Julia Malysheva

FASTER THAN REAL-TIME SIMULATION OF FLUID POWER-DRIVEN MECHATRONIC MACHINES

ACTA UNIVERSITATIS LAPPEENRANTAENSIS 960

Julia Malysheva

FASTER THAN REAL-TIME SIMULATION OF FLUID POWER-DRIVEN MECHATRONIC MACHINES

Dissertation for the degree of Doctor of Science (Technology) to be presented with due permission for public examination and criticism at Lappeenranta-Lahti University of Technology LUT, Lappeenranta, Finland on 1st of June, 2021, at noon.

Acta Universitatis
Lappeenrantaensis 960

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

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

Professor Matti Pietola
Department of Mechanical Engineering
Aalto University
Finland

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

ISBN 978-952-335-652-8
ISBN 978-952-335-653-5 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491

Lappeenranta-Lahti University of Technology LUT
LUT University Press 2021

Abstract

Julia Malysheva
Faster than real-time simulation of fluid power-driven mechatronic machines
Lappeenranta 2021
79 pages
Acta Universitatis Lappeenrantaensis 960
Diss. Lappeenranta-Lahti University of Technology LUT
ISBN 978-952-335-652-8
ISBN 978-952-335-653-5 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491

The level of automation of mechatronic machines, such as excavators, logging harvesters or fluid power-driven cranes, has increased significantly over the past few decades. In the machine industry, this led to the emergence and development of novel approaches for the new product development process, such as virtual prototyping. At the design and engineering stages of new mechatronic machine development, virtual prototypes are used for the studying of the design decision effects on machine dynamic behaviour, thus reducing the need for the construction of physical prototypes. Essentially, the virtual prototype of the mechatronic machine is a physics-based simulation model. Mechanical and fluid power components are the most important parts of such simulation models. These components are also inherent to other types of mechatronic systems such as aircraft, heavy industrial process machines, ships, offshore cranes, etc. Depending on the task, the virtual prototype can be run in real time or faster than real time. The required high simulation speed is often a major stumbling block in the employment of more advanced simulation models.

In the work, the problem of a faster than real-time simulation of a mechatronic machine that includes the mechanical and fluid power components is considered. For this task, two different simulation models for a fluid power-driven crane were built and their properties compared. The first simulation model was built using a computationally efficient dynamic topological formulation (Iterative Newton-Euler Formulation) for the multibody modelling of the crane's mechanical structure. The second simulation model was developed using commercial software and taken as a reference for the calculation accuracy and speed analysis. The fluid power components for both simulation models were built using mathematical modelling based on the lumped fluid power theory. The crane, whose dynamics were modelled in the work, is the PATU-655 fluid power-actuated mobile crane. The advantages and disadvantages of both simulation models in achieving faster than real-time simulation were discussed.

Due to the presence of the nonlinearities and singularities inherent in the mathematical model of the fluid power components, during the simulation a very small time step should be used in the integration algorithm in order to maintain numerical stability of the solution. This may result in the simulation time overflows and the inability to maintain the high simulation speed. Machine learning approach can help in solution of such a

problem. In particular, an artificial neural network (ANN) usage for fluid power system modelling can be beneficial. Thus, the work addresses the question of recurrent neural network (RNN) usage for the faster than real-time simulation of fluid power systems. A physics-based simulation model was created using an experimentally verified mathematical model of a hydraulic position servo system (HPS). The RNN of NARX architecture was developed, trained and tested on the training data produced by the physics-based simulation model. A pre-processing technique was developed and applied to the training data in order to speed up the training and simulation processes. The obtained results for the first time show that the employment of the RNN together with the developed pre-processing technique ensures the simulation speed-up of the complex fluid power system at the expense of a small decrease in accuracy.

In the work, another solution for the task of the fast simulation of fluid power systems with singularities originating (in particular) from the presence of small volumes is also proposed. The solution was based on the development and usage of an advanced pseudo-dynamic solver with adaptive criterion (AdvPDS), which is an enhanced version of a classical pseudo-dynamic solver (PDS). The AdvPDS seeks a steady-state solution of pressure building up in a small volume. Two main advantages of the proposed solver were obtained. The first was the higher accuracy and numerical stability of the solution compared with the PDS, owing to the enhanced solver structure and the use of an adaptive convergence criterion. The second was the faster calculation time compared with the conventional integration method, owing to the obtained possibility of larger integration time-step usage. Simulation results confirmed that the AdvPDS is better than conventional solvers for real-time systems that include fluid power components with small volumes. In addition, the work also studies which of the numerical integration methods incorporated into the AdvPDS ensure the efficient (fast and accurate) calculation of stiff fluid power models. Thus, the effect of three fixed-step integration methods (Euler, Runge-Kutta of fourth order, and modified Heun's method) were considered. In the work, the numerical stability of the modified Heun's method was improved by substituting the purely turbulent orifice model with the two-regime orifice model. The two-regime orifice accounts for both the turbulent and laminar flows and thus allows the avoidance of numerical problems related to the small pressure drops. The compiled C language that supports the real-time simulation was chosen as the implementation environment for the developed simulation models. The solutions obtained for the numerical examples using the AdvPDS based on the three integration approaches, their accuracies and calculation speeds were presented in comparison with the solution obtained using a conventional integration procedure. The results showed that, in general, the AdvPDS allows the solution of numerically stiff fluid power models in a very efficient way, ensuring accelerated simulation with high solution accuracy. It was also shown that the simulation speed-up can be obtained not only by the complexity reduction of the numerical integration method inside the AdvPDS, but also by increasing the numerical stability of the employed numerical integration method.

Keywords: faster than real-time simulation, mechatronic machine, machine learning, recurrent neural network, stiff fluid power system modelling and simulation, advanced pseudo-dynamic solver, numerical integration.

Реферат

Юлія Малишева
Симуляція мехатронних машин з гідравлічним приводом у режимі швидше, ніж реальний час
Лаппеенранта 2021
79 сторінок
Acta Universitatis Lappeenrantaensis 960
Diss. Lappeenranta-Lahti University of Technology LUT
ISBN 978-952-335-652-8
ISBN 978-952-335-653-5 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491

Рівень автоматизації мехатронних машин, таких як екскаватор, лісозаготівельні комбайни або гідравлічні крани значно виріс за останнє десятиліття. У машинобудівельній галузі це призвело до появи інноваційного підходу до процесу розробки нових продуктів, а саме віртуального прототипування. На етапах проєктування і розробки нової мехатронної машини віртуальні прототипи використовуються для вивчення впливу проєктних рішень на динамічну поведінку машини, що знижує потреби в створенні фізичних прототипів. Віртуальний прототип мехатронної машини – це фізично обґрунтована симуляційна модель. Механічна і гідравлічна складові є суттєвими частинами симуляційної моделі. Ці складові входять до складу також й інших мехатронних систем, таких як літаки, важкі підіймально-транспортні машини, кораблі, морські крани тощо. Залежно від завдання віртуальний прототип може працювати в режимі реального часу або в режимі швидше, ніж реальний час. Вимога щодо високої швидкості обчислень часто є основною перепоною при використанні більш точних та досконалих симуляційних моделей.

В роботі розглядається проблема симуляції у режимі швидше, ніж реальний час мехатронної машини із механічною і гідравлічною складовими. Для цього були створені дві різні симуляційні моделі гідравлічного крана та проведено порівняння їх властивостей. Перша симуляційна модель була побудована із застосуванням обчислювально ефективного топологічного динамічного формулювання (методу Ньютона-Ейлера) для багатотільного моделювання механічної конструкції крана. Друга симуляційна модель була розроблена з використанням комерційного програмного забезпечення й використовувалась як еталонна для оцінки точності та швидкості обчислень. Гідравлічна складова для обох симуляційних моделей була побудована шляхом математичного моделювання з використанням теорії гідравлічних ланцюгів із зосередженими параметрами. В роботі проведено моделювання динаміки мобільного крану із гідравлічним приводом PATU-655. Також проведено порівняння симуляційних моделей по досягненню симуляції у режимі швидше, ніж реальний час.

Через наявні у математичних моделях гідравлічної складової нелінійності та сингулярності, в алгоритмі інтегрування при моделюванні слід використовувати дуже малий часовий крок для підтримки обчислювальної стійкості рішення. У протилежному випадку переповнення часу унеможливлює зберігання високої швидкості симуляції. У розв'язанні такої проблеми може допомогти підхід із використанням машинного навчання. Зокрема, може бути корисним використання штучних нейронних мереж для моделювання гідравлічної системи. В роботі розглядається питання використання рекурентної нейронної мережі (РНМ) для симуляції гідравлічних систем в режимі швидше, ніж реальний час. Було створено фізично обґрунтовану симуляційну модель на базі експериментально перевіреної математичної моделі гідравлічної сервосистеми (ГСС). РНМ NARX-архітектури була розроблена, навчена і протестована на навчальних даних, створених за допомогою симуляційної моделі. Методика попередньої обробки був розроблений і застосований до навчальних даних, щоб прискорити процеси навчання і симуляції. Отримані результати вперше показали, що використання РНМ спільно із розробленою методикою попередньої обробки даних можуть забезпечити прискорення симуляції складної гідравліко-динамічної системи коштом невеликого зниження точності.

Також в роботі запропоновано альтернативне розв'язання задачі швидкої симуляції гідравлічних систем із сингулярністю, зокрема, що виникає внаслідок присутності малих об'ємів. Розв'язання базується на розробці та використанні вдосконаленого псевдодинамічного інтегратора з адаптивним критерієм (ВПДІ), який є поліпшеною версією класичного псевдодинамічного інтегратора (ПДІ). ВПДІ шукає стаціонарне рішення для тиску, що виникає у малому об'ємі. Отримано дві основні переваги запропонованого інтегратору. По-перше, вищими є точність і стабільність обчислень у порівнянні з ПДІ завдяки вдосконаленій структурі інтегратора й використанню адаптивного критерію збіжності. По-друге, менший час обчислень, у порівнянні з традиційним методом інтегрування, завдяки можливості використання більшого кроку інтегрування. Результати симуляції підтвердили, що ВПДІ є кращим варіантом, ніж традиційні інтегратори для систем реального часу, що включають гідравлічну компоненту із малим об'ємом. Крім того, в роботі також досліджується які з методів чисельного інтегрування, що входять до складу ВПДІ, забезпечують ефективне (швидке і точне) обчислення жорстких моделей гідравлічних систем. Таким чином, було розглянуто вплив трьох методів інтегрування з фіксованим кроком (Ейлера, Рунге-Кутта четвертого порядку та модифікованого методу Хойна). У роботі була поліпшена чисельна стійкість модифікованого методу Хойна шляхом заміни суто турбулентної моделі отвору дворежимною моделлю отвору. Дворежимна модель отвору враховує як турбулентний, так і ламінарний потоки, що дозволяє уникнути числових проблем, пов'язаних з малими перепадами тиску. Компільовану мову С, що підтримує симуляцію в реальному часі, було обрано як середовище реалізації для розроблених симуляційних моделей. Рішення, отримані для чисельних прикладів з використанням ВПДІ на основі трьох підходів інтегрування, їх точності й швидкості обчислень, були представлені у порівнянні із рішеннями, що були отримані із використанням традиційної процедури інтегрування. Результати

показали, що в цілому ВДПІ дозволяє дуже ефективно вирішувати чисельно жорсткі моделі гідравлічних систем, забезпечуючи швидку симуляцію із високою точністю. Також було показано, що прискорення симуляції може бути отримано не тільки шляхом зменшення складності методу чисельного інтегрування всередині ВПДІ, але й шляхом підвищення його чисельної стійкості.

