the data in **Xwin** matrix.
3. Calculate the vector of derivatives **Xdot** using the Newton formula and the data in **Xwin** matrix.
4. Calculate **P** by natural spline interpolation using the data in **Pwin** matrix.
5. Calculate the vector of actuator forces **F**=f(**P**,**X**,**Xdot**).
6. Calculate the vector of torques produced by actuator forces **T**=f1(**F**).
7. Solve the differential equation $\mathbf{M}\ddot{\mathbf{X}} = \mathbf{Q}$ on iteration *i*, where **M** is the mass matrix of the machine, **X** is the vector of generalized coordinates, **Q** is the vector of forces and torques calculated in steps 5 and 6 accordingly.
8. Calculate constraint forces taking the vectors **X**, **F** as an input.
9. *CurrIter = CurrIter+1; t = t+dt;*
10. } // end of <for> loop

It is more convenient to represent the algorithms that contain conditional constructs in a visual form. The field of visual programming deals with various forms of graphical representation of algorithms (Kovartsev et al., 2017). A kind of algorithm representation in the form of control flow diagrams known as Graph-Symbolic Programming Technology have been developed at the department of Software systems of Samara

National Research University (Egorova and Zhidchenko, 2015). It presents an algorithm as a graph with the nodes representing the actions and the arcs representing the flow of control between these actions. Conditions that allow the control to be transferred between the actions are defined by labelling the arcs with conditional statements called predicates. Figure 5.7 shows the algorithm for the simulation of a machine motion described above in a visual form created with the Graph-Symbolic Programming Technology.

A visual representation facilitates the comprehension, the development and the modification of the algorithms, especially those that have a complicated logic. The graphical elements allow more clear representation of actions and control flow and help the developer to concentrate on the logic by the manipulation of graphical objects but not text strings.

Graph-Symbolic Programming Technology provides the means for checking some of the algorithm properties. It also allows building the source code of the program automatically if the actions are defined in some programming language.

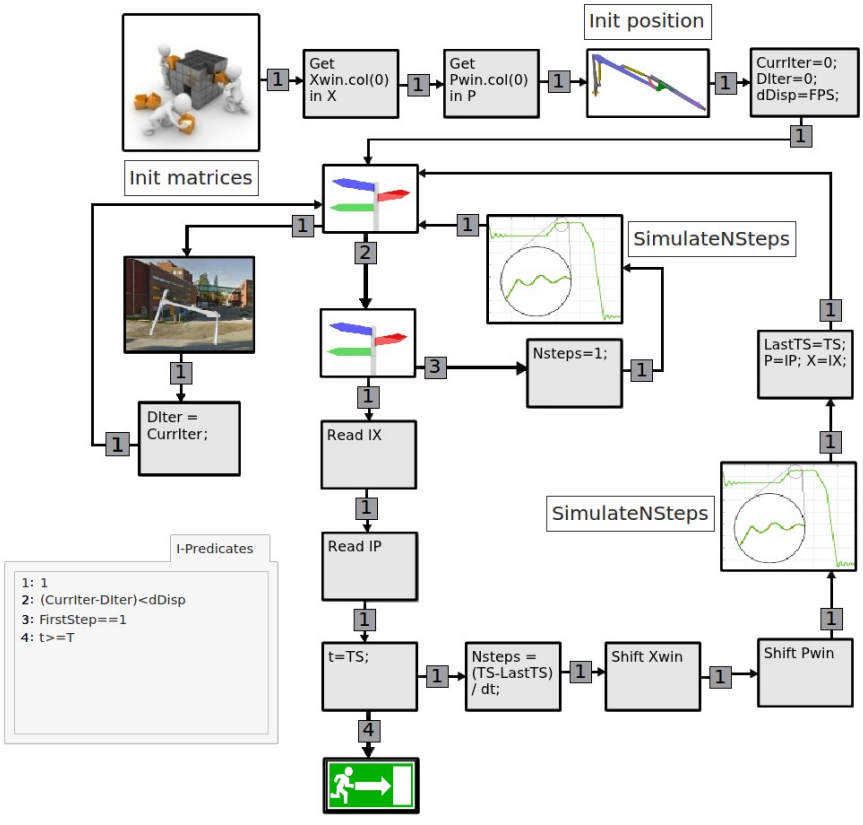

Figure 5.7: Visual representation of the algorithm for the motion simulation of a machine

5.2 Fatigue Life Estimation of Hydraulically Actuated Mobile Working Machines

5.2.1 Description of the Proposed Method

This study uses stress-based approach for fatigue life estimation. For each structural component of a machine the finite-element model (FE-model) is created or is obtained from the manufacturer of that machine. The FE-model discretizes the complex geometry of the component and linearizes the problem of stress calculation. Given the loads applied to the component, the FE-model provides the stress in each element of the mesh – a discretized representation of the component.

Using the load history of the component as an input for FE-model, the stress history of each mesh element can be obtained. The stress history and material properties are directly used by stress-based fatigue life estimation methods.

Computing and storing the stress history for each mesh element of each machine component requires a lot of computational and storage resources. It seems more reasonable to consider just some set of the mesh elements. For this purpose, a time period with typical load history of the machine is analyzed and the elements of the mesh with highest stress levels are chosen as a reference set for fatigue life estimation. Since higher stress levels provide less fatigue life, the calculation of fatigue life only for this set of elements is sufficient for fatigue life estimation of the entire component. The nodes belonging to the elements from the reference set are called hot-spots.

Finite-element analysis is usually applied for static problems when the simulated component does not move. In order to make it suitable for investigating the moving components of mobile working machines a technique called "Inertia Relief" is commonly used. This technique is provided by FEA software. It calculates the acceleration obtained by the simulated body as a result of application of the boundary conditions. Additional boundary condition is generated. It applies the opposite acceleration to the center of mass of the body or distributes the opposite acceleration among several elements of the model to make the equilibrium and eliminate the rigid body motion.

Utilizing the "Inertia Relief" technique, the whole process of fatigue life estimation for mobile working machine can be constructed the following way:

1. Real-time data from working machine are used as an input for digital-twin model to simulate the motion of machine components. This simulation can assume the machine component as a rigid or flexible body. In this study the rigid body motion is assumed, and the flexibility is taken into account only at the FEA step.

2. During the simulation the acceleration of each machine component as well as constraint forces are calculated at every time step and stored for later processing.
3. Using the mesh constructed from CAD drawing of each component and setting constraint forces as boundary conditions, a finite-element analysis is performed to calculate the stress in hot spots of each component. Acceleration of the component is utilized in Inertia Relief technique to eliminate the rigid body motion and provide quasi-static analysis. The time step for FEA, i.e. the moments in time at which the stress is calculated, can differ from that one used for simulation. The main requirement is that it should be suitable for accounting all vibrations of the component.
4. The stress history obtained at step 3 is used for fatigue life estimation. This study uses the rainflow counting method to extract stress cycles from fluctuating stress history, but other counting methods can also be applied depending on the fatigue estimation software being used. Applying the Palmgren-Miner rule to the stress cycles extracted from the stress history the cumulative damage can be obtained and fatigue life can be estimated.

The hot spots considered at step 3 can be defined manually by the manufacturer of the machine or by its user relying on the experience of machine utilization. They can also be defined automatically. For this purpose, each machine component can be covered with the nodes positioned from each other with some user-defined distance. These nodes can also be placed automatically during the mesh construction process. At the beginning of machine lifecycle, the stress history is calculated and stored for all the nodes. After some period of time, that is large enough to cover a common cycle of machine operation, for example, a week or a month during which the machine was used in all modes of its operation, the nodes with highest levels of cumulative stress are identified and defined as hot spots. The time period for hot spot identification can be chosen with the help of remote surveillance technique.

Following this approach, it is possible to account for different manners of machine operation used by different users. Gathering statistics along the lifecycle of several different machines it is also possible to define the safest manners of operation for every type of machine and every kind of work performed with them (mode of operation). It can be accomplished by first choosing the machines with lowest cumulative stress levels and then finding the differences between the manners of operation of these machines and other machines performing the similar tasks. Comparing the stress histories of different models of the machines executing the similar tasks it is possible to choose the best suitable model for a particular mode of operation. It can be useful for companies owing large fleets of different machines and operating them the similar way throughout the lifecycle, for example, for mining companies.

5.2.2 Application of the proposed method to the test model

This chapter considers an application of the proposed method for the fatigue life estimation to the test model of the mobile crane which is described in detail in Chapter 6. The model uses iterative Newton-Euler dynamic formulation to simulate dynamics of the crane. The simulated mobile crane consists of the pillar that rotates around the vertical axis, lift boom and jib boom that are actuated by hydraulic cylinders and an extension boom that is actuated by hydraulic cylinder located inside the jib boom. The INEF formulation provides the system of differential equations that is solved to get the values of independent coordinates of the system: the angle of the lift boom θ_1, the angle of the jib boom θ_2, the angle of rotation of the pillar θ_0 and the length of the extension boom L. This system of differential equations is derived analytically using the types of joints between the booms, geometrical parameters of the booms and their inertial properties as the input values. Inertia properties such as the mass, inertia tensor and position of center of mass can be obtained using the FEA software and CAD drawings of the booms together with their material properties. The system of differential equations is solved numerically, for example, by the Runge-Kutta fourth-order method to get the values of independent coordinates at each time step.

To get the values of constraint forces in the joints at each time step the following system of equations is considered:

$$\mathbf{M}\ddot{\mathbf{X}} = f(\mathbf{F}_{ext}) \tag{5.13}$$

where $\mathbf{M}$ is the mass matrix of the system, $\ddot{\mathbf{X}}$ is the vector of linear and angular accelerations of the bodies comprising the system, $f(\mathbf{F}_{ext})$ is the vector function that defines for each body and each component of its linear and angular acceleration the sum of constraint and external forces and moments acting on that body. This system of equations can be solved for constraint forces using the accelerations and external forces such as actuator forces and gravitational forces as input values. The system of equations (5.13) is just another dynamic formulation of the multibody system. Using it this way eliminates the need of solving a differential-algebraic system of equations. If the differential equation solver is implemented as a custom program, the calculation of constraint forces can be performed at each time step after the values of independent coordinates at this step have been found. This saves the time and storage space needed for saving the intermediate results of calculations.

The calculated constraint forces as well as actuator forces are stored as CSV-files (text files in which lines consist of numeric values separated by some symbol, i.e., comma). Each line represents a time step and comprises the value of time and values of components of the force. These files are used as input parameters that represent boundary conditions for finite-element analysis at each time step. In this study the open-source software Code-Aster was used for finite-element analysis. It allows defining a time-varying function with the values stored in the CSV-file. Using this capability the

time-varying boundary conditions are defined to calculate the stress in the points of interest at each time step. These values of stress comprise the stress history.

The Code-Aster software contains also some capabilities for fatigue analysis. For example, the CALC_FATIGUE command allows (among its other capabilities) the calculation of cumulative damage using the stress history and the S-N curve (Wohler curve) of the material. This command uses the rainflow counting method to extract the elementary load cycles from the stress history. The cumulative damage is calculated as the sum of the damages associated with the elementary cycles using the Palmgren-Miner rule.

Since the Palmgren-Miner linear damage hypothesis assumes that the total damage is a sum of contributions from elementary load cycles over the time, it is possible to develop an iterative procedure for fatigue life estimation during the lifetime of machine. For this purpose, the real-time data gathered from the machine are being split to portions suitable for processing on the available computing resources. As multibody dynamics simulation can be performed in real time without considerable requirements for resources, the main limitations are the random-access memory available for FEA software and the amount of time needed for stress calculation at all time steps of the period being processed. Depending on the available resources and the complexity of the mesh, the size of time period for stress history calculation should be defined. This definition could be made manually or automatically on the basis of measurements of time needed to process the first datasets acquired from the machine. The best solution should periodically adjust the size of the time period being processed depending on the load of the computing system.

The selected size of time period for processing defines the time boundaries for multibody simulation and the size of CSV files with calculated constraint forces and actuator forces. After finite-element analysis of that time period the calculated cumulative damage is added to the total damage of machine component that is stored in the database. The maximum value of damage among all hot spots should be monitored in order to produce an alarm when it exceeds the set limit.

5.2.3 Using Different Methods for Fatigue Life Estimation during the Machine Lifecycle

Different methods for fatigue life estimation can be utilized at different periods of the lifecycle. As the fatigue life of mobile working machines is evaluated in the design stage, its estimation during the early period of machine lifecycle is of low importance. The stress-based methods can be used at this stage for cumulative damage calculation to monitor and predict the failure caused by improper use of the machine. Such misuse can generate the stress levels that differ significantly from those being assumed at the design stage and as a result the fatigue life also changes. For example, hydraulically actuated mining drills are reported to be misused by scaling, which mean that instead of drilling, the boom is used to pry loose rocks from the wall (Rantalainen, 2012). Real-time

estimation of fatigue life together with the remote surveillance technique can detect such kind of unexpected work cycles of the machine in order to warn the owner of the machine and to account for them in the design of new machines.

When the cumulative damage reaches the value of crack initiation, the transition can be made to using the fracture mechanics methods in order to calculate the speed of crack propagation. This is especially important when machines remain in operation after the design fatigue life has expired. It is often the case for large and expensive machines like mining excavators or gantry cranes (Li, 2015).