Ключові слова: симуляція в режимі швидше, ніж реальний час, мехатронні машини, машинне навчання, моделювання та симуляція жорстких моделей гідравлічних систем, вдосконалений псевдодинамічний інтегратор, чисельне інтегрування.

Acknowledgements

This work was carried out as a part of Sustainable product processes through simulation (SIM) platform in the Laboratory of Intelligent Machines at the LUT School of Energy Systems at Lappeenranta-Lahti University of Technology LUT, Finland, between 2016 and 2020.

I am thankful to my scientific supervisor Professor Heikki Handroos for guidance, insightful advice, and support in my research work.

I would like to thank the dissertation reviewers Professor Asko Ellman from Tampere University (Finland) and Professor Matti Pietola from Aalto University (Finland) for their time and valuable comments that helped to improve this manuscript.

I would like to express my gratitude to my co-authors Victor Zhidchenko, Stanislav Ustinov and Ming Li for rewarding collaboration, and new insights. Also I would like to thank all the colleagues at the Laboratory of Intelligent Machines for their help and creating fruitful research environment. Doctor Hamid Roozbahani, Professor Huapeng Wu and Juha Koivisto deserve a special mention.

I deeply thankful to my family for their love and support. And a special thank you to my mother Dina Malysheva, who did not let the research fire go out.

The last word of acknowledgment I have saved for my beloved husband Mihail Vinokurov, who have supported me for all these years and our precious son Rasmus Eliel.

Julia Malysheva
May 2021
Lappeenranta, Finland

To my family

Contents

Nomenclature

Latin alphabet

$\mathbf{A}$	rotation matrix	–
A	area	m^2
a_i	Denavit-Hartenberg parameter	m
B_e	effective bulk modulus	Pa
C_d	discharge coefficient	–
C_v	flow constant	–
$C_1 \ldots C_9$	empirical constant	–
d_i	Denavit-Hartenberg parameter	m
dt	integration time step	s
F	internal force	N
F_C	Coulomb friction force	N
F_f	friction force	N
F_{st}	Stribeck friction force	N
$f(\cdot)$	function	–
$\mathbf{G}$	vector of gravity terms	–
$g(\cdot)$	function	–
H	cylinder stroke	m
$\mathbf{I}$	identity matrix	–
$\mathbf{J}$	Jacobian	–
K, k_i	semi-empirical flow coefficients	–
k_v	viscous friction coefficient	Ns/m
L_i	laminar leakage flow coefficient	–
l_i	constant length	m
l_{ij}	experimentally defined leakage constant	–
$\mathbf{M}$	mass matrix	–
m	mass	kg
N	internal torque	Nm
n	number of links	–
n_e	time delay order for network error	–
n_x	time delay order for network input	–
n_y	time delay order for network output	–
p	pressure	Pa
Q	volume flow rate	m^3/s
Q_L	leakage volume flow rate	m^3/s
$\mathbf{R}$	position-vector	–
Re	Reynolds number	–
$\mathbf{r}_P$	position-vector of the point P in global coordinate frame	–
s_i	cylinder length plus its displacement	m
$\mathbf{T}$	homogeneous transformation matrix	–
t	time	s
$\mathbf{V}$	vector of centrifugal and Coriolis terms	–
V	volume	m^3
V_0	dead volume	m^3

v_C	Coulomb velocity	m/s
v_{Ci}	linear velocity of the centre of mass of the link *i*	m/s
v_{st}	Stribeck velocity	m/s
U	voltage	V
$\mathbf{u}$	input vector	–
$\mathbf{u}_P$	position-vector of the point *P* in local coordinate frame	–
$\mathbf{x}$	state vector	–
x_p	cylinder piston displacement	m
$\hat{Z}$	directional unit vector	–

Greek alphabet

α_i	Denavit-Hartenberg parameter	rad
β_i	angle	rad
γ_i	angle	rad
Δ	difference	
ε	parameter describing binary input	–
ζ	valve damping ratio	
$\boldsymbol{\theta}$	vector of joint coordinates	–
θ_i	Denavit-Hartenberg parameter	rad
κ	condition number	–
λ	eigenvalue	–
π	mathematical value $\pi = 3.14159...$	rad
ρ	fluid density	kg/m^3
σ_0	flexibility coefficient	
σ_1	damping coefficient	
$\boldsymbol{\tau}$	vector of torques	–
τ	time constant	s^{-1}
ψ	nonlinear mapping	
ω_i	link *i* rotational velocity	rad/s
ω_n	natural angular frequency	rad/s

Superscripts

^	parameter estimate
current	current value of parameter
next	next value of parameter
prev	previous value of parameter
T	matrix transpose

Subscripts

A	cylinder chamber *A*
B	cylinder chamber *B*
db	dead band
e	effective
f	friction
H	number of layers

limit	low-limit level of parameter
max	maximum value
min	minimum value
P	pump
p	piston
pseudo	artificial parameter
ref	reference
S	supply
s	spool
T	tank
tol high	lower tolerance criteria for parameter
tol low	higher tolerance criteria for parameter

Abbreviations

ANN	artificial neural network
AdvPDS	advanced pseudo-dynamic solver
CAD	computer-aided design
HITL	human-in-the-loop
HPS	hydraulic position servo
INEF	iterative Newton-Euler formulation
LM	Levenberg-Marquardt algorithm
MSE	mean-square error
NARMAX	nonlinear autoregressive moving average with exogenous inputs
NARX	nonlinear autoregressive network with exogenous inputs
NFIR	nonlinear finite impulse response
ODE	ordinary differential equations
PRMS	pseudo-random multilevel signal
RMSE	root-mean-square error
RNN	recurrent neural network
SIMO	single input multiple outputs

1 Introduction

1.1 Background and motivations

Since the levels of complexity and automation of mechatronic machines (excavators, logging harvesters, hydraulically-driven cranes, etc.) have increased significantly over the past few decades, the machine industry has shown great interest in harnessing the benefits of computer simulation. In the machine industry, this has led to the emergence of novel approaches in new product development processes, such as virtual prototyping (Mikkola & Handroos, 1996; Esqué, Raneda, & Ellman, 2003; Liu, Zhang, & Sun, 2019). Nowadays, the approach is also extensively used in product operation and maintenance periods (Boschert & Rosen, 2016). At the design and engineering stages of the mechatronic machine development process, a virtual prototype is used for studying the effects of design decisions on machine dynamic behaviour, thereby reducing the need for a physical prototype construction (Mikkola A. , 1997; Baharudin, Rouvinen, Korkealaakso, & Mikkola, 2014; Esqué, Raneda, & Ellman, 2003). Essentially, the virtual prototype of a mechatronic machine is a simulation model or, in other words, a mathematical representation of all machine elements as well as their interactions. To estimate the performances of the mechatronic machine under development, a simulation of the virtual prototype is used.

Often a major problem of virtual prototypes is their maximum simulation speed, which is particularly related to the complexity and characteristics of the employed mathematical models. A number of recent studies have been dedicated to the problems of real-time (Esqué, Raneda, & Ellman, 2003; Zhidchenko, Malysheva, Handroos, & Kovartsev, 2018; Zheng, Ge, & Liu, 2015; Rahikainen, Kiani, Sopanen, Jalali, & Mikkola, 2018) and faster than real-time simulation (Malysheva I. , Handroos, Zhidchenko, & Kovartsev, 2018) of the virtual prototypes of mechatronic machines. At the same time, the simulation models of mechatronic machines are also extensively exploited for studying human-machine interaction using human-in-the-loop (HITL) simulation. Moreover, HITL simulation can be used for the training of mechatronic machine operators (Baharudin, Rouvinen, Korkealaakso, & Mikkola, 2014). HITL simulation requires the simulation model to be run in real time (Pedersen, Hansen, & Ballebye, 2010). In addition, simulation models are used for real-time automation and control tasks (Zheng, Ge, & Liu, 2015; Pedersen, Hansen, & Ballebye, 2010) and for the failure prediction of the machine parts and systems (Andrade, Feucht, Haufe, & Neukamm, 2016). Moreover, a highly popular control engineering approach based on the employment of reinforcement learning (RL) for the optimal controller design for systems with nonlinear dynamics (Karpenko, Anderson, & Sepehri, 2006) shows a high need for simulation models that are able to run faster than real time. Such simulation models are able to provide large amounts of the training examples for a short period of time that are needed for a RL-agent training.

A typical simulation model of a mechatronic machine includes *a mechanical component* and *a fluid power component*. These components are also the essential parts of the simulation models of other types of mechatronic systems, such as aircraft, heavy industrial process machines, ships, offshore cranes, and so on. The mechanical component includes the mathematical representation of the structural elements (a set of rigid and/or flexible bodies) and their interconnections composing a multibody system. For the derivation of the mathematical model

of a multibody system composed of the rigid bodies the two main approaches are mainly used. The first is based on the concept of virtual work and Lagrange's equation. In the approach the multibody system is considered as a whole. Algorithms based on this approach use the space of generalised coordinates that follow certain minimisation principles and thus produce the trajectories that automatically satisfy the kinematic constraints of the system (Korkealaakso, 2009). The second approach that can be used to formulate the mathematical model of the multibody system is the direct approach, which is based on Newton and Euler equations. According to this approach, the dynamic equations are produced separately for each body while considering the motion explicitly in Cartesian space. The linear and angular momentum conservation principles are applied directly to each body. The constrained reaction forces are considered as external forces. The two approaches described above, as well as their modifications and combinations, are widely used for the mathematical model formulation. Although the approaches use different strategies, they provide equivalent dynamic formulations that can differ in computational efficiency for the specific multibody model (Korkealaakso, 2009).

The multibody system, which is the underlay to the mechanical component, often provides an interface to the external systems, such as the fluid power system (Baharudin, Rouvinen, Korkealaakso, & Mikkola, 2014; Esqué, Raneda, & Ellman, 2003; Zheng, Ge, & Liu, 2015; Pedersen, Hansen, & Ballebye, 2010; Mikkola A., 1997). In this case the fluid power actuator forces are taken in by the multibody system as the generalised forces. At the same time, the positions and velocities are fed back from the multibody system to the fluid power component. In composing the fluid power system model, the centralised pressures approach is usually used. The modelling of the fluid power units such as pumps, actuators and valves are based on the combination of the fluid dynamics and multibody models. In the centralised pressures approach, the components are interconnected by continuity equations (Merritt, 1967).