There are several ways for application of fracture mechanics methods to the old machines, where the cracks already exist. One option is to enter the position and other parameters of the cracks discovered by the inspection of the machine performed during the maintenance activities. The speed of propagation of discovered cracks can be monitored automatically in order to define the time of the next maintenance. This approach should not replace the inspection intervals defined by standards but could compliment them in case where the large or uncommon loads lead to the need for more frequent inspections.

It is also possible to detect the crack initiation zones automatically by analyzing the points with maximum stress levels along the lifecycle. Combination of this approach with subsequent application of fracture mechanics methods can be useful for remanufacturing process as part of sustainable practices. Fatigue damage obtained by the component of a machine during its lifecycle can serve as a basis for the decision about the possibility of using this component in remanufacturing process. Automatic detection of crack initiation zones and estimation of crack propagation speed can also define the locations on the component for repair.

For old machines that have been used for years and that are most vulnerable for fatigue damage (such as mining excavators) the following approach can be proposed. The hydraulic system of a machine is modified by adding pressure and position meters to the cylinders. The load sensor, if it is absent in the machine, is also added or simulated. The stress history of the machine is gathered during some period of time that is large enough to cover all different conditions of machine operation. The length of the time period depends on the intensity of the machine usage and the number of different conditions in which the machine can be used. The gathered stress history is extrapolated back in time for the full period of machine exploitation prior to measurements. The resulting "augmented" stress history is used for fatigue life estimation of the machine. The detected hot spots with highest stress levels can define the zones for thorough inspection and the calculation of a crack size and propagation speed can define the methods for repair. As a result, the remaining fatigue life of a machine can be extended. The applicability of the proposed method depends on the repetitiveness of operations that machine performs. Mining excavators are good examples of the machines which perform repetitive operations for years.

The similar approach is used in useful life estimation for prolonging the life of used cranes (Kai Qi, 2013; Liftech Consultants Inc., 2002). The difference is that in case of dockside cranes and gantry cranes all work cycles are nearly the same and vary in weight and size of the loads. In such conditions, the load history is represented by distribution of weight of the loads and total number of lifts during the life of the crane. Work cycles of working machines, on the other hand, differ from each other and gathering real-time data with the help of IoT technologies improves the accuracy of fatigue estimation.

In order to evaluate the developed methods in terms of the accuracy and performance, to test the designed software system structure and to better understand the problems arising in the implementation of the proposed approach, the experiments have been performed. The results of the experiments are presented in the next section.

6 Experiments

6.1 Description of the model used in the experiments

The proposed methods for remote surveillance and fatigue life estimation of hydraulically actuated mobile machines were tested on the example model of the mobile crane PATU-655 (Figure 6.1). This crane has two booms: the lifting boom and the extension boom of variable length. Its maximum load is 500 kg for the case of the maximum extension boom length. The crane has five hydraulic cylinders. Two of them comprise the slew mechanism, providing the rotation of the crane around the vertical axis. Other two cylinders raise the lifting boom and the extension boom accordingly and the fifth cylinder is located inside the extension boom and controls its length.

(a) (b)

Figure 6.1: The simulated mobile crane and its model in Simscape Multibody.

The kinematic and dynamic models of the crane were created using the crane geometry data provided in (Kovanen, 2003; Luostarinen et al., 2014). In papers (Zhidchenko et al., 2018; Malysheva et al., 2018) these models were compared with the reference models created in commercially available simulation software MATLAB/Simulink. The models have demonstrated the performance that allows simulating the motion of the crane performed during some period of time, spending several times less time than the period being simulated. It allows using limited computational resources for simulations with near real time speed either on board a machine or in the cloud.

6.1.1 Kinematic and dynamic models

The kinematic and dynamic models of the crane can be built using the fact that the crane forms a kinematic chain of the bodies interconnected through the revolute and prismatic joints. The numbering of the joints is indicated in the Figure 6.2. The crane has four independent coordinates: the angle of the pillar rotation θ_0, the angles of rotation of the lifting boom (θ_1) and the extension boom (θ_2), and the length of the extension boom L.

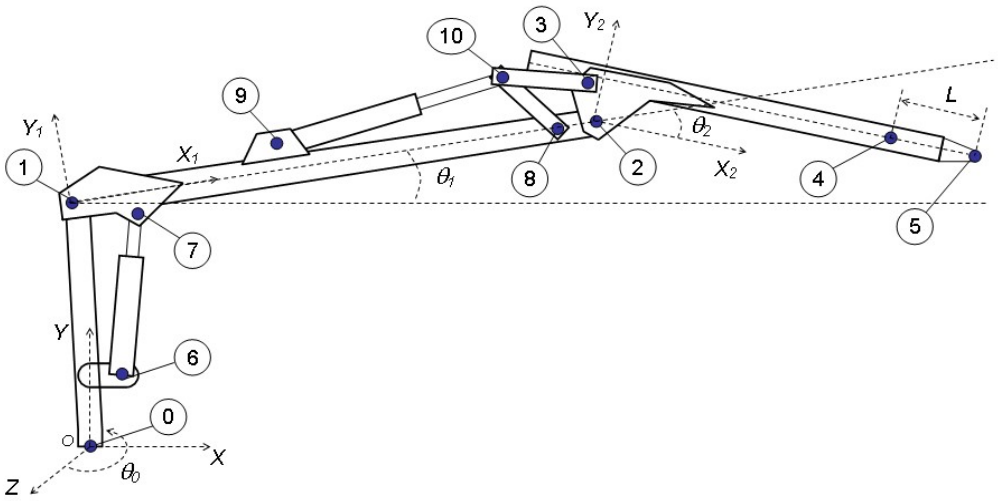

Figure 6.2: The numbering of the joints of the crane used in the model

The kinematics of the crane is rather simple. The global origin is located at the bottom of the pillar in the joint 0. Every point of the crane in a global space is represented by a 3x1 position vector $\mathbf{r} = [x, y, z]^T$.

A common practice for kinematic chains is to provide each link with the local coordinate system with the origin located in some joint. For example, the origin of the coordinate system associated with the first boom is located in the Joint 1. Using the angles of rotation of the booms, the global coordinates of each point of the crane can be obtained from its local coordinates in the following way.

Orientation of the body relative to some coordinate system is defined by the 3x3 rotation matrix. For example, rotation in the XY-plane counterclockwise through an angle θ about the origin of a Cartesian coordinate system can be described by the following matrix:

$$\mathbf{A}=\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.1}$$

where θ is the angle of rotation.

The position of any point located on the boom number j can be represented in the local coordinate system associated with the joint number i as follows:

$$\mathbf{r}^i = \mathbf{R}^j + \mathbf{A}^{ij}\mathbf{u}^j \tag{6.2}$$

where $\mathbf{r}^i$ is the position vector of the point relative to the joint i, $\mathbf{R}^j$ is the position vector of the origin of the local coordinate system associated with the joint j, relative to the

joint i, $\mathbf{A}^{ij}$ is the rotation matrix of boom j relative to boom i, $\mathbf{u}^j$ is the position vector of the point relative to the joint j.

Translation and rotation together can be represented by 4x4 transformation matrix $\mathbf{T}_{ij}$:

$$\mathbf{T}_{ij}=\begin{bmatrix} A^{ij} & R^j \\ 0 & 1 \end{bmatrix} \tag{6.3}$$

Using the transformation matrix, the position vector of a point can be represented as follows:

$$\mathbf{r}_4^j = \mathbf{T}_{ij}\mathbf{u}_4^j \tag{6.4}$$

where $\mathbf{r}_4^j = [r_x^j, r_y^j, r_z^j, 1]^T$ and $\mathbf{u}_4^j = [u_x^j, u_y^j, u_z^j, 1]^T$.

The notation described above allows defining the transformation matrices between the coordinate systems associated with the joints. For the joints 0, 1, 2 and 4 the transformation matrices are defined as follows:

$$\mathbf{T}_{01} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & R_x^{Jp1} \\ \sin(\theta_1) & \cos(\theta_1) & 0 & R_y^{Jp1} \\ 0 & 0 & 1 & R_z^{Jp1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{12} = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & r_x^{Jp2} \\ \sin(\theta_2) & \cos(\theta_2) & 0 & r_y^{Jp2} \\ 0 & 0 & 1 & r_z^{Jp2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.5}$$

$$\mathbf{T}_{24} = \begin{bmatrix} 1 & 0 & 0 & r_x^{Jp4} \\ 0 & 1 & 0 & r_y^{Jp4} \\ 0 & 0 & 1 & r_z^{Jp4} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $\mathbf{R}^{Jp1} = [R_x^{Jp1}, R_y^{Jp1}, R_z^{Jp1}]^T$ is the global position vector of the joint 1, $\mathbf{r}^{Jp2} = [r_x^{Jp2}, r_y^{Jp2}, r_z^{Jp2}]^T$ is the position vector of the joint 2 in the coordinate system associated with the joint 1 and $\mathbf{r}^{Jp4} = [r_x^{Jp4}, r_y^{Jp4}, r_z^{Jp4}]^T$ is the position vector of the joint 4 in the coordinate system associated with the joint 2.

Using these matrices, it is easy to calculate the global position vector of any point of the crane. For example, the global position vectors of the joint 2 and joint 4 are calculated as follows:

$$\mathbf{R}^{Jp2} = \mathbf{T}_{01}\mathbf{T}_{12}\mathbf{Z}, \quad \mathbf{R}^{Jp4} = \mathbf{T}_{01}\mathbf{T}_{12}\,\mathbf{T}_{24}\mathbf{Z} \tag{6.6}$$

where $\mathbf{Z} = [0, 0, 0, 1]^T$.

The dynamic model is built with the assumption that it should use the simplest expressions to be calculated as fast as possible. For kinematic chains such results could be obtained using the Iterative Newton-Euler Formulation (INEF) that was introduced in (Luh et al., 1980) and is described in detail in (Craig, 2005).

The iterative Newton-Euler dynamic formulation consists of two iterative procedures: forward (or outward) iterations and backward (or inward) iterations. In the forward iterations, for each link its velocity and acceleration is calculated, one link at a time, from the first to the last link. These iterations use the kinematic equations for the velocity and acceleration propagation from link to link. In the backward iterations, joint reaction forces and torques acting in the revolute joints are calculated one link at a time from the last link to the first one. Newton-Euler equations of motion are used in these iterations. Iterative Newton-Euler dynamic formulation assumes that the position, velocity and acceleration of each joint are known. It is aimed to be used to solve the inverse dynamics problem and to calculate the forces and torques needed in the joints to provide such velocities and accelerations.

Using INEF, the expressions for the torques acting in the joint 1 and the joint 2 can be obtained. These expressions can be used to derive the equations for the angular acceleration of the lifting boom and extension boom. The derived equations form the system of ordinary differential equations that can be solved to get the angles of rotation of the booms. Due to space constraints of the paper the complete system of differential equations is not included. However, it should be noted that it has quite complex structure even in the planar case.

To solve the system of ordinary differential equations, the torques acting in the rotational joints should be known. The torques are calculated from the forces provided by the hydraulic cylinders. These forces are obtained from the hydraulic model. The torque acting on the lifting boom is calculated from the angle of the boom and the cylinder force and stroke. The torque acting on the extension boom is calculated using the four-bar mechanism that connects the main and the extension booms.

The angles of the booms obtained as a solution of the system of ordinary differential equations are used to calculate the position of any point on the crane with the help of kinematic expressions mentioned above.

6.1.2 The model of hydraulic system

The hydraulic system of the mobile crane typically has the following structure. Several hydraulic actuators are controlled by separate proportional directional valves that often have the closed center position. The flow in that sort of applications is supplied by common actively controlled pump such as variable displacement pressure compensated pump. Such a pump maintains the assigned pressure level by adjusting the delivered flow. The drawback of the presented configuration is that the pressure level of the most loaded actuator affects the velocities of the others. In order to overcome this

phenomenon the pressure compensator can be added to each control valve. The pressure compensator will ensure the constant pressure drop across the control valve and, thus, will maintain the linear dependency between the flow rate and valve opening area (Axin and Krus, 2013).

The hydraulic circuit, which is used for modeling, is schematically presented in Figure 6.3. It includes three asymmetric hydraulic cylinders. Each of them produces the force with the magnitude that can be written as follows:

$$F_s = (p_A A_1 - p_B A_2) - F_{fr} \tag{6.7}$$

where p_A and p_B are the pressures in the cylinder chambers, A_1 and A_2 are the piston side and rod side areas, F_{fr} is the magnitude of the cylinder friction force. The cylinder friction model accounts its velocity dependency as follows (Andersson et al., 2007):

$$F_{fr} = F_C \tanh(v/v_C) + bv \tag{6.8}$$

where F_C is a Coulomb friction; v_C is a Coulomb velocity threshold; v is a cylinder velocity; b is a viscous friction coefficient.