Mathematically, the mechanical and fluid power components as well as their interactions are expressed as a system of the algebraic and differential equations and referred to as the equations of motion (EOM) or the mathematical model. The system is usually solved using a numerical integration method that ensures the accuracy, stability and efficiency of a numerical solution (Dormand & Prince, 1980; Esqué, 2008).

However, the dynamic processes taking place in the fluid power systems are very complex. The flexibility of hydraulic fluid and the presence of small volumes introduce a numerical stiffness into the mathematically formulated models (Piché & Ellman, 1994). Other phenomena, such as friction in the fluid power units, valve closure, digital control signals and purely turbulent orifices introduce strong nonlinearities, discontinuities and singular states to the model (Piché & Ellman, 1994; Åman, Handroos, & Eskola, 2008). These features also make the hydraulic model numerically stiff and thus difficult to integrate (Piché & Ellman, 1994). In their work (Bowns & Wang, 1990), Bowns and Wang formulated the mathematical stiffness problem that arises during the solution of the fluid power systems in the presence of small volumes, particularly in hydraulic pipes, for the first time. Physically, the mathematical stiffness occurs when the pressure changes rapidly, owing to the low compliance of the fluid in the pipe. According to their observations, this causes the solutions of the system differential equations to decay at widely varying rates. However, it should be noted that the mathematical stiffness is often a local phenomenon, meaning that it may occur occasionally. For example, if

the orifice is located in the fluid power circuit, the stiffness increases approaching infinity if the relationship $\partial \Delta p / \partial Q$ is small, which is true when the volume flow Q tends towards zero. Moreover, according to (Ellman & Piché, 1996; Ellman & Piché, 1999; Åman, Handroos, & Eskola, 2008), if the purely turbulent description of the orifice is used, mathematical stiffness also occurs when the pressure drop Δp is approaching zero.

The numerical stiffness of the mathematical model directly affects the simulation time, which is a vital aspect in the real-time simulation in mechatronic applications. For instance, such a problem is highlighted in (Park, Yoo, Ahn, Kim, & Shin, 2020), where the authors tried to solve the problem of the real-time simulation of an excavator with a numerically stiff fluid power model. Thus, in order to achieve the real-time simulation speed, the model was divided into multiple sub-models to ensure a parallel execution using a local stiff integration solver. The same problem has been recently highlighted in a number of other works dedicated to human-in-the-loop and hardware-in-the-loop systems that included fluid power components. For example, in the work of (Ferreira, Almeida, Quintas, & de Oliveira, 2004), in order to ensure the hardware-in-the-loop real-time simulation for developed controller strategy testing, the authors first simplified the fluid power model and then used a third order explicit solver with a small time step. Thus, the above-described mechatronic applications show the need for the development of a method that can provide a generic practical solution to accelerate the simulation of the mechatronic systems with minor costs in terms of accuracy.

In the solution of the mathematical models, which include ordinary differential equations (ODEs), the family of explicit Runge-Kutta methods that use the integration time step of a fixed size are well established. However, in the research by (Hairer & Wanner, 1996) it was shown that the numerical integrators based on the explicit Runge-Kutta methods are not A-stable (i.e. the numerical stability of the method is not guaranteed for any integration step size), which is apparently why they are not very efficient at stiff problem-solving unless the very small integration time step is used. At the same time, integrators based on implicit methods are A-stable or even L-stable and provide accurate solutions for such problems. Unfortunately, the implicit methods are much more computationally expensive, since they involve solving a nonlinear system of algebraic equations at each time step. This requires the use of the modified Newton iteration scheme, which includes the calculation of an iteration matrix of the form $(\mathbf{I}-\Delta t \beta_0 \mathbf{J})$, where $\mathbf{I}$ is the identity matrix, $\mathbf{J}$ is the Jacobian and $\Delta t \beta_0$ is a scalar, and further its factorisation. The iteration scheme is repeated until a convergence criterion is reached (Esqué, 2008). Due to such iteration scheme usage, the amount of computations can vary from step to step, which can result in simulation time overflows. Thus, the implicit methods cannot be used directly in real-time applications. In contrast to the implicit methods, the previously mentioned explicit methods such as Euler, Runge-Kutta or Predictor-Corrector methods (Hairer & Wanner, 1996) consume much less calculation time in a single time step and thus can ensure a constant simulation time in time-critical real-time applications.

In the vast cases of computer simulations used in product development processes, the simulations are free of solution time restrictions. This means that the simulation of a few seconds is allowed to take several hours in real time. Consequently, all the control signals intended for the simulated model should be predefined (Korkealaakso, 2009). However, in the cases of the HITL simulators, where the operator produces a control signal during the simulation, the optimal controller design based on the employment of reinforcement learning

(RL), the digital twins for machine life-cycle assessment, and the decision support systems that aid the machine operator in challenging environments, the simulation should be run in synchrony with real time or much faster than real time. Thus, the real-time simulation and faster than real-time simulation can be considered special cases of conventional computer simulation. In these cases all the calculations related to the advancing in time of the simulation model should be completed within the predetermined time range. The time range is usually dictated by a time synchronous connection to the real world or by a favoured simulation speed. Figure 1.1 shows the conceptual difference between the real-time and faster than real-time simulations. Here t_n is the real-time instant, when the monitoring of the simulation model states is performed, T_{RT} and T_{FTRT} are the times needed for a single run of the simulation model.

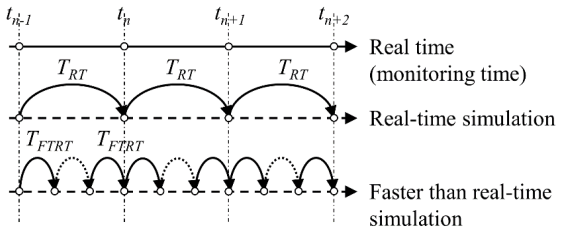

Figure 1.1: Real-time and faster than real-time simulations.

Accelerated simulation of the mechanical component can be obtained using computational efficient multibody representation (Malysheva, Handroos, Zhidchenko, & Kovartsev, 2018; Zhidchenko, Malysheva, Handroos, & Kovartsev, 2018). In the works the problem of faster than real-time simulation of mechatronic machine which included the mechanical and fluid power components was considered. For this task, two different simulation models for a hydraulically-driven crane were built and their properties compared. The first simulation model was built using a computationally efficient dynamic topological formulation (Iterative Newton-Euler Formulation) for the multibody modelling of the crane's mechanical structure. The second simulation model was developed using commercial software and taken as a reference for the calculation accuracy and speed analysis. The advantages and disadvantages of both simulation models in achieving the faster than real-time simulation were discussed. Julia Malysheva was the first author of the paper (Malysheva, Handroos, Zhidchenko, & Kovartsev, 2018) and co-author in both of the other papers. In these papers the author was responsible for the development of the reference models of the hydraulic mobile crane using commercial software and their translation to the compiled programming language, as well as for the development of the mathematical model of the crane fluid power system. She was also responsible for performing experiments with the reference model, gathering and processing the simulation results and writhing the respective parts of the papers.

However, the faster than real-time simulation of the fluid power component is an even more challenging task. In the fluid power system research area, in order to improve the computational efficiency of the solution of the numerically stiff fluid power model, different approaches have been proposed. In particular, the accelerated simulation can be obtained using a *semi-empirical modelling* approach for particular fluid power units in the simulation model. For example, the use of the two-regime flow orifices (Ferreira, Almeida, Quintas, & de Oliveira, 2004; Ellman & Piché, 1996) instead of the purely turbulent orifice model allows singularities, which can appear when the pressure drop across the orifice is close to zero, to be avoided. On the other

hand, the integration of the fluid power model can be performed with the help of *special solvers* (Piché & Ellman, 1994). To overcome the stiffness of differential equations in fluid power systems with the small volumes, a classical pseudo-dynamic solver (PDS) was proposed by Åman and Handroos (Åman & Handroos, 2008; Åman & Handroos, 2009; Åman & Handroos, 2010). This solver can be related to the class of explicit solvers. The PDS algorithm ensured the accuracy increase and reduction of the computational time needed for the simulation of the stiff fluid power circuits. The PDS was based on the assumption that if the considered volume is small enough, the pressure building up in the volume can be substituted by a steady-state pressure value. This was achieved by implementing an iterative technique with the substitution of the small volume with a volume that is large enough to obtain a numerically stable pressure solution. During the stiff fluid power model integration, the PDS could use the larger integration time steps than the conventional integrators without being trapped in the numerical instability area, which affects the computational time of the simulation. However, in their work, only a short-term simulation (about two seconds) with predefined inputs was considered, which did not give a full picture of the solver characteristics. The research was continued by Malysheva, Ustinov and Handroos by proposing in (Malysheva, Ustinov, & Handroos, 2020; Malysheva & Handroos, 2020) a new and enhanced version of the classical pseudo-dynamic solver, referred to as an advanced pseudo-dynamic solver with adaptive criterion (AdvPDS). The new solver had two main advantages. The first was the higher accuracy and numerical stability of the solution compared with the PDS, owing to the enhanced solver structure and the use of an adaptive convergence criterion. The second was the faster calculation time compared with the conventional integration method, owing to the obtained possibility of larger integration time step usage. Julia Malysheva was the principal author and investigator of the paper.

Another method for solving pressures in small volumes has recently been introduced, in (Kiani Oshtorjani, Mikkola, & Jalali, 2019). The proposed method was based on singular perturbation theory. The modified version of this theory was used for the algorithm. The main principle of the algorithm was the replacement of a stiff differential equation of pressure by the algebraic equation in accordance with singular perturbation theory. The replacement of the differential equation allows a numerically stable response of the pressure to be achieved at different integrator time steps. Consequently, the time step of the integration can be increased without significant losses in calculation accuracy, which allows the method to be implemented in real-time simulations. However, the method can only be applied under the condition that the system boundary layer is exponentially stable (Rahikainen, Kiani, Sopanen, Jalali, & Mikkola, 2018; Kiani Oshtorjani, Mikkola, & Jalali, 2019).

A different approach was presented by Krus in (Krus, 2011) who applied distributed modelling using transmission line elements (or bi-lateral delay lines) for modelling and simulation of large hydromechanical systems. Usage of the transmission line elements for the component connection in the complex fluid power system allowed to isolate the components numerically from each other. Then a local implicit solver can be applied to each component separately. This allowed to use larger time steps for system simulation ensuring faster simulation speeds. Moreover, since all the calculations of fluid power component are done within its model the parallel computation of the component is possible. The proposed modelling method was successfully adopted in HOPSAN software developed in Linköping University.