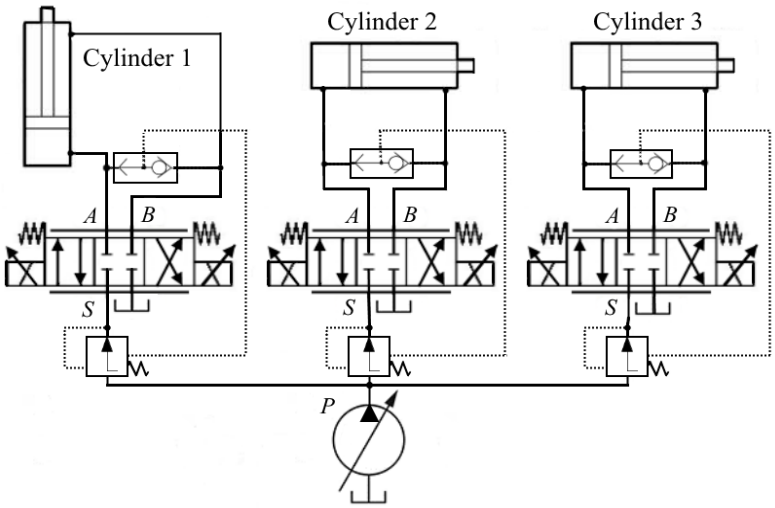

Figure 6.3: Hydraulic circuit used in the test simulation model

The directional control valves chosen for the modeling are of the type proportional 4/3 spool valve with positive overlap and the closed center position. This type of valves can be described by four connected turbulent orifices arranged in Wheatstone bridge. The flows Q_S, Q_A and Q_B in different positions of the spool proportional to input voltage U applied to valve solenoids can be considered as in (Galal Rabie, 2009):

$$\begin{cases} Q_A = C_v (U - U_d)^2 \sqrt{|p_S - p_A|}\,\mathrm{sgn}(p_S - p_A) \\ Q_B = C_v (U - U_d)^2 \sqrt{|p_B - p_T|}\,\mathrm{sgn}(p_B - p_T) \quad , U \geq U_d \\ Q_S = Q_A \end{cases}$$

$$\begin{cases} Q_A = -C_v (U + U_d)^2 \sqrt{|p_A - p_T|}\,\mathrm{sgn}(p_A - p_T) \\ Q_B = -C_v (U + U_d)^2 \sqrt{|p_S - p_B|}\,\mathrm{sgn}(p_S - p_B) \quad , U \leq -U_d \\ Q_S = -Q_B \end{cases} \tag{6.9}$$

$$Q_A = Q_B = 0 \quad , -U_d \leq U \leq U_d$$

where p_A, p_B, p_S and p_T are pressures in cylinder chambers, supply pressure and pressure in tank respectively; U_d is the voltage proportional to the positive overlap of the valve; C_v is the coefficient that accounts for the dependency between cross-section area of the valve orifices and supplied voltage. It is calculated from technical data of the control valve provided by its manufacturer. The flow equations (6.9) are written in unconventional way because of the factor $(U \pm U_d)^2$ that is used to model the effect of valve grooves with triangular cross-sections. In order to introduce the pressure compensators to the valves in model (6.9) the pressure drops $p_S - p_A$ and $p_S - p_B$ should be constant. The dynamics of pressure compensators are much faster than the dynamics of the system. Moreover, their introduction can increase the calculation load due to inherent nonlinearities. In order to keep the calculation time minimal, the dynamics of the pressure compensators are neglected. The dynamics of the proportional solenoids is described by a first order delay between the input voltage and feedback voltage from spool position.

According to (Galal Rabie, 2009), the flow Q_p supplied by pressure-compensated pump is calculated from the following equation:

$$\dot{Q}_p = \frac{k_p (p_{ref} - p_S) - Q_p}{t_p} \tag{6.10}$$

where k_p is the flow-pressure coefficient of the pump; t_p is the pump time constant; p_{ref} is the reference pressure of the pump.

The presented models of the hydraulic elements are interconnected with the following continuity equations (Handroos, 1990):

$$\dot{p}_S = \frac{B}{V}(Q_p - Q_S) \tag{6.11}$$

$$\dot{p}_A = \frac{B_A}{V_0 + A_1 x}(Q_A - A_1 v) \tag{6.12}$$

$$\dot{p}_B = \frac{B_B}{V_0 + A_2(H - x)}(A_2 v - Q_B) \tag{6.13}$$

Here B is the bulk modulus of the volume V emerged between the pump and corresponding directional valve. For each considered hydraulic cylinder Be_A and Be_B represent the corresponding bulk modulus of the volumes of the cylinder chambers; V_0 is the dead volume; x and v are the piston position and velocity respectively; H is the piston stroke.

The aforementioned hydraulic model augmented with the equations of dynamic model constitutes the simulation model that was used to simulate the motion of a real mobile crane. The data about the pressure and position of hydraulic cylinders obtained from the simulation model were used instead of sensor data of real machine.

6.2 Performance evaluation of simulation models

The simulation models used in remote surveillance applications should allow real-time simulations. If the execution of calculations based on these models takes more time than a time period being simulated, a user has to wait for simulation results. It has a negative impact on system scalability as less number of users can run their simulations during some fixed period of time. In the best case the simulations should run as fast as possible consuming less time than the simulated period. In (Malysheva et al., 2018) such kind of simulation is refereed to as "Faster-Than-Real-Time" (FTRT). Another benefit of FTRT simulations is that less computational power is required in order to run them in real time. It allows using such models with low-performance hardware, for example, on board the machines or in low-energy applications. Decrease of the requirements for computational power means that more simulation models can be run simultaneously in a cloud environment. As a result the system scalability can be improved.

In (Malysheva et al., 2018) the simulation models described in Chapter 6.1 were compared with the reference model created in commercially available software MATLAB/Simulink. The results of the comparison have shown that in a planar case the simulations that use INEF consume less than a half of real time period being simulated.

In order to evaluate the models in a spatial case, the same reference model as in (Malysheva et al., 2018) was used. This model considers the same hydraulic model as described in Chapter 6.1, the same geometry and inertial properties of the crane, but the dynamics is modelled in MATLAB/Simulink using the features provided by Simscape Multibody. The results provided by the reference model were compared with the simulation results of the model described in Chapter 6.1.

The motion of the crane during the time period of 5 seconds was simulated. Figure 6.4 shows the trajectory of the boom tip calculated by the two models. Unlike the trajectory, which is similar for both models the execution time substantially differs. Figure 6.5 presents the execution time of simulation program in three cases:

1. Reference model executed on PC with Intel®Core™ 2Duo CPU 2.26GHz, 4Gb RAM;
2. The model described in Chapter 6.1 (FTRT model) executed on the same PC;
3. The model described in Chapter 6.1 (FTRT model) executed on Raspberri Pi platform (1GHz ARM Cortex-A5 CPU, 256Mb RAM);

FTRT model allows simulations to be executed more than ten times faster than real-time on a PC and nearly two times faster on a Raspberry Pi platform. These results demonstrate that the models described in Chapter 6.1 can be used in real time simulations for remote surveillance on hydraulically actuated mobile working machines. High performance provided by these models makes it possible to use them in resource-constrained cloud applications.

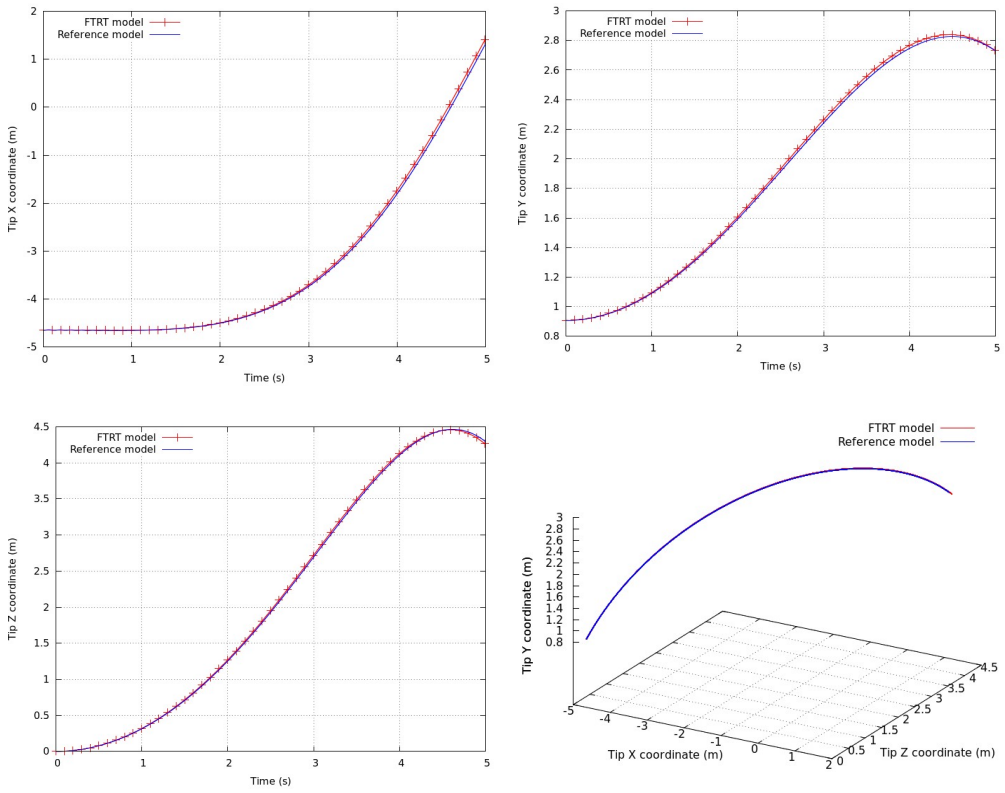

Figure 6.4: Trajectory of the boom tip in performance evaluation test

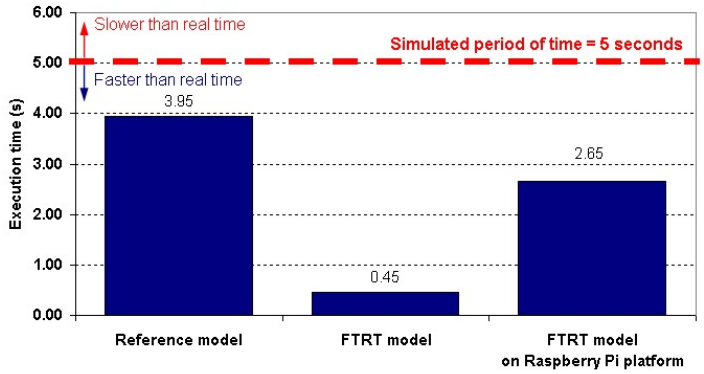

Figure 6.5: Execution time of the programs implementing different models

6.3 Test of remote surveillance of hydraulically actuated mobile machines

6.3.1 Software structure used in the test environment

An applicability of proposed methods was tested by gathering remotely the data generated by the simulation model. The test environment is presented in Figure 6.6.

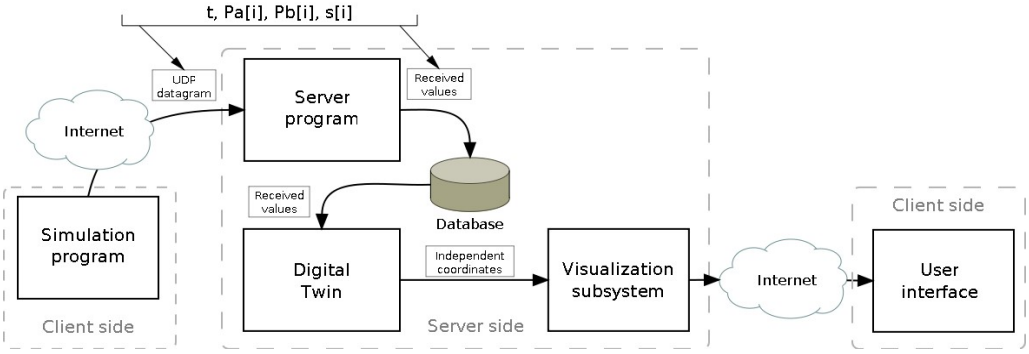

Figure 6.6: Test environment for remote surveillance

The simulation program calculates the values of the pressure and position of each cylinder using the simulation model. It is located on a separate computer which is presented as "Client side" in Figure 6.6. These values are interpreted as sensor data and transmitted to the server side which represents the cloud environment used for storage and processing of sensor data. In the experiment the server side is presented by a separate computer. The server program receives sensor data and saves them into the database. In the experiment sensor data are transmitted as text string composed of real numbers converted into text using decimal format with six decimal digits after the point.

Each sample represented by a text string is transmitted inside an UDP datagram. The sampling rate used in the experiment is 100 samples per second.

Another simulation model which is labelled "Digital Twin" in Figure 6.6 reads sensor data from the database. The digital twin utilizes the algorithm presented in Chapter 5.1.9 to calculate the values of position and pressure between the samples and also to calculate the forces produced by the cylinders. The calculated cylinder positions are used as independent coordinates to obtain the position of other parts of the machine. This is performed by the visualization subsystem. The user can watch the visualization results in the user interface. In the experiment the visualization results are presented through the web-interface in the browser which is run on a separate computer.

The software structure presented in Figure 6.6 can be recommended as a template for the implementation of remote surveillance in IIoT environment. A practice of storing telemetry data in a database for later processing is common for the IoT platforms. The digital twin simulating the machine dynamics have proven the performance suitable for real-time representation of machine motion even with the use of small computational resources. This property is useful for the cloud-based systems processing a large number of machines simultaneously. In-browser visualization of machine motion provides rich capabilities for the users while minimizing the load on cloud resources.

6.3.2 Experimental conditions and results

The simulation model used in the experiments utilizes the equations presented in Chapter 6.1 for modelling the motion of hydraulically actuated mobile crane. The motion is produced as a result of the control signal which is presented by a sequence of the values of input voltage for control valves. The input voltage of control valves for the cylinders of lifting and extension booms, as well as the cylinders of the slewing mechanism, used in the experiment, is shown in Figure 6.7.

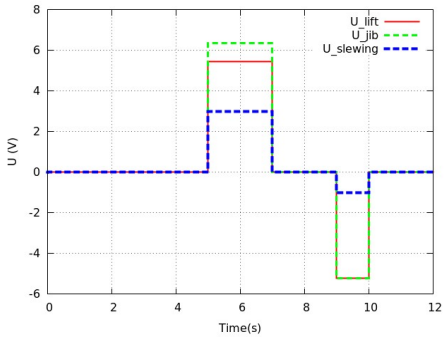

Figure 6.7: Input voltage of the control valves used in the experiment for remote surveillance

The input voltage presented in Figure 6.7 causes the motion of each boom actuated by the corresponding cylinder. Figure 6.8 shows the positions of the lifting boom and

extension boom calculated by the simulation model. The dashed line in Figure 6.8 represents the positions calculated by the digital twin with the use of received data.

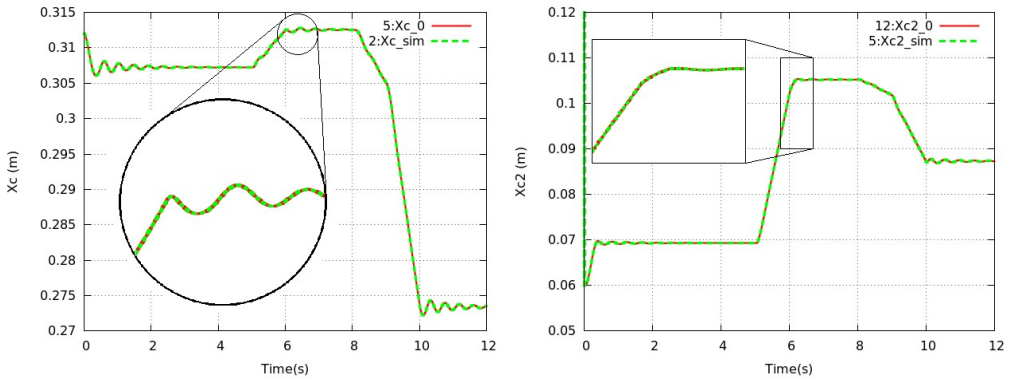

Figure 6.8: Calculated positions of the lifting boom and extension boom

The position of each cylinder calculated by the digital twin is just the interpolation of the sampled values generated by the simulation model and transmitted over the network. The accuracy of calculations is defined by the discretization error.

The cylinder forces are not transmitted to the digital twin. These forces are calculated using the dynamic model of the crane. The forces produced by the lifting cylinder and extension cylinder and calculated by the simulation model, as well as their values obtained by the digital twin from the dynamic model are presented in Figure 6.9.

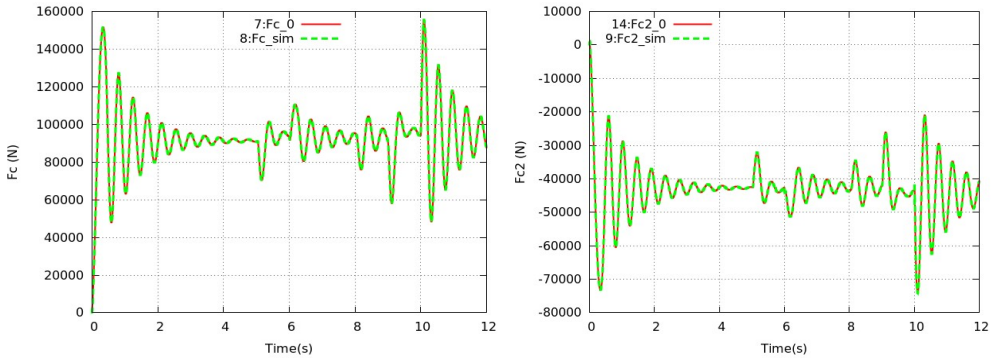

Figure 6.9: The forces produced by the cylinders of lifting boom and extension boom

The forces depend on the values of the pressure and velocity (5.6). The pressure values are restored by interpolation of the received data. The velocity of the cylinder rod is calculated using the dynamic model.

The position of each cylinder in any moment of time allows calculating the position of any part of the crane using the kinematic equations presented in Chapter 6.1.1. The

calculated values compose the position history of the machine and can be stored for the later analysis or presented to the user. Figure 6.10 shows one of the possible representations of the results implemented in the current work.

The motion of the crane is presented to the user by the animation displayed in the web browser. The background displayed during the animation represents the working environment of the machine. It can be generated automatically by several ways. For example, it can represent the 3D map obtained from the GPS coordinates of the machine. At the time of writing, the OpenStreetMap (OpenStreetMap, 2019) project provides the means for automatic generation of such a 3D map for free.

Figure 6.10: Example of the user interface for the remote surveillance application

The background presented in Figure 6.10 was obtained automatically from the Google Street View service and the GPS coordinates representing some point near the LUT University. The disadvantage of this approach is the requirement of using commercial services from Google in case when a large number of requests will be generated by a real system of remote surveillance.

6.3.3 Influence of sensor accuracy on the simulation results

In order to test the influence of sensor accuracy on the accuracy of simulation results, an artificial error was added to the input data of the remote surveillance model. The error was generated as a pseudo-random real number evenly distributed in the predefined range. It was added to the sensor data received from the client-side simulation program

and stored in the database (see Figure 6.6) and represented the noise introduced by the sensor inaccuracy. The same conditions of the experiment were used as described in the Chapter 6.3.2.

Figure 6.11 shows the examples of the position sensor data from the lift cylinder with two levels of an artificially generated error: 1% and 10%. In order to account for the measurement errors, some kind of smoothing should be applied to the input data. Two kinds of smoothing were tested: simple moving average (SMA) and Savitzky-Golay filter.

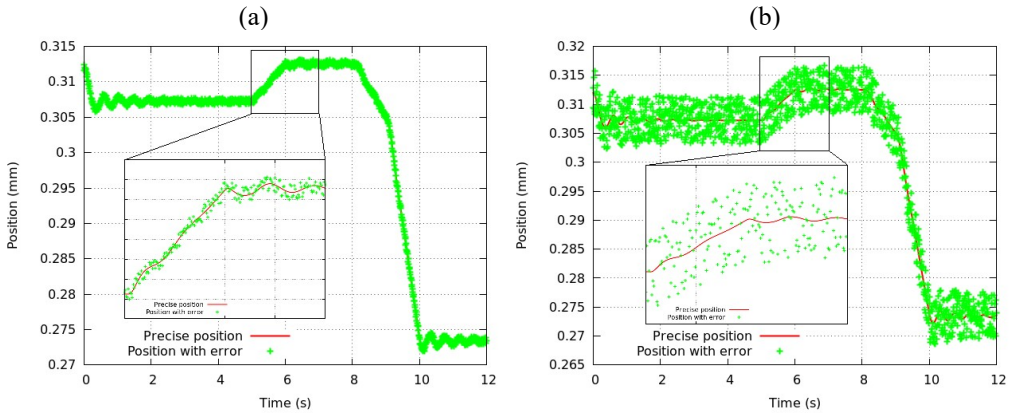

Figure 6.11: Examples of the sensor data with artificially added measurement errors

SMA is commonly used to smooth out short-term fluctuations in the input signal by creating a series of averages of different subsets of the full data set. A series of real numbers that represent a signal received during some period of time is scanned with the use of a "window" that selects a subset of data. The window moves (runs) along the data giving the name for the method. The first element of the moving average is obtained by taking the average of the initial N signal values where N is the size of the window. Then the window "shifts forward" modifying the subset by excluding the first number of it and including the next value in the series. The size of the window defines the ability of smoothing the signal fluctuations. The more is the window size the more fluctuations can be excluded from the signal. Figure 6.12 shows the dependency of the resulting normalized root mean square error of the calculation of the position and force of the lift and jib cylinders using remote surveillance technique on the window size used in SMA.

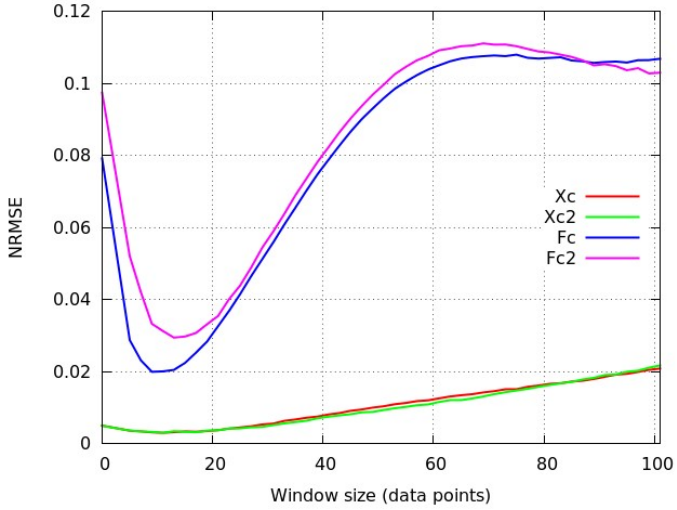

Figure 6.12: Normalized root mean square error of calculated positions and forces as a function of window size used in SMA for smoothing input data

The input data were generated using the pattern of the control signal shown in Figure 6.7. The data from the position and pressure sensors were disturbed by the artificially generated noise with the value of normalized root mean square error of 1%. The window size equal to 13 data points provides the minimum error of position and force calculation.

Savitzky-Golay filter (Savitzky and Golay, 1964) is a widely used digital filter that performs smoothing by fitting successive sub-sets of adjacent data points with a low-degree polynomial by the method of linear least squares. This method uses subsets of data of the length m (window size) to calculate m convolution coefficients that define a polynomial function fitting the subset. SMA can be considered as a special case of Savitzky-Golay filter when all convolution coefficients are equal to $1/m$. Figure 6.13 shows the dependency of the normalized root mean square error of calculation of the position and the force in each cylinder on the window size when the smoothing of data is performed by Savitzky-Golay filter. The window size equal to 27 data points provides the minimum error of position and force calculation.

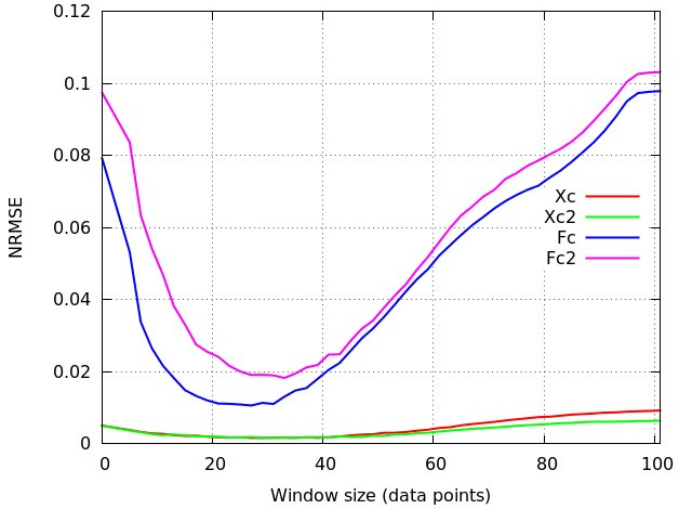

Figure 6.13: Normalized root mean square error of calculated positions and forces as a function of window size used in Savitzky-Golay filter

Three sets of experiments were performed: with the error added to the cylinder position data, the error added to the pressure data and the error added to the data obtained from all position and pressure sensors.

Figure 6.14 shows the dependency of the normalized root mean square error of calculation of the position and the force in each cylinder on the level of normalized root mean square error added to the position sensor data. The level of added error was being changed from 0 to 10% of the range of the measured parameter with the step of 0.1%. As the most of position and pressure sensors provide the accuracy which is less than 10% (the values less than 1% are common), this range covers the majority of sensors available on the market.

Input data containing the noise from the artificially introduced errors were smoothed by Savitzky-Golay filter with the optimum window size equal to 27 data points. The solid lines on the plots represent an approximation of the experimental data with the third degree polynomials created using an implementation of the nonlinear least-squares Marquardt-Levenberg algorithm provided by the gnuplot software.

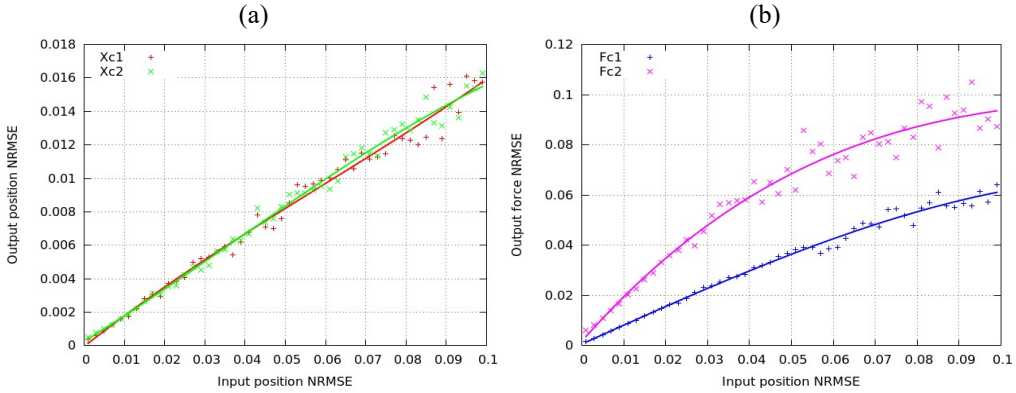

Figure 6.14: Dependency of the normalized root mean square error of calculation of the position and the force in each cylinder on the accuracy of the position sensors

Figure 6.15 shows the dependency of the normalized root mean square error of calculation of the position and the force in each cylinder on the level of normalized root mean square error added to the pressure sensor data. As the position of each cylinder is calculated by interpolation of the position sensor data, an artificial error added to the pressure data does not influence the accuracy of position calculation.