In contrast to the plain mathematical modelling of the simulated system, the modelling with an *artificial neural network* (ANN) can offer a way to achieve high simulation speeds while preserving accurate physical modelling. The approach is supported by the fact that, in general, with the correct architecture and proper training dataset, the universal approximating capabilities of neural networks guarantee that any continuous function can be modelled to any desired precision (Hornik, 1991). The recurrent neural network (and its variants) is a network architecture that has proved itself to be successful in the tasks of time-series prediction and dynamic systems identification and control (Ogunmolu, Gu, Jiang, & Gans, 2017; Bianchi, Maiorino, Kampffmeyer, Rizzi, & Jenssen, 2017). In contrast to common ANNs (such as multilayer feedforward networks), where the current output depends only on the input, in recurrent neural networks (RNNs) the current output can depend on the current input as well as on the history of previous inputs, outputs, errors and/or network states. This architectural feature can be considered a local memory and it enables RNNs to account for temporal information (Sinha, Gupta, & Rao, 2000; Petlenkov, 2007). Several recently published research papers have studied the modelling of complex dynamic systems with RNNs (Petlenkov, 2007), including hydraulic systems (Patel & Dunne, 2003). These studies have shown quite promising results. In the works, the variation of a recurrent neural network, namely a nonlinear autoregressive network with exogenous inputs (NARX), was employed. The reason for the architecture choice was based on results obtained in (Siegelmann, Horne, & Giles, 1997), where it was shown that NARX networks outperform conventional RNNs regarding problems with long-term dependencies and are computationally as strong as Turing machines. However, the modifications of NARX architecture, such as a nonlinear finite impulse response (NFIR) and nonlinear autoregressive moving average with exogenous inputs (NARMAX) architectures can also be used for dynamic system modelling (Łacny, 2012; Schram, Verhaegen, & Krijgsman, 1996). In (Malysheva, Li, & Handroos, 2020; Malysheva, Ustinov, & Handroos, 2020; Malysheva & Handroos, 2020), a physics-based simulation model was created using an experimentally verified mathematical model of a hydraulic position servo system (HPS). The RNN of NARX architecture was developed, trained and tested on the training data produced by the physics-based simulation model. A pre-processing technique was developed and applied to the training data in order to speed up the training and simulation processes. The obtained results show for the first time that the employment of the RNN together with the developed pre-processing technique ensures the simulation speed-up of the complex fluid power system at the expense of a small decrease in accuracy. Julia Malysheva was the principal author and investigator in the paper.

Another important aspect that should be considered is the implementation of the developed simulation model. Specifically, the choice of a programmable language for the implementation can significantly affect the simulation speed. According to the research (Pastorino, Cosco, Naets, Desmet, & Cuadrado, 2016), the interpreted languages such as MATLAB and Python NumPy are well developed and easy to use for software development, debugging and testing and are thus very popular among mechanical engineers. However, they are troublesome for real-time simulations due to their low computational efficiency. On the other hand, the compiled languages, such as C, C++ and Fortran can ensure the real-time simulation of the simulation model. Moreover, if the real-time simulation is required to be performed on the target machine (for example, onboard), the software written in the compiled language can be used without extensive modifications (Pastorino, Cosco, Naets, Desmet, & Cuadrado, 2016).

1.2 Scope of the work

1.2.1 Research questions

1. To study the state-of-the-art methods and approaches allowing for the acceleration of the simulation model computation of the complex mechatronic machines, which include the mechanical and fluid power components at real-time and faster than real-time simulation speeds.
2. To examine the problem of the computationally efficient mathematical modelling of the multibody system with fluid power actuation in comparison with the modelling using commercial software in terms of the model accuracy and simulation speed.
3. To investigate the ability of the recurrent neural network together with the developed training data pre-processing technique to model a complex fluid power system and provide accelerated and accurate simulation.
4. To investigate the effectiveness of the reduction of the numerical stiffness (originating from the presence of the small volumes in fluid power circuit) using developed advanced pseudo-dynamic solver in achieving accelerated simulation of the fluid power circuit.

1.2.2 Research methods

In this section an overview of the research methods that were used in the work in order to answer the research question is presented. A literature review was carried out to evaluate the state of knowledge and find out the suitable state-of-the-art methods and approaches allowing for the acceleration computation of the simulation model of the complex mechatronic machines at real-time and faster than real-time simulation speeds.

To answer the question of how efficient is mathematical modelling of the multibody system with fluid power actuation in comparison with the modelling using commercial software a case study of a hydraulic mobile crane was implemented. Within this framework mathematical modelling of the crane and construction of the crane dynamic model in commercial software were performed. The simulation of the models provided the data for analysis.

For verification of the proposed modelling approaches concerning accelerated computation of the fluid power simulation models the experiments with five fluid power circuits of different complexity were carried out. Taking into account the inherent numerical stiffness of the mathematical representation of the fluid power systems, the study investigated the following modelling approaches. The first approach employs machine learning and the recurrent neural networks as a tool for the complex fluid power system accelerated and accurate simulation. In this framework the best trained network was selected using statistical analysis. The second approach investigates the effectiveness of the reduction of the numerical stiffness (originating from the presence of the small volumes in fluid power circuit) using special solvers. Within this approach the performances of the classical pseudo-dynamic solver was studied and further improved in the novel advanced pseudo-dynamic solver with adaptive criteria using simulation.

1.3 Scientific contribution of thesis

The main contribution of the work lies in the research of methods that allow the realisation of faster than real-time simulation of mechatronic machines.

1. The development of the detailed simulation model of an example hydraulic mobile crane composed of multibody mechanical components, and the mathematical model of the fluid power system using commercial software. Following translation to the compiled programming language, the model showed the ability to calculate faster than real-time at acceptable accuracy levels.
2. The work successfully demonstrated that RNNs of the NARX architecture employment can ensure the faster simulation of a complex fluid power system in contrast to conventional mathematical modelling. RNNs of the NARX were developed, trained and tested on the training data produced by the mathematical-based simulation model of the fluid power system. A pre-processing technique was developed and applied to the training data in order to speed up the training and simulation processes. The obtained results show for the first time that the employment of the RNN together with the developed pre-processing technique ensures the simulation speed-up of the complex fluid power system at the expense of a small decrease in accuracy.
3. The advanced pseudo-dynamic solver with adaptive criterion has been proposed for the efficient solution of fluid power systems with singularities originating (in particular) from the presence in the system of small volumes. There are two main advantages of the proposed solver. The first is the higher accuracy and numerical stability of the solution compared with the classical pseudo-dynamic solver, owing to the enhanced solver structure and the use of an adaptive convergence criterion. The second is the faster calculation times compared with conventional integration methods such as the fourth order Runge-Kutta method, owing to the achieved possibility of larger integration time step usage. Simulation results confirm that the advanced pseudo-dynamic solver is more efficient than conventional solvers for the solution of the real-time systems that include fluid power components with small volumes. The described advantages allow its use in simulations of mobile machines in real-time and faster than real-time applications.
4. The effect of the three numerical integration methods (Euler, Runge-Kutta of fourth order, and modified Heun's method with improved stability) used inside the AdvPSD on the solution efficiency of the stiff mathematical model was studied. The stability of the modified Heun's method was improved by the use of the two-regime orifice model. Analysis of the obtained simulation results showed that, in general, harnessing the power of the AdvPDS allows the solution of numerically stiff hydraulic models in a very efficient way, ensuring accelerated simulation with high solution accuracy. It was also shown that the simulation speed-up can be obtained not only by the complexity reduction of the numerical integration method inside the AdvPDS, but also by increasing the numerical stability of the employed numerical integration method.

1.4 Thesis outline

The present doctoral dissertation is based on the five research publications and consists of two parts. The first part provides an overview of the methods and approaches that can ensure

accelerated simulation of the mechatronic machines and the real-time or faster than real-time simulation. Also, the first part outlines the most importing findings presented in the five articles the dissertation is based on. The second part introduces the detailed results that have been or will be published in the previously listed five scientific articles.

Chapter 1 covers the background, motivation and scope of the study and presents the scientific contribution of this thesis.

In Chapter 2, the mathematical modelling of a multibody system with fluid power actuation using computational effective INEF formulation in comparison with commercial modelling of the same system is presented in comparison with their accuracy and simulation time. The example multibody system and fluid power circuit modelling are also presented in the chapter.

In Chapter 3, the approach to the fluid power system modelling with the recurrent neural network of NARX architecture for faster than real-time simulations is introduced, and the simulation results obtained are discussed.

Chapter 4 presents the computationally efficient practical method for solving the dynamics of fluid power circuits in the presence of singularities using the developed advanced pseudo-dynamic solver with adaptive criterion. The advantages of using the modified Hein method of improved numerical stability for pressure integration inside AdvPSD are also presented in this chapter.

In Chapter 5 the conclusions are presented.

2 Fast simulation of a mobile working machine

In this chapter, an example of a multibody system with a fluid power drive is considered. The selected system is the PATU-655 mobile hydraulic crane, which consists of mechanical and fluid power components. Structurally, the crane consists of a pillar, a lifting boom, a system of four interconnected side links, an outer boom and an extension boom. The crane has five hydraulic cylinders. Two of them actuate the slew mechanism, providing the rotation of the crane around the vertical axis. Two other cylinders raise the lifting boom and the outer boom respectively and the fifth cylinder provides the sliding motion of the extension boom and controls its length. The maximum admissible crane load for the case of the maximum boom extension is 500 kg. In this chapter, the full crane model is developed using commercial software as well as a computational efficient mathematical model, which considers only the planar motion of the crane (the dynamics of the slew mechanism is not taken into account). The two model calculation speeds and their accuracies are presented in comparison (Malysheva I. , Handroos, Zhidchenko, & Kovartsev, 2018; Zhidchenko, Malysheva, Handroos, & Kovartsev, 2018).

2.1 Mathematical modelling of the multibody system of a crane

2.1.1 Crane kinematics

Most real-time methods for presenting the dynamics of multibody systems consisting of rigid bodies use relative coordinates, taking advantage of the mechanism topology (topological formulation). Also, the considered mobile crane kinematics can be presented using a topological formulation. In this case the crane is considered as the chain of the links (crane booms) connected through the revolute and prismatic joints (Jalon & Bayo, 1994).

In order to represent the crane kinematics, the global frame OXYZ is set up at the base of the crane. The local coordinate frames with the origins located at the joints are assigned to each link using the Denavit–Hartenberg convention. According to the convention, each local frame $O_i x_i y_i z_i$ has the origin at the point representing the joint between the two adjacent links. The z_i-axis is aligned in the direction of the joint i motion (rotational or translational), the x_i-axis is parallel to the common normal $x_i = \pm(z_i \times z_{i-1})$ and the y_i-axis is chosen in order to complete the right-handed coordinate system. The crane has four independent joint coordinates: the angle of the pillar rotation, the angles of rotation of the lifting boom and extension boom, and the length of the extension boom. The joint numbering, orientation of assigned local frames and joint coordinates are shown in Figure 2.1.