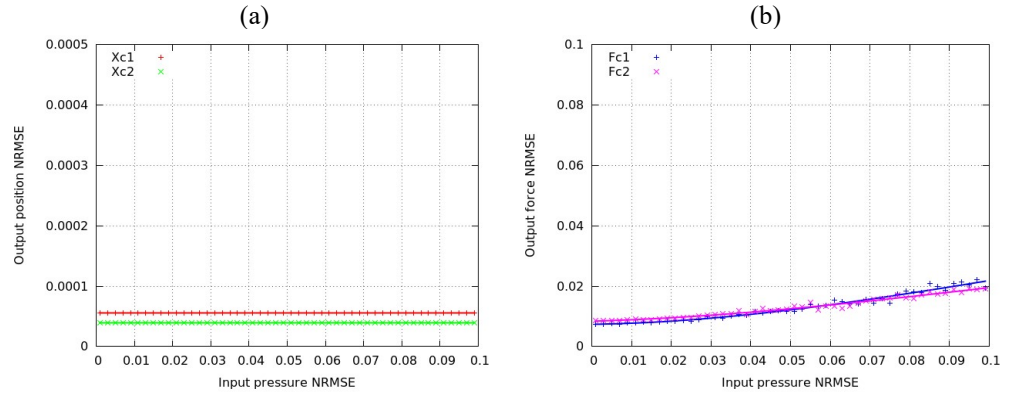

Figure 6.15: Dependency of the normalized root mean square error of calculation of the position and the force in each cylinder on the accuracy of the pressure sensors

The total error of calculation of the position and the force in each cylinder as a function of the maximum error introduced by the position and pressure sensors is presented in Figure 6.16. The experimental data show that the total error of calculation of positions and forces linearly increases with the decrease of sensor accuracy for the values of sensor accuracy that are most common for the commercially available sensors (less than 5%). In the experiment the total error of calculation did not exceed the error value introduced by the position and pressure sensors more than twice, if the sensor data were smoothed by Savitzky-Golay filter.

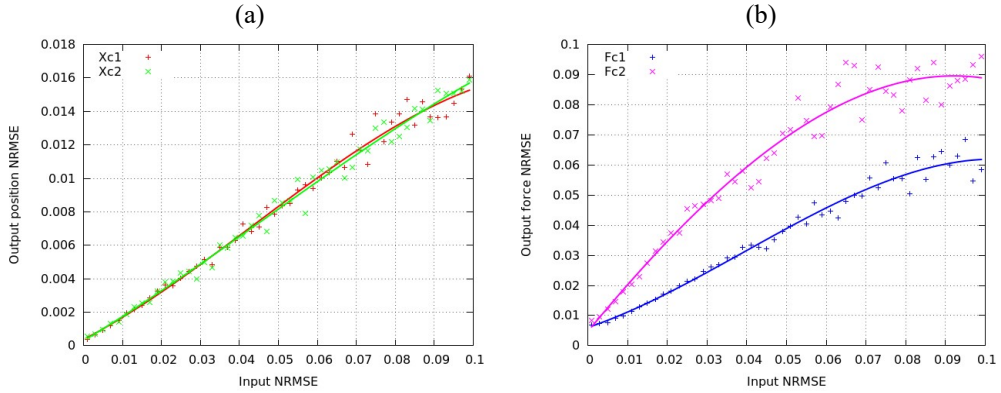

Figure 6.16: The total error of calculation of the position and the force in each cylinder as a function of a maximum error of input data from position and pressure sensors

In order to compare the influence of two popular smoothing methods, the simple moving average and Savitzky-Golay filter, on the accuracy of calculations, the experiments were performed using the optimum window sizes for both methods. Figure 6.17 shows the dependency of the normalized root mean square error of the cylinder position calculation on the accuracy of the position and pressure sensors for the cases of using two different smoothing methods. The window size for SMA was 13 data points and for Savitzky-Golay filter it was 27 data points.

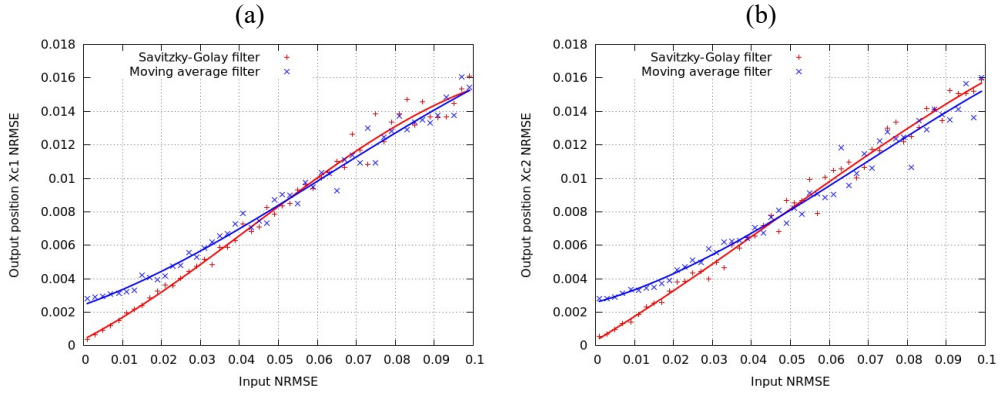

Figure 6.17: Dependency of the normalized root mean square error of the cylinder position calculation on the accuracy of the position and pressure sensors for two different smoothing methods

Figure 6.18 shows the dependency of the normalized root mean square error of the force calculation for each cylinder on the accuracy of the position and pressure sensors for the cases of using two different smoothing methods.

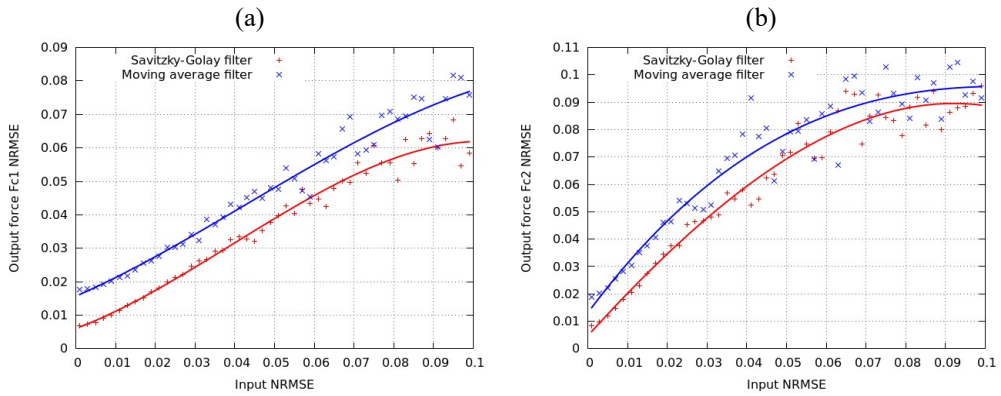

Figure 6.18: Dependency of the normalized root mean square error of the force calculation in each cylinder on the accuracy of the position and pressure sensors for two different smoothing methods

The use of Savitzky-Golay filter provided better accuracy of the force calculation than the SMA. For the values of the normalized root mean square error of input data less than 5% the use of Savitzky-Golay filter provided better accuracy for the position and the force calculation.

The forces calculated by the digital twin constitute the input for the calculation of the load history which can be used for the fatigue life estimation of the machine.

6.4 Test of Fatigue Life Estimation of Hydraulically Actuated Mobile Machines

To validate the proposed methods the simulation results were compared with the results of prior research performed by Professor Aki Mikkola in his dissertation (Mikkola, 1997) that was used as a reference. In his research A.Mikkola considered the mobile crane PATU 655. He measured the stress in two points of the crane boom during two work cycles of crane operation defined by the input voltage profile of control valves and positions of the booms. The measured results were compared with the simulated ones obtained with the help of ANSYS and ADAMS simulation software systems. Figure 6.19 shows the input voltage of the control valves considered in the research. The input voltage was the same for all hydraulic cylinders of the crane. The duration of work cycles was 8 seconds.

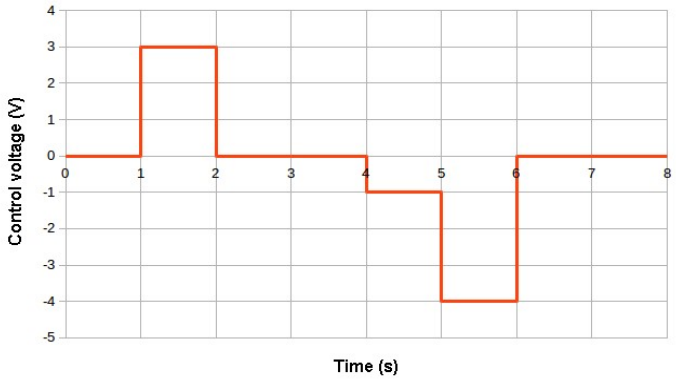

Figure 6.19: Input voltage of the control valves considered in the experiment

As the hydraulic circuit considered in A.Mikkola's work differs from that one simulated in this study, in order to compare the stress histories of the crane its motion was replicated. The initial position of each cylinder that was 0.307 m for the lift cylinder and 0.071 m for the jib cylinder and the load mass 370 kg were also replicated. The extension boom was considered to be fully extended. Figure 6.20 shows the motion of the crane considered in the reference research work as a result of supplying the input voltage profile described above to control valves.

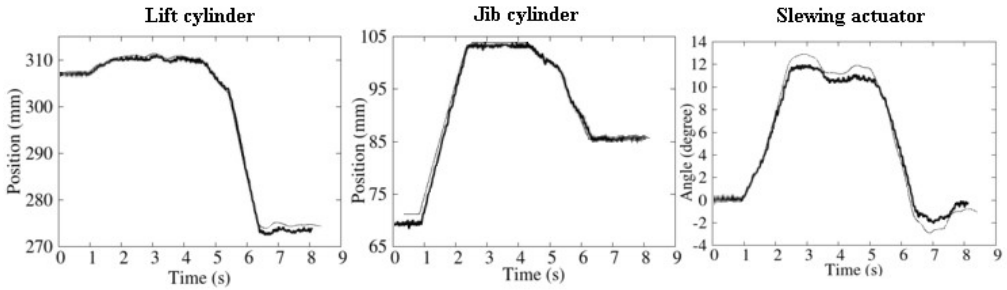

Figure 6.20: Positions of the actuators in the reference research

In order to simulate the similar motion of the crane as in the reference research work, the input voltage profile of control valves was adjusted to provide the similar positions of the lift cylinder and the jib cylinder. The values of input voltage applied to control valves were changed but the moments in time when the change was taking place remained the same as in the reference work. The slewing actuator in this study was simulated by applying the torque to the pillar of the crane. The torque was adjusted in time to provide the similar variation of angle as in the reference research work. Figure 6.21 shows the simulated positions of the actuators in comparison with the reference work.

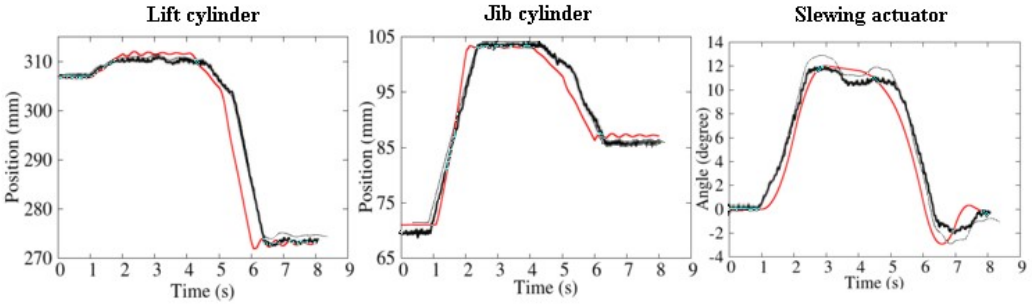

Figure 6.21: Simulated positions of the actuators in comparison with the reference values

The differences between the model used in this study and the model used in reference research work are summarized in the following table:

Table 6.1: Differences between the test model and reference model

Test model	**Reference model**
In multibody dynamics simulation the crane is considered as rigid multibody system	In multibody dynamics simulation the crane is considered as flexible multibody system
The hydraulic model includes valves equipped with pressure compensators	The hydraulic model includes counter balance valves
The pillar is rotated by direct application of the torque	The pillar is rotated by the slew mechanism consisting of two hydraulic cylinders
The load is considered to be a point rigidly attached at the tip of the crane	The load is considered to be a body connected to the crane via a rigid link and a universal joint
The booms are modelled as volumes using the 3D mesh constructed from the CAD drawings of the booms	The boom's structure is modelled using linear 6 DOF beam elements. The lift arm is modelled using 14 elements and the jib arm - using four elements.