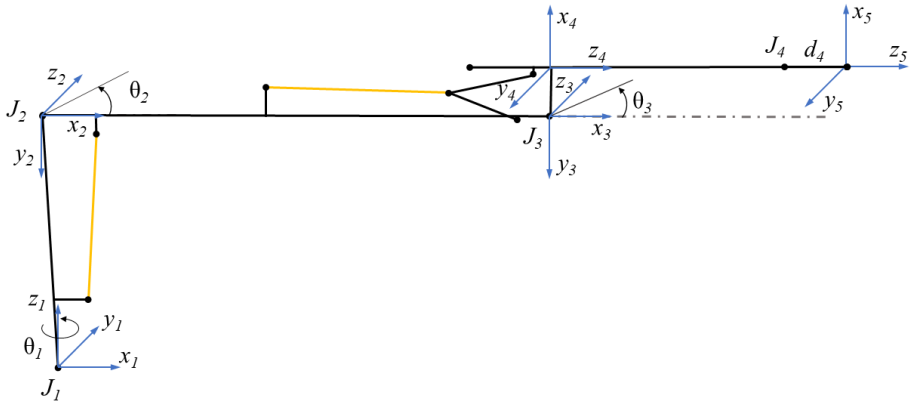

Figure 2.1: Crane kinematics: Joint numbering, orientation of assigned local frames and joint coordinates.

The Denavit–Hartenberg parameters of the crane can be written as shown in Table 2.1.

Table 2.1: Denavit–Hartenberg parameters.

Joint *i*	$\boldsymbol{\theta_i}$	$\boldsymbol{d_i}$	$\boldsymbol{a_i}$	$\boldsymbol{\alpha_i}$
1	θ_1	d_1	a_1	$-\pi/2$
2	θ_2	0	$-a_2$	0
3	$\theta_3 + \pi/2$	0	$-a_3$	$-\pi/2$
4	0	$d_4 + L$	0	0

In Table 2.1, θ_i is the joint angle measured as a rotation angle from x_{i-1} to x_i about z_{i-1}, d_i is the joint distance measured from the origin O_{i-1} to the intersection of the z_{i-1} and x_i along the z_{i-1}, a_i is the link length measured from the intersection of the z_{i-1} and x_i to the origin O_i,
α_i is the link twist angle measured as a rotation angle from z_{i-1} to z_i about x_i. Thus, the joint coordinates that build up the generalised coordinates of the system are: θ_1, θ_2, θ_3, and d_4.

In general, the configuration of a rigid body in a three-dimensional space, meaning its position and orientation relative to some reference frame, can be described by a 4×4 homogeneous transformation matrix:

$$\mathbf{T}_A^B = \begin{bmatrix} \mathbf{A}_A^B & \mathbf{R}_A^B \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.1}$$

where $\mathbf{A}_A^B$ defines the 3×3 rotation matrix with $\det(\mathbf{A}_A^B) = 1$ of frame *B* with respect to frame *A*, and $\mathbf{R}_A^B$ defines the 3×1 position-vector of the origin of the frame *B* rigidly attached to the body with respect to reference frame *A*.

The position of any point *P* of the body can be represented in the reference coordinate frame as:

$$\mathbf{r}_P = \mathbf{R}_A^B + \mathbf{A}_A^B \mathbf{u}_P \tag{2.2}$$

where $\mathbf{u}_P$ is the 3×1 position-vector of the point *P* in the local coordinate frame. In the homogeneous representation, the position of point *P* can be represented as follows:

$$\begin{bmatrix} \mathbf{r}_P \\ 1 \end{bmatrix} = \mathbf{T}_A^B \begin{bmatrix} \mathbf{u}_P \\ 1 \end{bmatrix} \tag{2.3}$$

Using the parameters presented in Table 2.1, the position and orientation of the *i*-th local coordinate frame with respect to (*i-1)*-th coordinate frame can be derived using the general form of the homogeneous transformation matrix for the adjacent coordinate frames:

$$\mathbf{T}_{i-1}^{i} = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\sin(\theta_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

$$\mathbf{T}_0^1 = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & a_1\cos(\theta_1) \\ \sin(\theta_1) & 0 & \cos(\theta_1) & a_1\sin(\theta_1) \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

$$\mathbf{T}_1^2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_2\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

$$\mathbf{T}_2^3 = \begin{bmatrix} -\sin(\theta_3) & 0 & -\cos(\theta_3) & -a_3\sin(\theta_3) \\ \cos(\theta_3) & 0 & -\sin(\theta_3) & a_3\cos(\theta_3) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

$$\mathbf{T}_3^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_4 + L \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

Then, the link transformations can be multiplied together to find the full transformation that relates coordinate frame 0 to frame 4 as:

$$\mathbf{T}_0^4 = \mathbf{T}_0^1\mathbf{T}_1^2\mathbf{T}_2^3\mathbf{T}_3^4 \tag{2.9}$$

In order to connect the hydraulic model to the kinematic model, it is necessary to determine the relationship between cylinder movements and joint coordinate change (Figure 2.2). This can be done using a trigonometric approach. For the joint variable θ_2 this relationship can be written as:

$$\theta_2 = \gamma_1 + \gamma_2 + \gamma_3 + \operatorname{acos}\left(\frac{l_1^2 + l_2^2 - s_1^2}{2l_1l_2}\right) - \frac{\pi}{2} \tag{2.10}$$

where $\gamma_1,\ \gamma_2,\ \gamma_3$ are the constant angles defined by the crane structure; l_1, l_2 are the constant lengths defined from the crane structure, and s_1 is the cylinder length plus its displacement.

The joint variable θ_3 is related to the cylinder displacement s_2 as:

$$\theta_3 = \gamma_5 + \beta_3 + \gamma_4 - \frac{\pi}{2} \tag{2.11}$$

$$\beta_3 = \beta_5 + \beta_6 \tag{2.12}$$

$$\beta_4 = \beta_2 + \gamma_5 \tag{2.13}$$

$$\beta_5 = \operatorname{acos}\left(\frac{l_6^2 + l_7^2 - s_2^2}{2 l_6 l_7}\right) \tag{2.14}$$

$$\beta_6 = \operatorname{acos}\left(\frac{l_5^2 + l_7^2 - l_4^2}{2 l_5 l_7}\right) \tag{2.15}$$

$$l_7 = \sqrt{s_2^2 + l_6^2 - 2 s_2 l_6 \cos\beta_4} \tag{2.16}$$

Here β_i represents the variable rotation angles, whereas γ_i and l_i represent constant angles and lengths that can be defined from the crane structure.

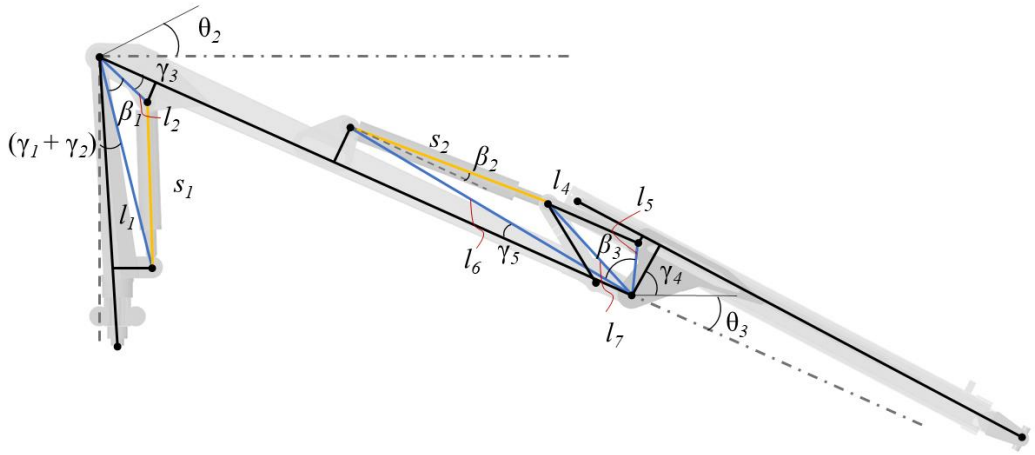

Figure 2.2: Joint coordinate change due to cylinder movements.

2.1.2 Crane dynamics modelling

A widely used formulation to express the multibody dynamic model is based on Lagrangian dynamics (Khalil & Dombre, 2002; Craig, 2005). The computational complexity of such a formulation is $O(n^4)$. In order to provide the fast simulation of the dynamic model, a dynamic

formulation with low computational complexity should be employed. One such formulation that is widely used in the robotics community is the Iterative Newton-Euler formulation (INEF) (Luh, Walker, & Paul, 1980; Craig, 2005) of *O(n)* computational complexity. This formulation is employed in the work for crane dynamics modelling.

The INEF consists of outward and inward iterations. Outward iterations are applied link-by-link starting from link 1 and moving to link *n*. These iterations are intended to calculate for each link *i* the rotational velocity ω_i and acceleration $\dot{\omega}_i$, as well as the linear acceleration of the centre of mass $\dot{v}_{Ci}$. Then, using Newton-Euler equations of motion, the inertial forces F_i and torques N_i acting at the centre of mass of each link *i* are calculated. During inward iterations, for each link the reaction forces f_i and torques n_i acting at the joints are calculated from the force-balance and moment-balance respectively. The link-by-link calculation starts from link *n* and moves backwards to link 1. The algorithm for INEF can be presented as Algorithm 2.1. Each link in the multibody system can be presented as shown in Figure 2.3.

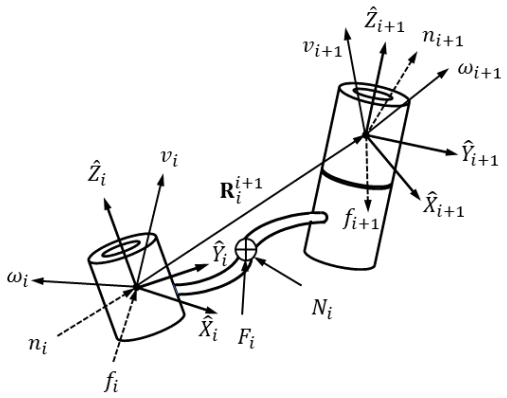

Figure 2.3: Forces acting in link *i*

Algorithm 2. 1: INEF

Input: $n, \theta_{i+1}, \dot{\theta}_{i+1}, \ddot{\theta}_{i+1}, A_i^{i+1}, A_{i+1}^{i}, \hat{Z}_{i+1}, \mathbf{R}_i^{i+1}, \mathbf{R}_{i+1}^{Ci+1}, m_{i+1}, \dot{d}_{i+1}, \ddot{d}_{i+1}, I_{Ci+1}$
Output: τ_i

Initialisation :

$i = 0, \omega_0 = 0, \dot{\omega}_0 = 0, \dot{v}_0 = g\hat{Z}_0, f_n = 0, n_n = 0$

LOOP 1 (Outward iterations):

for $i = 0$ to $n - 1$ **do**
  $\omega_{i+1} = A_i^{i+1}\omega_i + \dot{\theta}_{i+1}\hat{Z}_{i+1}$
  if (joint $i + 1$ is prismatic) **then**
    $\dot{\omega}_{i+1} = A_i^{i+1}\dot{\omega}_i$
  else $\dot{\omega}_{i+1} = A_i^{i+1}\dot{\omega}_i + A_i^{i+1}\omega_i \times \dot{\theta}_{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\hat{Z}_{i+1}$
  end if
  if (joint $i + 1$ is prismatic) **then**
    $\dot{v}_{i+1} = A_i^{i+1}\left(\omega_i \times \mathbf{R}_i^{i+1} + \omega_i \times \left(\omega_i \times \mathbf{R}_i^{i+1}\right) + \dot{v}_i\right) + 2\omega_{i+1} \times \dot{d}_{i+1} + \ddot{d}_{i+1}\hat{Z}_{i+1}$
  else $\dot{v}_{i+1} = A_i^{i+1}\left(\omega_i \times \mathbf{R}_i^{i+1} + \omega_i \times \left(\omega_i \times \mathbf{R}_i^{i+1}\right) + \dot{v}_i\right)$
  end if
  $\dot{v}_{Ci+1} = \dot{\omega}_{i+1} \times \mathbf{R}_{i+1}^{Ci+1} + \omega_{i+1} \times \left(\omega_{i+1} \times \mathbf{R}_{i+1}^{Ci+1}\right) + \dot{v}_{i+1}$

13: $F_{i+1} = m_{i+1}\dot{v}_{Ci+1}$
14: $N_{i+1} = I_{Ci+1}\dot{\omega}_{i+1} + \omega_{i+1} \times I_{Ci+1}\omega_{i+1}$
15: **save** F_{i+1}, N_{i+1}
16: **end for**
LOOP 2 (Inward iterations):
17: **for** $i = n - 1$ to 1 **do**
18: $f_i = A_{i+1}^i f_{i+1} + F_i$
19: $n_i = N_i + A_{i+1}^i n_{i+1} + \mathbf{R}_i^{Ci} \times F_i + \mathbf{R}_i^{i+1} \times A_{i+1}^i f_{i+1}$
20: $\tau_i = n_i^T \hat{Z}_i$
21: **save** τ_i
22: **end for**

The INEF can be applied straightforwardly to the considered system or can be used analytically for obtaining closed-form dynamic equations as:

$$\boldsymbol{\tau} = \mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{V}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right) + \mathbf{G}(\boldsymbol{\theta}) \tag{2.17}$$

where $\mathbf{M}(\boldsymbol{\theta})$ is the mass matrix of the system, $\mathbf{V}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right)$ is a vector of centrifugal and Coriolis terms, $\mathbf{G}(\boldsymbol{\theta})$ is a vector of gravity terms. Originally, INEF is intended for inverse dynamics problem solution. Thus, in (2.17) it is assumed that the position, velocity, and acceleration of each joint are known and the forces and torques needed in the joints are calculated. In order to find the solution for the joint coordinates $\boldsymbol{\theta}$, the torques $\boldsymbol{\tau}$ acting in the joints should be equated to the torques created by hydraulic cylinders, which are created from the hydraulic cylinder forces. In the work, only the angular accelerations of joint 2 and 3 are considered. The described above equations form the ODE system the solution of which gives the angles of rotation of the booms. The obtained angles of the booms are then can be used for the position calculation of other crane points with the help of the mentioned above kinematic expressions.

2.2 Fluid power system modelling

Fluid power system modelling can be approached from the point of view of lumped fluid theory (Merritt, 1967). According to this theory, any fluid power system can be considered as a number of separate volumes with evenly distributed pressures. The volumes are separated by throttles and orifices that create pressure drops in the fluid when it passes through them. In turn, the pressure drop together with orifice geometrical parameters are used for the volume flow calculation. Finally, pressure built up in each volume can be calculated using a continuity equation that relates to the effective bulk modulus with respect to the considered volume and the difference between inlet Q_{in} and outlet Q_{out} volume flows (Merritt, 1967).

In the following subsections, the modelling based on the lumped fluid power theory of the five fluid power circuits of different complexities is presented. All the circuits include the elements that negatively affect their simulation speed. These elements can be described mathematically as nonlinearities, discontinuities and singular states. These elements make the mathematical models of the circuits numerically stiff and thus difficult to integrate. In particular, the numerical stiffness of the model causes the solutions of the system of the differential equations to decay at widely varying rates.

2.2.1 Circuit 1: Simple fluid power system

One of the simplest fluid power systems can be constructed using two sequentially connected orifices, as shown in Figure 2.4. In the circuit, two compressible volumes V_1 and V_2 are considered.

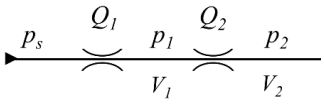

Figure 2.4: Simple fluid power system (Circuit 1).

The pressure developing in the volume V_1 can be calculated from the following equation:

$$\dot{p}_1 = \frac{B_e}{V_1}(Q_1 - Q_2) \tag{2.18}$$

where B_e is the effective bulk modulus in this part of the circuit, Q_1 and Q_2 are the orifice inlet and outlet volume flows, respectively. If we consider the sharp-edged orifice, the dependency of volume flow on the pressure drop can be approximated using the turbulent orifice model (Merritt, 1967) as:

$$Q = C_d A \sqrt{\frac{2\Delta p}{\rho}} \tag{2.19}$$

In (2.19) Δp is the pressure drop across the orifice, A is the orifice cross-section area, C_d is the discharge coefficient and ρ is the fluid density. Thus, for the simple circuit the volume flows will be dependent on the respective pressure drops as:

$$Q_1 = k_1\sqrt{p_s - p_1} \tag{2.20}$$

$$Q_2 = k_2\sqrt{p_1 - p_2} \tag{2.21}$$

where k_1 and k_2 are the semi-empirical volume flow coefficients, which can be determined for each orifice as $k_i = C_{di}A_i\sqrt{\frac{2}{\rho}}$, ($i$ = 1, 2).

If volume V_1 is set such that $V_1 << V_2$, it will introduce higher order dynamics to the system making its mathematical model numerically stiff.

2.2.2 Circuit 2: Fluid power system with two-way flow control valve

The second system under investigation includes a two-way flow control valve. The valve is often used in mobile working machines. The fluid power circuit related to the system is schematically depicted in Figure 2.5. The circuit consists of a pressure power source, two-way

flow control valve, orifice and 2/2 directional control valve. A two-way flow control valve consists of two components: pressure compensator and control throttle. Regardless of the load pressure, the valve ensures a constant volumetric flow by varying the size of the control throttle and thus regulating the speed of an actuator. The volume between the pressure compensator and control throttle is assumed to be a small volume, the presence of which increases the stiffness of the system. The power source is assumed to be an ideal pressure source with constant pressure. It is composed of a hydraulic accumulator, pump, pressure relief valve and tank. To reach the tank, hydraulic fluid flow passes through a two-way control valve and two orifices after the valve. One of the orifices is an ordinary sharp-edged orifice, whereas the other is a 2/2 directional control valve, the opening of which can be controlled by signal U_d.

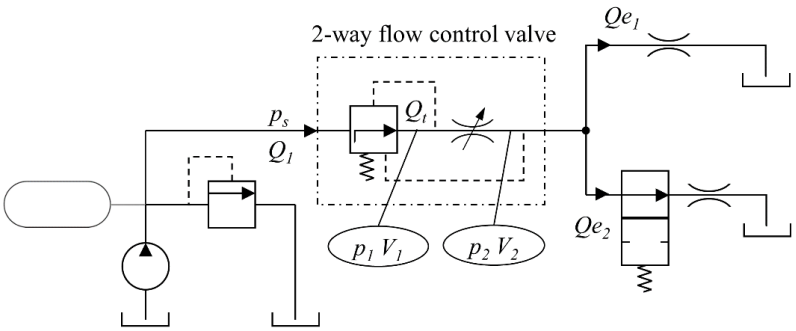

Figure 2.5: Fluid power system with two-way flow control valve (Circuit 2).

Pressure in the system can be integrated from the following continuity equations:

$$\dot{p}_1 = \frac{B_e}{V_1}(Q_1 - Q_t) \tag{2.22}$$

$$\dot{p}_2 = \frac{B_e}{V_2}(Q_t - Q_{e1} - Q_{e2}) \tag{2.23}$$

where B_e is the oil effective bulk modulus, V_1 and V_2 are pipeline volumes, where V_1 is a small volume, Q_1 and Q_t are volumetric flows through the pressure compensator and control throttles, and Q_{e1} and Q_{e2} are orifice and directional control valve volume flows, respectively. Volume flows Q_1 and Q_t can be obtained as follows:

$$Q_1 = K\sqrt{|p_s - p_1|}\text{sign}(p_s - p_1) \tag{2.24}$$

$$Q_t = k_t\sqrt{|p_1 - p_2|}\text{sign}(p_1 - p_2) \tag{2.25}$$

where p_s is the supply pressure, and K and k_t denote the semi-empirical flow coefficients for the pressure compensator throttle and for the control throttle respectively. Both coefficients can be integrated from the following differential equations:

$$\dot{K} = \frac{C_5 - p_1 + p_2 - \big(C_1 + C_2(p_s - p_1)\big)K}{C_3} \tag{2.26}$$

$$\ddot{k}_t = (U_e - C_9)C_6C_7^2 - 2\dot{k}_tC_8C_7 - k_tC_7^2 \tag{2.27}$$

where U_e is the signal applied to control throttle (opening), and C_1, C_2, C_3, C_5, C_6, C_7, C_8, C_9 are empirical constants (Handroos & Vilenius, 1991). Volume flows Q_{e1} and Q_{e2} are obtained according to the following flow equations:

$$Q_{e1} = k_1\sqrt{|p_2 - p_t|} \tag{2.28}$$

$$Q_{e2} = k_2\sqrt{|p_2 - p_t|} \tag{2.29}$$

where k_1 and k_2 are semi-empirical flow coefficients for the orifice and directional control valve, and p_t is the tank pressure. The initial values and constants of the system used in the equations are shown in Table 2.2.

Table 2.2: Circuit 2 parameters.