The test environment used in the experiment is shown in Figure 6.22. Real-time sensor data were produced by the simulation program that was calculating the pressure in the piston side (*Pa*) and blind side (*Pb*) of the lift cylinder and jib cylinder as well as position of those cylinders. These data together with the timestamp and the value of the torque applied to the pillar were sent by UDP protocol to the server side. The server program was receiving the data and storing them to the database. The received values were used as an input for the simulation program that was reproducing the crane motion using remote surveillance technique. This program was calculating the constraint forces and acceleration of each boom that were used as boundary conditions for the FEA software. In the experiment the open-source FEA software "Code_Aster" (Electricité de France, 2017), (Aubry, 2013) was utilized. The values of stress in hot spots at every

time step were calculated as a result of finite-element analysis and were stored as a stress history to be used in estimation of fatigue life.

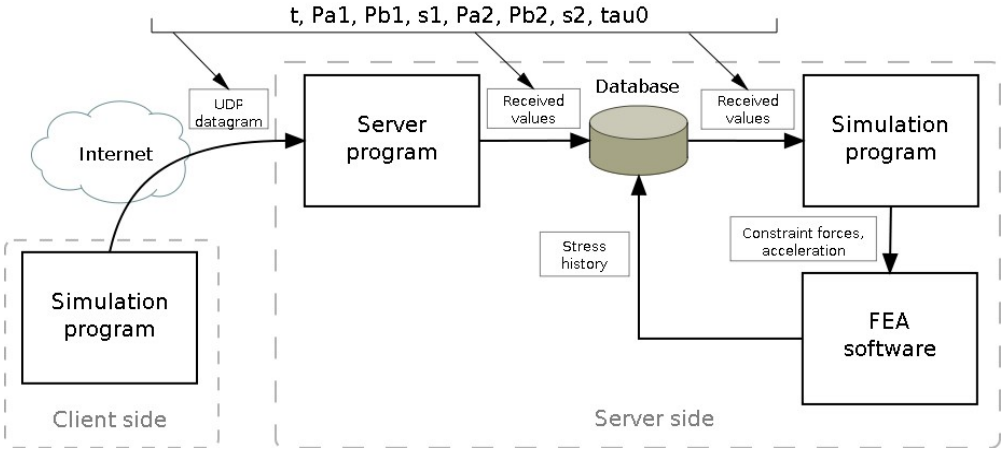

Figure 6.22: Test environment for fatigue life estimation

The software system structure presented in Figure 6.22 can be recommended as a basis for building cloud-based software for the fatigue life estimation. It extends the structure proposed for the remote surveillance in Chapter 6.3.1 with the module performing the stress history calculation. In the current work a straightforward approach of calculating the stress by FEA was utilized. The subsystem for the FEA can be based on open-source or proprietary software. The software developed by the open-source project Code_Aster that was used in the current work provides the finite-element analysis of different problems and includes some capabilities for the fatigue estimation. The experiments show that it can be used in the proposed infrastructure.

As FEA is known to be a resource consuming process, additional measures should be taken to minimize the time needed for stress calculation. These could be tuning the mesh used in FEA, selecting the hot spots for stress calculation, or preprocessing the load history to decrease the volume of input data for the FEA. A promising approach is the use of an artificial neural network for stress approximation. The neural network should be constructed for each model of mobile working machines and can be trained on the stress data calculated from the load history with FEA software. The field data being gathered from many machines by the lifecycle support system provide a rich dataset for training. The use of artificial neural networks can drastically reduce the time needed for stress calculation. Unlike FEA, the neural networks can easily be utilized in real-time. In order to preserve accuracy, the neural network can be used for rough approximation of the stress levels with the later processing of the machines that have the highest stress levels by the FEA.

The stress history calculated in the experiment was compared with the results obtained in the reference research work. The stress was calculated in two points on the lift boom,

marked A and B in Figure 6.23. In these points the stress gauges were located in the reference work. The points were located on the upper flange of the profile, their distance from the edge of the profile was 1/4 of the total width of the profile.

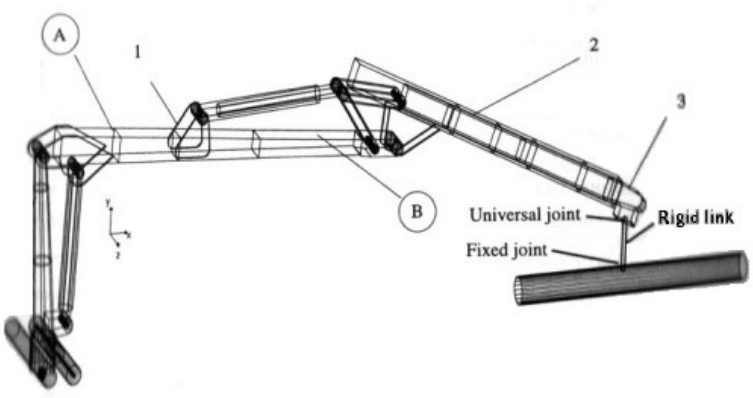

Figure 6.23: Location of the points in which the stress history was calculated (Mikkola, 1997)

The strain gauges provide the scalar values of composition stress, the different components of which cannot be separated. In order to compare these values with simulation results, that consist of individual stress components, a superposition of simulated stress components onto each other was performed in the reference research work to produce the composition stress. As the reference work does not provide any equation for calculation of the composition stress, the equivalent stress was used as a common way for calculation of scalar value from the stress tensor. The equivalent stress was calculated the following way (Bishop and Sherratt, 2000):

$$\sigma_{eq} = \sqrt{\frac{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}{2}} \tag{6.14}$$

where σ_1, σ_2 and σ_3 are the components of the principal stress.

The stress histories consisting of the composition stress measured and calculated in the reference work and the equivalent stress calculated in this study are presented in Figure 6.24. The bold black lines represent the stress levels measured experimentally in the reference work. The bold grey lines represent the stress levels calculated in the reference work. Red and blue lines show equivalent stress calculated in this study.

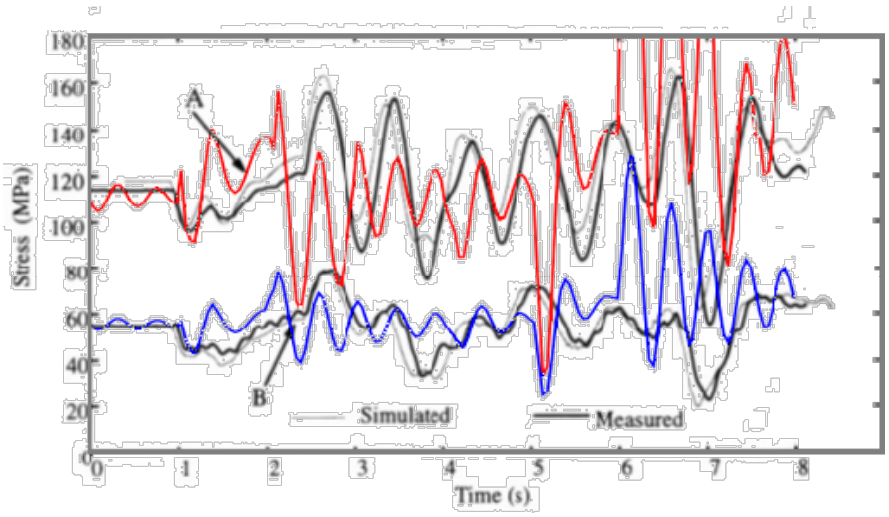

Figure 6.24: The stress histories obtained in the reference work and in the current study

The differences between the hydraulic models mentioned in Table 6.1 lead to the higher frequency of stress oscillations in the model created in this study. Large amplitude of oscillations during the last two seconds of simulation is caused by low damping of the model created in this study. During this period of simulation the control valves are closed and system dynamics is defined by the damping introduced by hydraulic system. As pressure compensators are known to have low damping (Andersson, 1997), the system oscillates.

The stress levels calculated for the first second of the time period being simulated should be discussed separately. As the crane does not move during this period, the stress levels are defined by the static load produced by actuator forces and constraint forces. Small fluctuations of stress levels calculated by the test model are caused by the rigidity of the model. They can be eliminated if longer transient period is used prior to stress calculation in order to achieve the stationary state. The mean levels of calculated stress in static mode for the test model and reference model differ by 2-3%. This result allows the conclusion that the actuator forces and constraint forces calculated with the remote surveillance technique can be used as input for FEA-based stress calculation.

This allows proposing the described approach for on-line monitoring and prediction of fatigue life of hydraulically actuated heavy equipment. The use of precise simulation models is necessary as the accuracy of simulation of each component is important for the total applicability of simulation results.

7 Conclusions

This study proposes the methods for utilization of the recent advantages in computing and networking, known as Digital Twin and Internet of Things concepts, in supporting the lifecycle of hydraulically actuated mobile working machines. Two application areas are proposed and developed: remote surveillance of mobile working machines and estimation of fatigue life of their mechanical structure. The issues associated with the transmission, storage and processing of sensor data in solving the tasks of running physics-based simulations are considered. The proof-of-concept software and environments for IoT-enabled solutions implementing these tasks are created and tested. The results of the study show that the proposed methods can be used for remote monitoring of hydraulically actuated mobile working machines, for the calculation of forces acting on its mechanical structure and of stress resulting from that action, and for the estimation of fatigue life on the basis of calculated stress history. The information provided by the use of the proposed methods helps to extend the lifetime of working machines by controlling their fatigue damage, and also to increase the efficiency of their utilization by monitoring the motion and the forces produced by the machine. The method for fatigue life estimation can be useful for remanufacturing, as it provides the information about the fatigue damage accumulated by each part of mechanical structure of the machine along its operation.

The contributions of this study include:

- A review of research publications dedicated to the Internet of Things area made with the automatic text analysis techniques;
- Determination of the main problems of running physics-based digital twins in the Internet of Things environment and proposal of the techniques that facilitate solution of these problems;
- A method for the remote surveillance on the hydraulically actuated mobile working machines based on simulation model of the machine;
- A method for fatigue life estimation of hydraulically actuated mobile working machines based on simulation model of the machine;
- A structure of the software system for the implementation of the developed methods in the Internet of Things environment.

The future work on the topic should involve the experiments with the real machines to clarify the limitations of the proposed methods more thoroughly. The tests should also be made using the data transmission technologies and protocols used by the existing IoT

platforms to define the ways of development for better adoption of the proposed approach.

The current study uses the Iterative Newton-Euler formulation to build the dynamic model of the machine. The suitability of other formulations as well as the models created with existing software and physics engines should also be investigated.

The fatigue estimation method should be extended with the usage of several fatigue estimation methodologies on different stages of the machine lifecycle as described in the chapter 5.2.3. The experiments with the models which consider the flexibility of the mechanical structure should be performed.

In order to accelerate the calculation of stress in the mechanical structure of the machines, several approaches could be tested. The use of artificial neural networks looks promising for solving this task. The procedure for determining the parameters of the neural network optimized for each particular machine type should be developed. Another possible solution would be an application of the lookup tables for the fast estimation of the stress by utilizing the results of the previously calculated stress distribution obtained in the similar load conditions. This could be a suitable approach for the Big Data oriented systems as it combines the physics-based simulations with the capabilities of existing systems for storing and fast processing of multiple data streams coming from many machines.

Thus, the use of the well-known and verified methods for physics-based simulation together with the capabilities of current information technologies opens the wide opportunities in the areas of Internet of Things and Digital Twin. The current work has considered some of the possible applications, but many other of them should be considered and developed in the nearest future.

References

Abolfazli, S., Sanaei, Z., Wong, S.Y., Tabassi, A., Rosen, S. (2015). Throughput measurement in 4G wireless data networks: Performance evaluation and validation. In: *Proceedings of the IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pp. 27–32. Langkawi, Malaysia.

Abrahamsson, P., Helmer, S., Phaphoom, N., Nicolodi, L., Preda, N., Miori, L., Angriman, M., Rikkilä, J., Wang, X., Hamily, K., Bugoloni, S. (2013). Affordable and Energy-Efficient Cloud Computing Clusters: The Bolzano Raspberry Pi Cloud Cluster Experiment. *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, 2, pp. 170–175.

Åman, R., Handroos, H., Eskola, T. (2008). Computationally Efficient Two-Regime Flow Orifice Model for Real-Time Simulation. *Simulation Modelling Practice and Theory*, 16(8), pp. 945–961.

Andersson, B.R. (1997). Valves contribution to system damping. In: *The 5th Scandinavian International Conference on Fluid Power (SICFP'97)*, Linköping, Sweden.

Andersson, S., Söderberg, A., Björklund, S. (2007). Friction models for sliding dry, boundary and mixed lubricated contacts". *Tribology int.*, 40(4), pp.580–587.

Armstrong, W. (1979). Recursive Solution to the Equations of Motion of an n-Link Manipulator. In: *Proceedings of 5th World Congress on Theory of Machines and Mechanisms, (Montreal)*, pp. 1343–1346.

Arthur, D. and Vassilvitskii, S. (2007). K-Means++: The Advantages of Careful Seeding. In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 8, pp. 1027–1035.