Symbol	Value	Symbol	Value
B_e	1.5×10^{9} Pa	C_1	4.65×10^{7}
V_1	1.0×10^{-5} m^3	C_2	-1.79×10^{4}
V_2	1.0×10^{-3} m^3	C_3	4.0×10^{11}
k_1	5.62×10^{-7}	C_5	1.02×10^{6}
k_2	5.73×10^{-7}	C_6	5.26×10^{-7}
K	0.05×10^{-9}	C_7	200
p_t	0 Pa	C_8	0.45
k_t	0.1×10^{-7}	C_9	1.2

2.2.3 Mathematical stiffness analysis of fluid power circuit model

To analyse the mathematical stiffness of Circuit 2, its state-space representation should be derived. If we assign the state and input vectors as $\mathbf{x} = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5]^T = [p_1 \quad p_2 \quad K \quad k_t \quad \dot{k}_t]^T$ and $\mathbf{u} = [u_1 \quad u_2 \quad u_3]^T = [p_s \quad U_e \quad \varepsilon]^T$, respectively, then the state equations can be written as follows:

$$\begin{aligned}
\dot{x}_1 &= \frac{B_e}{V_1}\big(x_3\sqrt{u_1 - x_1} - x_4\sqrt{x_1 - x_2}\big)\\
\dot{x}_2 &= \frac{B_e}{V_2}\big(x_4\sqrt{x_1 - x_2} - k_1\sqrt{x_2} - u_3k_2\sqrt{x_2}\big)\\
\dot{x}_3 &= \frac{1}{C_3}(x_2 - x_1 + C_2x_1x_3 - C_2x_3u_1 - C_1x_3 + C_5)\\
\dot{x}_4 &= x_5\\
\dot{x}_5 &= C_6C_7^2u_2 - 2C_7C_8x_5 - C_6C_7^2C_9 - C_7^2x_4
\end{aligned} \tag{2.30}$$

The obtained state-space representation (2.30) is a multi-input multi-output nonlinear model, where ε is the parameter that describes the binary input of the 2/2 directional control valve. If ε is equal to 0 the valve is closed and the term $k_2\sqrt{x_2}$ will also be equal to zero. One way to detect the stiffness in the problem is to estimate the dominant eigenvalues of its Jacobian directly. In linear system theory, the eigenvalues of the system Jacobian describe the behaviour modes inherent in the model. In nonlinear systems, eigenvalues and eigenvectors are time-varying. Nevertheless, it is possible to apply this approach to nonlinear problems through model linearisation. Linearisation means that constantly differentiating nonlinearities are linearly approximated about their operating points. As the linearised solutions can be considered good approximations of nonlinear system solutions about the operating point, the observations obtained locally can be generalised to the rest of the system. Further, to simplify the model we also assume that flows through the compensator and control throttles have constant coefficients K and k_t, which is often valid and frequently employed in fluid power systems design. The Jacobian of the system can be calculated as

$$\mathbf{J} = \left.\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\bar{\mathbf{x}},\mathbf{u}=\bar{\mathbf{u}}} \tag{2.31}$$

where $\mathbf{F}$ is the left-hand side of the first and second equation of (2.30), $\mathbf{x}$ is the model state vector, $\mathbf{u}$ is the model input vector, and $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ is the operating point. The Jacobian for the considered system can be written as follows:

$$\begin{bmatrix} -\frac{B_e}{2V_1}\left(\frac{K}{\sqrt{u_1 - x_1}} + \frac{k_t}{\sqrt{x_1 - x_2}}\right) & \frac{B_e}{2V_1}\frac{k_t}{\sqrt{x_1 - x_2}} \\ \frac{B_e}{2V_2}\frac{k_t}{\sqrt{x_1 - x_2}} & \frac{B_e}{2V_2}\left(\frac{k_1 + k_2}{\sqrt{x_2}} + \frac{k_t}{\sqrt{x_1 - x_2}}\right) \end{bmatrix} \tag{2.32}$$

To characterise the level of numerical stiffness of the model, we employ a condition number of the Jacobian, which, according to numerical analysis theory, can be written as

$$\kappa(\mathbf{J}) = \frac{|\lambda_{max}(\mathbf{J})|}{|\lambda_{min}(\mathbf{J})|} \tag{2.33}$$

where $\lambda_{max}(\mathbf{J})$ and $\lambda_{min}(\mathbf{J})$are the maximum and minimum eigenvalues of the Jacobian, respectively, which for $\mathbf{J} \in \mathbf{M}_{n\times n}$ should satisfy $|\mathbf{J} - \lambda\mathbf{I}| = \mathbf{0}$, where $\mathbf{I}$ is the identity matrix. The condition number shows how much the eigenvalues of the system differ, i.e. small values of κ show that the problem is well-conditioned, whereas large values of κ indicate the ill-conditioned problem and the system can be considered stiff. The condition number can be determined for the certain configuration of the system. This means that the Jacobian should be calculated in the operating point $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$. To define such a point, the physical characteristics of the state variables should be considered. Thus, physical restrictions should be imposed on the state variables and inputs (Table 2.3). Under those restrictions, the operating point can be chosen as $\mathbf{x} = [198\cdot10^5, 1.5\cdot10^5, 10^{-6}, 10^{-7}, 0]^T$ and $\mathbf{u} = [200\cdot10^5, 6]^T$. Note that x_1 and x_2 are calculated from the first and second equation of (9) by substituting x_3 and x_4 with the constant values and assuming that all the rates are equal to zero. At this point, let us consider the two cases. In the first case $V_1 = 10^{-3}$ m^3, i.e. the volume between compensator and control throttle, is quite large. The condition number of (11) in the chosen operating point for this case is κ =

1.28. In the second case the volume is reduced to $V_1 = 10^{-5}$ m^3 and the corresponding condition number becomes as large as $\kappa = 77.81$. Analysing the system Jacobian (2.32), this effect can be seen through the fact that the small volume V_1 appears in the denominator of the Jacobian elements and, thus, makes the eigenvalues differ significantly in magnitude and the mathematical model (2.30) become numerically stiff.

Table 2.3: Physical restrictions imposed on the state variables and inputs.

Variable	Lower limit	Upper limit
x_1	0	-
x_2	0	-
x_3	0	10^{-6}
x_4	10^{-7}	-
u_1	0	-
u_2	0	10
u_3	0	1

2.2.4 Circuit 3: Hydraulic position servo system

The detailed mathematical model of the hydraulic position servo system (HPS) was employed in this work. The model was experimentally verified in (Liu, Wu, Handroos, & Haario, 2012). The fluid power system includes a differential cylinder with an attached sliding load, a 4/3-proportional directional valve and a pump. The system is presented schematically in Figure 2.6. The system is controlled through the voltage signal U supplied to the valve input. The dynamic characteristics of the solenoid valve are presented using the second order model as:

$$\ddot{U}_s = K\omega_n^2 U - 2\zeta\omega_n \dot{U}_s - \omega_n^2 U_s \tag{2.34}$$

where K is the valve gain, U_s is the signal proportional to the valve spool displacement, ζ is the valve damping ratio, and ω_n is the natural angular frequency.

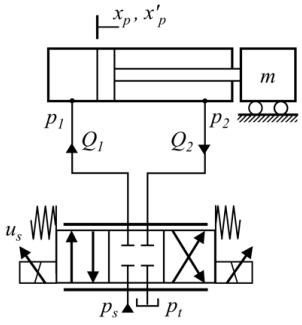

Figure 2.6: Schematic representation of hydraulic position servo system (Circuit 3).

The volume flow rates model of the 4/3-proportional directional valve with turbulent orifices can be presented as:

$$Q_1 = \begin{cases} C_v U_s sign(p_s - p_1)\sqrt{|p_s - p_1|}, U_s \geq 0 \\ C_v U_s sign(p_1 - p_t)\sqrt{|p_1 - p_t|}, U_s < 0 \end{cases}$$
$$Q_2 = \begin{cases} C_v U_s sign(p_2 - p_t)\sqrt{|p_2 - p_t|}, U_s \geq 0 \\ C_v U_s sign(p_s - p_2)\sqrt{|p_s - p_2|}, U_s < 0 \end{cases} \quad (2.35)$$

In (2.35) Q_1 and Q_2 are the volume flow rates produced by the valve, C_v is the flow constant that accounts for cross-sectional areas of the valve orifices. p_1, p_2, p_s, and p_t are respectively the pressures inside the cylinder chambers, the supply pressure, and the pressure in the tank.

As shown in [26], the external leakage model of the servo valve with the critical overlap that accounts for the wear to the edges of the spool and the sleeves can be presented as:

$$\begin{cases} Q_{L1t} = l_{1t} f_L(U_s, U_{s1t})(p_1 - p_t), 0 \leq U_s < U_{s1t} \\ Q_{L2t} = l_{2t} f_L(U_s, U_{s2t})(p_2 - p_t), U_{s1t} < U_s \leq 0 \\ Q_{Ls1} = l_{s1} f_L(U_s, U_{ss1})(p_s - p_1), U_{ss1} < U_s \leq 0 \\ Q_{Ls2} = l_{s2} f_L(U_s, U_{ss2})(p_s - p_2), 0 \leq U_s < U_{ss2} \end{cases} \quad (2.36)$$

where Q_{Lij} are the leakage flows between the respective ports ($ij = 1t, 2t, s1, s2$); l_{ij} are the experimentally defined leakage constants; U_{sij} are the proportional-to-the-position-of-the-spool voltage signals at which the leakage between respective ports i and j disappears; and f_L is the experimentally-defined function that accounts for the shape change as:

$$f_L(U_s, U_{sij}) = 1 - 3\left(\frac{U_s}{U_{sij}}\right)^2 + 2\left(\frac{U_s}{U_{sij}}\right)^3 \quad (2.37)$$

According to Newton's second law, the equation of motion for a double-acting hydraulic cylinder can be written as:

$$m\ddot{x}_p = p_1 A_1 - p_2 A_2 - F_f \quad (2.38)$$

where $\ddot{x}_p$ is the acceleration of the cylinder piston; m is the load mass; 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 respectively; and F_f is the cylinder friction force. In its turn, the friction formed in the cylinder can be represented using the LuGre friction model (Canudas de Wit, Olsson, Astrom, & Lischinsky, 1995):

$$\begin{cases} F_f = \sigma_0 z + \sigma_1 \dot{z} + k_v \dot{x}_p \\ \dot{z} = \dot{x}_p - \dfrac{|\dot{x}_p|}{g(\dot{x}_p)} z \\ g(\dot{x}_p) = \dfrac{1}{\sigma_0}\left(F_C + (F_{st} - F_C)\exp\left(-\left(\dfrac{\dot{x}_p}{v_{st}}\right)^2\right)\right) \end{cases} \quad (2.39)$$

where σ_0 is the flexibility coefficient, σ_1 is the damping coefficient, k_v is the friction coefficient, F_C is the Coulomb friction, F_{st} is the Stribeck friction and v_{st} is the Stribeck

velocity. More specifically, z represents the non-measurable internal state, $g(\dot{x}_p)$ describes the friction behaviour during constant velocity motion, and $k_v\dot{x}_p$ is the viscous friction. The leakage flow between the cylinder chambers can be approximated as:

$$Q_{Li} = L_i(p_1 - p_2) \tag{2.40}$$

where L_i is the laminar leakage flow coefficient. The presented mathematical models of the hydraulic elements are connected with the following continuity equations:

$$\begin{cases} \dfrac{V_1}{B_{e1}}\dot{p}_1 = Q_1 - A_1\dot{x}_p + Q_{Li} - Q_{L1t} + Q_{Ls1} \\ \dfrac{V_2}{B_{e2}}\dot{p}_2 = -Q_2 + A_2\dot{x}_p - Q_{Li} - Q_{L2t} + Q_{Ls2} \end{cases} \tag{2.41}$$

where the chamber volumes are calculated as $V_1 = A_1x_p + V_{01}$ and $V_2 = A_2(H - x_p) + V_{02}$. Here H is the cylinder stroke and $V_{0i}$are the dead volumes connected to the respective ports. In (2.41) the compressibility of hydraulic oil is accounted for by the effective bulk modulus B_{ei}. The effective bulk modulus for each cylinder chamber is calculated regarding the local chamber pressure using the following empirical formula (Jelai & Kroll, 2003):

$$B_{ei} = a_1B_{e\max}\log\left(a_2\frac{p_i}{p_{\max}} + a_3\right) \tag{2.42}$$

where $B_{e\max}$ denotes the maximum bulk modulus of the oil, $p_{\max}$ is the maximum pressure in the system, and a_i (i = 1, 2, 3) are the empirical constants. The values of parameters used in the hydraulic model described in this section are shown in Table 2.4.