ARXIV (2018). General Information About arXiv [Retrieved August 28, 2018], https://arxiv.org/help/general

Aubry, J-P. (2013). *Beginning with Code_Aster. A Practical Introduction to Finite Element Method Using Code_Aster, Gmsh and Salome.* Framasoft.

Axin, M. and Krus, P. (2013). Design rules for high damping in mobile hydraulic systems. In: *Proceedings of the 13th Scandinavian International Conference on Fluid Power*, pp. 13–20. Linköping, Sweden.

Baharudin, M.E. (2016). *Real-time simulation of multibody systems with applications for working mobile vehicles*. PhD Thesis, Lappeenranta University of Technology, Lappeenranta, Finland.

Balafoutis, C., Patel, R., Misra, P. (1988). Efficient Modeling and Computation of Manipulator Dynamics Using Orthogonal Cartesian Tensors. *IEEE Journal of Robotics and Automation*, 4, pp. 665–676.

Będkowski, W. (2014). Assessment of the fatigue life of machine components under service loading – a review of selected problems. *J. Theor. Appl. Mech.*, 52, pp. 443–58.

Bishop, N.W.M. and Sherratt, F. (2000). *Finite element based fatigue calculations.* Glasgow, Scotland: NAFEMS.

Blei, D. (2012). Probabilistic Topic Models. *Communications of the ACM*, 55(4), pp. 77–84.

Blei, D., Griffiths, T., Jordan, M. (2010). The nested Chinese restaurant process and Bayesian non parametric inference of topic hierarchies. *J.ACM*, 57(2), pp. 1–30.

Blei, D., Ng, A., Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3, pp. 993–1022.

Bokrantz, J., Skoogh, A., Berlin, C., Stahre, J. (2017). Maintenance in digitalised manufacturing: Delphi-based scenarios for 2030. *International Journal of Production Economics*, 191, pp. 154–169.

Brandl, H., Johanni, R., Otter, M. (1986). Very Efficient Algorithm for the Simulation of Robots and Similar Multibody Systems Without Inversion of the Mass Matrix. *IFAC Proceedings Volumes*, 19(14), pp. 95–100.

Chen, P., Zhang, N.L., Liu, T., Poon, L., Chen, Z., Khawar, F. (2017). Latent tree models for hierarchical topic detection. *Artif. Intell.*, 250, pp. 105–124.

Craig, J. (2005). *Introduction to robotics: mechanics and control*. Upper Saddle River, NJ, USA: Pearson/Prentice Hall.

Collins, J. (1993). *Failure of Materials in Mechanical Design: Analysis, Prediction, Prevention*. New York, USA: John Wiley & Sons Inc.

Danicic, D., Sedmak, S., Ignjatovic, D., Mitrovic, S. (2014). Bucket Wheel Excavator Damage by Fatigue Fracture – Case Study. *Procedia Materials Science*, 3, pp. 1723–1728.

Diez-Olivan, A., Del Ser, J., Galar, D., Sierra, B. (2019). Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0. *Information Fusion*, 50, pp. 92–111.

Ding, L., Fang, W., Luo, H., Love, P., Zhong, B., Ouyang, X. (2018). A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory. *Automation in Construction*, 86, pp. 118–124.

Egorova, D. and Zhidchenko, V. (2015). Visual Parallel Programming as PaaS cloud service with Graph-Symbolic Programming Technology. In: *Proceedings of the Institute for System Programming*, 27(3), pp. 47–56. Samara: ISP RAN.

Electricité de France (2017). Code_Aster® Analysis of Structures and Thermomechanics for Studies and Research [Retrieved Aug.15, 2018], https://www.code-aster.org/V2/UPLOAD/DOC/Presentation/ plaquette_aster_en.pdf

Ellis, G. (2012). *Control system design guide: Using your computer to understand and diagnose feedback controllers (4th ed.)*. Butterworth-Heinemann.

Erikstad, S.O. (2017). Merging Physics, Big Data Analytics and Simulation for the Next-Generation Digital Twins. In: *HIPER 17: 11th Symposium on High-Performance Marine Vehicles*, pp. 139–149.

Fang, Y. (2016). *Real-time safety assistance to improve operators' situation awareness in crane lifting operations*. PhD dissertation, Georgia Institute of Technology.

Featherstone, R. and Orin, D. (2000). Robot Dynamics: Equations and Algorithms. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, pp. 826–834. San Francisco, CA, USA.

Featherstone, R. (2008). *Rigid Body Dynamics Algorithms*. Canberra, Australia: Springer Science+Business Media.

Featherstone, R. (1983). The Calculation of Robot Dynamics using Articulated-Body Inertias. *Int. J. Robotics Research*, 2(1), pp. 13–30.

Featherstone, R. (1987). *Robot Dynamics Algorithms*. Boston/Dordrecht/Lancaster: Kluwer Academic Publishers.

Filho, S.E., Burlamaqui, A.M., Aroca, R.V., Goncalves, L.M. (2017). NPi-Cluster: A Low Power Energy-Proportional Computing Cluster Architecture. *IEEE Access*, 5, pp. 16297–16313.

Främling, K., Holmström, J., Ala-Risku, T., Kärkkäinen, M. (2003). Product agents for handling information about physical objects. *TKO-B*, 153/03, pp. 20.

Fujitake, M. and Yoshimi, T. (2017). Estimation system of construction equipment from field image by combination learning of its parts. In: *11th Asian Control Conference (ASCC), Gold Coast, QLD*, pp. 1672–1676.

Galal Rabie, M. (2009). *Fluid power engineering*. New York, USA: McGraw-Hill.

Gomes, C., Thule, C., Broman, D., Larsen, P. G., Vangheluwe, H. (2018). Co-simulation: a Survey. *ACM Comput. Surv.*, 1(1), 35 pages.

Gottvald, J. (2010). The calculation and measurement of the natural frequencies of the bucket wheel excavator SCHRS 1320/4X30. *Transport*, 25(3), pp. 269–277.

Grieves, M. and Vickers, J. (2017). Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems. *Transdisciplinary Perspectives on Complex Systems*, ed. F. J. Kahlen, S. Flumerfelt et al. (Cham: Springer), pp. 85–113.

Groth, C., Porziani, S., Biancolini, M., Costa, E., Celi, S., Capellini, K., Rochette, M., Morgenthaler, V. (2018). The medical digital twin assisted by Reduced Order Models and Mesh Morphing. In: *International CAE Conference* 2018, Vicenza, Italy.

Hajji, W., Tso, F.P. (2016). Understanding the Performance of Low Power Raspberry Pi Cloud for Big Data. *Electronics* 2016, 5, 29. doi:10.3390/electronics5020029.

Handroos, H. (1990). *Methods for combining a theoretical and an empirical approach in modelling pressure and flow control valves for CAE-programs for fluid power circuits*. PhD thesis, Acta Polytechnica Scandinavica, Mechanical Engineering series, 96, Helsinki, Finland.

Handroos, H., Keskinen, E., Vilenius M. (1993). Effect of design parameters on vibrational behaviour of a counter balance valve equipped hydraulic crane. In: *2nd JHPS International Conference on Fluid Power*, Tokyo, Japan.

Handroos, H. and Halme, J. (1996). Semi-empirical model for a counter balance valve. In: *Third JHPS International Symposium on Fluid Power*, Yokohama, Japan.

Haug, E. J. (1989). *Computer Aided Kinematics and Dynamics of Mechanical Systems. Vol. 1: Basic Methods.* Needham Heights, MA, USA: Allyn & Bacon, Inc.

He, X. and Goldenberg, A. (1989). An Algorithm for Efficient Computation of Dynamics of Robotic Manipulators. In: *Proceedings of Fourth International Conference on Advanced Robotics, (Columbus, OH)*, pp. 175–188.

Hollerbach, J. (1980). A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-10(11), pp. 730–736.

IEEE (2018). IEEE Standard for Floating-Point Arithmetic. In: *IEEE Std 754-2008*, pp. 1–70, 29 Aug. 2008. doi: 10.1109/IEEESTD.2008.4610935

ITU-T (2012). Overview of the Internet of things, Recommendation ITU-T Y.2060 [Retrieved May.20, 2019], http://handle.itu.int/11.1002/1000/11559

Johnston, S., Basford, P., Perkins, C., Herry, H., Tso, F.P., Pezaros, D., Mullins, R., Yoneki, E., Cox, S., Singer, J. (2018). Commodity single board computer clusters and their applications. *Future Generation Computer Systems*. 89. 10.1016/j.future.2018.06.048.

Kai Qi, Weixiong Wang, Xinhua Wang, Aihua Jiang, Boqing Liu, Zhongliang Guo, Jin Liu (2013). Safety Assessment and Fatigue Life analysis of Aged Crane Structures. In: *Proceedings of the 13th International Conference on Fracture 2013 (ICF-13),* 6, pp. 4648–4652.

Kahn, M. and Roth, B. (1971). The Near Minimumtime Control of Open-loop Articulated Kinematic Chains. *Journal of Dynamic Systems, Measurement, and Control*, 93, pp. 164–172.

Keskinen, E.K., Saarinen, H., Tuokko, R. (1993). Mechanics of seal friction with special reference to hydraulic cylinder applications. In: *The Third Scandinavian International Conference on Fluid Power*, Linköping, Sweden.

Keyes, D. E., McInnes, L. C., Woodward, C., Gropp, W., Myra, E., Pernice, M., … Wohlmuth, B. (2013). Multiphysics simulations: Challenges and opportunities. *The International Journal of High Performance Computing Applications*, 27(1), pp. 4–83.

Khodadadi Sadabadi, K., Shahbakhti, M. (2016). Dynamic Modelling and Controller Design of Combustion Phasing for an RCCI Engine. In: *ASME Dynamic Systems and Control Conference*, Minneapolis, MN, USA.

Kotelnikov, V.A. (1933). On the transmission capacity of "ether" and wire in electrocommunications, (English translation), Izd. Red. Upr. Svyazzi RKKA, Reprint in Modern Sampling Theory: Mathematics and Applications, Editors: J. J. Benedetto und PJSG Ferreira, Birkhauser (Boston) 2000. [Retrieved April 6, 2019], http://ict.open.ac.uk/classics/1.pdf

Kovanen, J. (2003). *Improving dynamic characteristics of open-loop controlled log crane*. PhD Thesis, Lappeenranta University of Technology, Lappeenranta, Finland.

Kovartsev, A. (2011). *Vychislitelnaya matematika [Computational mathematics]*. Samara, Russia: Ofort.

Kovartsev, A., Zhidchenko, V., Popova-Kovartseva, D. (2017). *Metody i tehnologii visualnogo programmirovaniya [Methods and technologies for visual programming]*. Samara, Russia: Ofort.

Lapid, M., Steen, T., Schild, C. (2018). Troubleshooting and Analyzing Network Booting Challenges Using a Raspberry Pi Testbed. HPC Mini-Showcase [Retrieved March 16, 2019], https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-18-27425

Lazakis, I., Dikis, K., Michala, A.L., Theotokatos, G. (2016). Advanced Ship Systems Condition Monitoring for Enhanced Inspection, Maintenance and Decision Making in Ship Operations. *Transportation Research Procedia*, 14, pp. 1679–1688.

Li, W., McCallum, A. (2006). Pachinko allocation: DAG-structured mixture models of topic correlations. In: *International Conference on Machine Learning*, pp. 577–584.

Liftech Consultants Inc. (2002). Predicting and prolonging the life of used cranes [Retrieved August 19, 2018], http://www.liftech.net/wp-content/uploads/2014/12/Predicting-and-Prolonging-the-Life-of-Used-Cranes.pdf

Liu, T., Zhang, N.L., Chen, P. (2014). Hierarchical latent tree analysis for topic detection. *ECML/PKDD*, pp. 256–272.

Lucia, D.J., Beran, P.S., Silva, W.A. (2004). Reduced-order modeling: new approaches for computational physics. *Progress in Aerospace Sciences*, 40(1–2), pp. 51–117.

Luostarinen, L., Åman, R., Handroos H. (2014). Development of control interface for HIL simulation of electro-hydraulic energy converter. *Int. Rev. Modelling and Simulations (IREMOS)*, 7(4), p. 653.

Luh, J.Y.S., Walker, M., Paul, R. (1980). On-line computational scheme for mechanical manipulators. *Trans. of the ASME J. of Dynamic Syst., Meas., and Control*, 102(2), pp. 69–76.

Malysheva, I., Handroos, H., Zhidchenko, V., Kovartsev, A. (2018). Faster than real-time simulation of a hydraulically actuated log crane. In: *2018 Global Fluid Power Society PhD Symposium (GFPS)*. doi: 10.1109/GFPS.2018.8472405

Matsuishi, M. and Endo, T. (1968). Fatigue of metals subjected to varying stress. *Japan Society of Mechanical Engineers*, Fukuoka, Japan.

McMillan, S. and Orin, D. (1995). Efficient Computation of Articulated-Body Inertias Using Successive Axial Screws. *IEEE Trans. on Robotics and Automation*, 11, pp. 606–611.

Mikkola, A. M. (1997). *Studies on fatigue damage in a hydraulically driven boom system using virtual prototype simulations*. Doctoral thesis, Lappeenranta University of Technology, Lappeenranta, Finland.

Minav, T., Hänninen, H., Sinkkonen, A., Laurila, L., Pyrhönen, J. (2014). Electric or Hydraulic Energy Recovery Systems in a Reach Truck – A Comparison. *Strojniški vestnik - Journal of Mechanical Engineering*, 60(4), pp. 232–240.

Miner, M. (1945). Cumulative damage in fatigue. *Journal of Applied Mechanics, Trans. ASME*, 67, A159–A164.

Moradi Afrapoli, A. (2018). *A Hybrid Simulation and Optimization Approach towards Truck Dispatching Problem in Surface Mines*. PhD thesis, University of Alberta, Edmonton, Canada.

Nazarov, A. V., Kozyrev, G. I., Shitov, I. V., Obruchenkov, V.P., Drevin, A.V., Kraskin V.B., Kudryakov, S.G., Petrov A.I., Sokolov S.M., Yakimov, V.L., Loskutov A.I. (2007). *Sovremennaya telemetriya v teorii i na praktike [Modern Telemetry in Theory and Practice]*. St. Petersburg, Russia: Nauka i Tekhnika.

Negri, E., Fumagalli, L., Macchi, M. (2017). A Review of the Roles of Digital Twin in CPS-based Production Systems. *Procedia Manufacturing*, 11, pp. 939–948.

NLTK (2018). Natural Language Toolkit [Retrieved August 28, 2018], http://www.nltk.org/

Nyquist, H. (1928). Certain topics in telegraph transmission theory, Trans. AIEE. 47(2), pp. 617–644. doi:10.1109/t-aiee.1928.5055024. Reprint as classic paper in: Proc. IEEE, 90, No. 2, 2002 [Retrieved April 6, 2019], https://web.archive.org/web/20130926031230/http://www.ieee.org/publications_standards/publications/proceedings/nyquist.pdf

OpenStreetMap (2019). *OpenStreetMap project* [Retrieved April 6, 2019], https://www.openstreetmap.org

Orin, D., McGhee, R., Vukobratovic, M., Hartoch, G. (1979). Kinematic and Kinetic Analysis of Open-chain Linkages Utilizing Newton-Euler Methods. *Mathematical Biosciences*, 43, pp. 107–130.

Ou, Z., Pang, B., Deng, Y., Nurminen, J.K., Ylä-Jääski, A., Hui, P. (2012). Energy- and Cost-Efficiency Analysis of ARM-Based Clusters. In: *12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, pp. 115–123.

Paisley, J., Wang, C., Blei, D., Jordan, M. (2015). Nested Hierarchical Dirichlet Processes. *IEEE Trans. Pattern Anal. Mach. Intell*., 37(2), pp. 256–270.

Palmgren, A. (1924). Die Lebensdauer von Kugellagern. *VDI Zeitschrift*, 339–341.

Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, pp. 2825–2830.

Piche, R., Ellman, A. (1994). Numerical Integration of Fluid Power Circuit Models Using Two-Stage Semi-Implicit Runge-Kutta Methods. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 208, pp. 167–175.

Prytz, R. (2014). *Machine learning methods for vehicle predictive maintenance using off-board and on-board data*. Licentiate thesis, Halmstad University, Sweden.

Rakin, M., Arsić, M., Bošnjak, S.M., Gnjatović, N.B., Međo, B. (2013). Integrity assessment of bucket wheel excavator welded structures by using the single selection method. *Tehnički vjesnik*, 20(5), pp. 811–816.

Rantalainen, T. (2012). *Simulation of structural stress history based on dynamic analysis*. PhD Thesis, Lappeenranta University of Technology, Lappeenranta, Finland.

RFC8085 (2018). UDP Usage Guidelines. [Retrieved March 20, 2019], https://tools.ietf.org/html/rfc8085

Richard, H.A., Sander, M., Fulland, M., Kullmer, G. (2008). Development of fatigue crack growth in real structures. *Engineering Fracture Mechanics*, 75(3–4), pp. 331–340.

Rögnvaldsson, T., Nowaczyk, S., Byttner, S., Prytz, R., Svensson, M. (2018). Self-monitoring for maintenance of vehicle fleets, *Data Mining and Knowledge Discovery*, 32(2), pp. 344–384.

Safari Khatouni, A. (2018). *Experimentation and Characterization of Mobile Broadband Networks*. PhD dissertation. [Retrieved March 20, 2019], http://hdl.handle.net/11583/2708886

Savitzky, A. and Golay M. J. E. (1964). Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, 36(8), pp. 1627–1639.

Shabana, A. A. (2010). *Computational Dynamics, 3rd Edition*. UK: Wiley.

Shannon, C.E. (1949). Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers*, 37 (1), pp. 10–21. doi:10.1109/jrproc.1949.232969. Reprint as classic paper in: Proc. IEEE, 86, No. 2, (1998) [Retrieved April 6, 2019], https://web.archive.org/web/20100208112344/http://www.stanford.edu/class/ee104/shannonpaper.pdf

Sørensen, J.K. (2016). *Reduction of Oscillations in Hydraulically Actuated Knuckle Boom Cranes*. PhD dissertation, University of Agder, Norway.

Srinivasan, K., Chang, C., Huang, C., Chang, M., Sharma, A., Ankur, A. (2018). An Efficient Implementation of Mobile Raspberry Pi Hadoop Clusters for Robust and Augmented Computing Performance. *J Inf Process Syst*, 14(4), pp. 989–1009.

Steinbach, M., Karypis, G., Kumar, V. (2000). A comparison of document clustering techniques. In: *KDD Workshop on Text Mining, Boston*, 400, pp. 525–526.

Stepanenko, Y. and Vukobratovic, M. (1976). Dynamics of Articulated Open-chain Active Mechanisms. *Mathematical Biosciences*, 28, pp. 137–170.

Tso, F.P., White, D., Jouet, S., Singer, J., Pezaros, D. (2013). The Glasgow Raspberry Pi Cloud: A Scale Model for Cloud Computing Infrastructures. In: *Proceedings of the 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops*, pp. 108–112.

Uicker, J. (1967). Dynamic Force Analysis of Spatial Linkages. *Transactions of the ASME Journal of Applied Mechanics*, 34, pp. 418–424.

Vacca, A. (2018). Energy Efficiency and Controllability of Fluid Power Systems. *Energies, MDPI, Open Access Journal*, 11(5), pp. 1–6.

VanderPlas J. (2016). *Python Data Science Handbook. Essential Tools for Working with Data.* O'Reilly Media, Inc.

Vereshchagin, A. (1974). Computer Simulation of the Dynamics of Complicated Mechanisms of Robot Manipulators. *Engineering Cybernetics*, 6, pp. 65–70.

Viitanen, T. and Siljander, A. (Eds.) (2019). *A Review of Aeronautical Fatique Investigations in Finland April 2017 - March 2019*. ICAF National Review - Finland, no. VTT-CR-00352-19, VTT Technical Research Centre of Finland.

Walker, M. and Orin, D. (1982). Efficient Dynamic Computer Simulation of Robotic Mechanisms. *Trans. ASME, J. Dynamic Systems, Measurement & Control*, 104, pp. 205–211.

Whittaker, J.M. (1935). *Interpolatory Function Theory*. Cambridge, UK: Cambridge University Press.

Wu, Y., Li, W., Yang, P. (2015). A Study of Fatigue Remaining Useful Life Assessment for Construction Machinery Part in Remanufacturing. In: *The 22nd CIRP conference on Life Cycle Engineering, Procedia CIRP 29*, pp. 758–763.

Zhidchenko, V., Malysheva, I., Handroos, H., and Kovartsev A. (2018). Faster than real-time simulation of mobile crane dynamics using digital twin concept. *Journal of Physics: Conference Series*, 1096. 012071. 10.1088/1742-6596/1096/1/012071.

Zhidchenko, V., Handroos, H., and Kovartsev A. (2019). On-line calculation of fatigue in hydraulically actuated heavy equipment using IoT and Digital Twin concepts. In: *Proceedings of the V International Conference on Information Technology and Nanotechnology (ITNT-2019), Sbornik trudov ITNT-2019*, pp. 382–389. Samara: Novaya technika.

Zhidchenko, V., Handroos, H., and Kovartsev A. (2019). Fatigue life estimation of hydraulically actuated mobile working machines using Internet of Things and Digital Twin concepts. *Submitted for publication in the Journal of Physics: Conference Series.*

Zhong, D., Lv, H., Han, J., Quanrui, W. (2014). A Practical Application Combining Wireless Sensor Networks and Internet of Things: Safety Management System for Tower Crane Groups. *Sensors*, 14, pp. 13794–13814.

ACTA UNIVERSITATIS LAPPEENRANTAENSIS

834. KURONEN, TONI. Moving object analysis and trajectory processing with applications in human-computer interaction and chemical processes. 2018. Diss.

835. UNT, ANNA. Fiber laser and hybrid welding of T-joint in structural steels. 2018. Diss.

836. KHAKUREL, JAYDEN. Enhancing the adoption of quantified self-tracking wearable devices. 2018. Diss.

837. SOININEN, HANNE. Improving the environmental safety of ash from bioenergy production plants. 2018. Diss.

838. GOLMAEI, SEYEDMOHAMMAD. Novel treatment methods for green liquor dregs and enhancing circular economy in kraft pulp mills. 2018. Diss.

839. GERAMI TEHRANI, MOHAMMAD. Mechanical design guidelines of an electric vehicle powertrain. 2019. Diss.

840. MUSIIENKO, DENYS. Ni-Mn-Ga magnetic shape memory alloy for precise high-speed actuation in micro-magneto-mechanical systems. 2019. Diss.

841. BELIAEVA, TATIANA. Complementarity and contextualization of firm-level strategic orientations. 2019. Diss.

842. EFIMOV-SOINI, NIKOLAI. Ideation stage in computer-aided design. 2019. Diss.

843. BUZUKU, SHQIPE. Enhancement of decision-making in complex organizations: A systems engineering approach. 2019. Diss.

844. SHCHERBACHEVA, ANNA. Agent-based modelling for epidemiological applications. 2019. Diss.

845. YLIJOKI, OSSI. Big data - towards data-driven business. 2019. Diss.

846. KOISTINEN, KATARIINA. Actors in sustainability transitions. 2019. Diss.

847. GRADOV, DMITRY. Experimentally validated numerical modelling of reacting multiphase flows in stirred tank reactors. 2019. Diss.

848. ALMPANOPOULOU, ARGYRO. Knowledge ecosystem formation: an institutional and organisational perspective. 2019. Diss.

849. AMELI, ALIREZA. Supercritical CO2 numerical modelling and turbomachinery design. 2019. Diss.

850. RENEV, IVAN. Automation of the conceptual design process in construction industry using ideas generation techniques. 2019. Diss.

851. AVRAMENKO, ANNA. CFD-based optimization for wind turbine locations in a wind park. 2019. Diss.

852. RISSANEN, TOMMI. Perspectives on business model experimentation in internationalizing high-tech companies. 2019. Diss.

853. HASSANZADEH, AIDIN. Advanced techniques for unsupervised classification of remote sensing hyperspectral images. 2019. Diss.

854. POPOVIC, TAMARA. Quantitative indicators of social sustainability applicable in process systems engineering. 2019. Diss.

855. RAMASAMY, DEEPIKA. Selective recovery of rare earth elements from diluted aqueous streams using N- and O –coordination ligand grafted organic-inorganic hybrid composites. 2019. Diss.

856. IFTEKHAR, SIDRA. Synthesis of hybrid bio-nanocomposites and their application for the removal of rare earth elements from synthetic wastewater. 2019. Diss.

857. HUIKURI, MARKO. Modelling and disturbance compensation of a permanent magnet linear motor with a discontinuous track 2019. Diss.

858. AALTO, MIKA. Agent-based modeling as part of biomass supply system research. 2019. Diss.

859. IVANOVA, TATYANA. Atomic layer deposition of catalytic materials for environmental protection. 2019. Diss.

860. SOKOLOV, ALEXANDER. Pulsed corona discharge for wastewater treatment and modification of organic materials. 2019. Diss.

861. DOSHI, BHAIRAVI. Towards a sustainable valorisation of spilled oil by establishing a green chemistry between a surface active moiety of chitosan and oils. 2019. Diss.

862. KHADIJEH, NEKOUEIAN. Modification of carbon-based electrodes using metal nanostructures: Application to voltammetric determination of some pharmaceutical and biological compounds. 2019. Diss.

863. HANSKI, JYRI. Supporting strategic asset management in complex and uncertain decision contexts. 2019. Diss.

864. OTRA-AHO, VILLE. A project management office as a project organization's strategizing tool. 2019. Diss.

865. HILTUNEN, SALLA. Hydrothermal stability of microfibrillated cellulose. 2019. Diss.

866. GURUNG, KHUM. Membrane bioreactor for the removal of emerging contaminants from municipal wastewater and its viability of integrating advanced oxidation processes. 2019. Diss.

867. AWAN, USAMA. Inter-firm relationship leading towards social sustainability in export manufacturing firms. 2019. Diss.

868. SAVCHENKO, DMITRII. Testing microservice applications. 2019. Diss.

869. KARHU, MIIKKA. On weldability of thick section austenitic stainless steel using laser processes. 2019. Diss.

870. KUPARINEN, KATJA. Transforming the chemical pulp industry – From an emitter to a source of negative CO2 emissions. 2019. Diss.

871. HUJALA, ELINA. Quantification of large steam bubble oscillations and chugging using image analysis. 2019. Diss.

ISBN 978-952-335-426-5
ISBN 978-952-335-427-2 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491
Lappeenranta 2019