Table 2.4: Circuit 3 parameters.

Symbol	Value	Symbol	Value
m	210 kg	L_i	1.59×10^{-12} m^3/sPa
A_1	8.04×10^{-4} m^2	l_{1t}	1.15×10^{-13}
A_2	4.24×10^{-4} m^2	l_{2t}	7.21×10^{-13}
V_{01}	2.13×10^{-4} m^3	l_{s1}	5.96×10^{-13}
V_{02}	1.07×10^{-4} m^3	l_{s2}	2.92×10^{-13}
C_v	2.36×10^{-5} m^3/sV$\sqrt{}$Pa	U_{s1t}	7.94×10^{-1} V
H	1 m	U_{s2t}	-5.92×10^{-1} V
p_s	1.40×10^{7} Pa	U_{ss1}	-9.09×10^{-2} V
p_t	9×10^{5} Pa	U_{ss2}	9.08 V
K	9.90×10^{-1}	σ_0	3.20×10^{2} N/m
ω_n	3.31×10^{2} rad/s	σ_1	6.30 Ns/m
ζ	6.18×10^{-1}	k_v	1.28×10^{3} Ns/m
a_1	0.50	F_C	2.15×10^{6} N
a_2	90	F_{st}	1.13×10^{10} N
a_3	3	v_{st}	3.47×10^{2} m/s
$B_{e\max}$	1.80×10^{9} Pa	p_{max}	2.80×10^{7} Pa

The HPS model presented in this section gives an extremely good fit to the experimental data as demonstrated in (Liu, Wu, Handroos, & Haario, 2012). Thus, performing the mathematical model validation with real experimental data is not necessary.

2.2.5 Circuit 4: Fluid power system with pressure compensating proportional valve

The next fluid power system considered here is the modified and extended version of the previous one. The pressure compensator was added to the 4/3-proportional directional valve and the valve leakage was removed. In the system, the small volume appears between the directional valve and pressure compensator. In Figure 2.7 the small volume and the pressure developing within it are denoted by V_3 and p_3, respectively. Further only the differences from Circuit 3 will be explained.

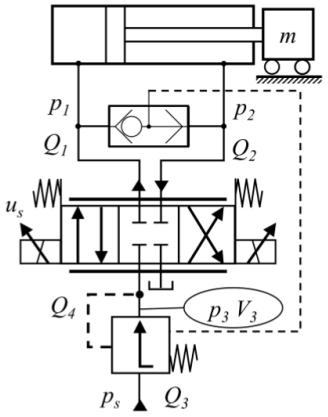

Figure 2.7: Fluid power system with pressure compensating proportional valve (Circuit 4).

The volume flow rates model of the 4/3-proportional directional valve using turbulent orifice model with triangular groove cross-section can be presented as

$$Q_1 = \begin{cases} C_v(U_s - U_{db})^2 \text{sign}(p_s - p_1)\sqrt{|p_s - p_1|}, U_s \geq U_{db} \\ C_v(U_s - U_{db})^2 \text{sign}(p_1 - p_t)\sqrt{|p_1 - p_t|}, U_s \leq -U_{db} \\ 0, \text{otherwise} \end{cases}$$
$$Q_2 = \begin{cases} -C_v(U_s + U_{db})^2 \text{sign}(p_2 - p_t)\sqrt{|p_2 - p_t|}, U_s \geq U_{db} \\ -C_v(U_s + U_{db})^2 \text{sign}(p_s - p_2)\sqrt{|p_s - p_2|}, U_s \leq -U_{db} \\ 0, \text{otherwise} \end{cases} \quad (2.43)$$

In (2.43), C_v is the flow constant that accounts for cross-sectional areas of the valve orifices, U_{db} is the dead band voltage of the valve, and p_1, p_2, p_s, and p_t are the pressures in two cylinder chambers, the supply pressure, and the pressure in the tank, respectively. In this work, the directional valve is assumed as being ideal, such that there are no internal leakages. The volume flow Q_3 related to the pressure compensator is modelled using the semi-empirical approach developed in (Handroos & Vilenius, 1991):

$$Q_3 = K\sqrt{p_s - p_3} \tag{2.44}$$

$$\dot{K} = \frac{1}{C_3}\left(C_5 - p_3 + p_{shuttle} - \left(C_1 + C_2(p_s - p_3)\right)K\right) \tag{2.45}$$

where $p_{shuttle} = \max(p_1, p_2)$ is the output of the shuttle valve. The volume flow Q_4 between valve and pressure compensator can be considered as equal to Q_1 if $U_s \geq U_{db}$, and equal to $-Q_2$ if $U_s \leq -U_{db}$.

The pressures that are building up in the circuit can be calculated from

$$\begin{cases} \dfrac{V_1}{B_{e1}}\dot{p}_1 = Q_1 - A_1\dot{x}_p + Q_{Li} \\ \dfrac{V_2}{B_{e2}}\dot{p}_2 = -Q_2 + A_2\dot{x}_p - Q_{Li} \\ \dfrac{V_3}{B_{e3}}\dot{p}_3 = Q_3 - Q_4 \end{cases} \tag{2.46}$$

Equations (2.43) – (2.46) as well as (2.34), (2.38) – (2.40), (2.42) make up the mathematical model of Circuit 4. The presence in the model of the pressurised small volume makes the mathematical equations stiff and, hence, computationally costly. Parameters used in Circuit 4 are presented in Table 2.5.

Table 2.5: Circuit 4 parameters.

Symbol	Value	Symbol	Value
m	210 kg	C_1	4.65×10^{7}
V_{01}	1×10^{-3} m^3	C_2	-1.79×10^{4}
V_{02}	1×10^{-3} m^3	C_3	4×10^{11}
K_v	9.90×10^{-1}	C_5	8×10^{5}
A_1	8.04×10^{-4} m^2	L_i	1.72×10^{-13} m^3/sPa
A_2	4.24×10^{-4} m^2	V_3	1×10^{-5} m^3
p_t	9×10^{5} Pa	k_v	1.28×10^{3} Ns/m
p_s	1.40×10^{7} Pa	U_{db}	2 V
C_v	2.31×10^{-9} m^3/sV$\sqrt{\text{Pa}}$	$B_{e\max}$	1.80×10^{9} Pa
ω_n	3.31×10^{2} rad/s	$p_{\max}$	2.80×10^{7} Pa
ζ	6.20×10^{-1}	σ_0	3.20×10^{2} N/m
a_1	0.50	σ_1	6.30 Ns/m
a_2	90	F_C	2.15×10^{6} N
a_3	3	F_{st}	1.13×10^{10} N
H	1 m	v_{st}	3.47×10^{2} m/s

2.2.6 **Circuit 5: Fluid power system of the mobile crane**

The fluid power system used in the crane modelling has the following structure, presented in Figure 2.8. The flow is supplied to the system by a variable displacement pressure-compensated

pump. Such a pump maintains the assigned pressure level by adjusting the delivered flow. Three asymmetric double-acting hydraulic cylinders are controlled through the block of the three proportional directional valves with the closed centre position. 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 is 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 the valve opening area.

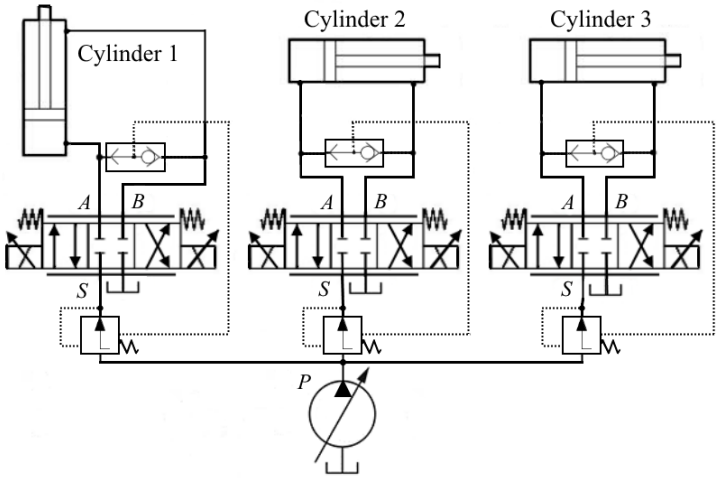

Figure 2.8: Fluid power system of the mobile crane (Circuit 5).

The hydraulic circuit, which is used for modelling, is schematically presented in Figure 2.8. Each hydraulic cylinder produces the force that can be derived using (2.38), where the friction force in cylinder F_f accounts for its velocity dependency as follows (Andersson, Söderberg, & Björklund, 2007):

$$F_f = F_C \tanh\frac{\dot{x_p}}{v_C} + k_v \dot{x_p} \tag{2.47}$$

In (2.47), F_C is a Coulomb friction; v_C is a Coulomb velocity threshold; $\dot{x_p}$ is a cylinder velocity; and k_v is a viscous friction coefficient.

The volume flows created by the proportional 4/3 spool valve with positive overlap and the closed centre position can be modelled as:

$$Q_A = \begin{cases} C_v(U - U_{db})^2\sqrt{|p_S - p_A|}\,\text{sign}(p_S - p_A), & U \geq U_{db} \\ -C_v(U + U_{db})^2\sqrt{|p_A - p_T|}\,\text{sign}(p_A - p_T), & U \leq -U_{db} \\ 0, & -U_{db} < U < U_{db} \end{cases} \tag{2.48}$$

$$Q_B = \begin{cases} C_v(U - U_{db})^2\sqrt{|p_B - p_T|}\,\text{sign}(p_B - p_T), & U \geq U_{db} \\ -C_v(U + U_{db})^2\sqrt{|p_S - p_B|}\,\text{sign}(p_S - p_B), & U \leq -U_{db} \\ 0, & -U_{db} < U < U_{db} \end{cases} \tag{2.49}$$

$$Q_P = \begin{cases} Q_A, & U \geq U_{db} \\ -Q_B, & U \leq -U_{db} \\ 0, & -U_{db} < U < U_{db} \end{cases} \tag{2.50}$$

where p_A, p_B, p_S and p_T are pressures in cylinder chambers, supply pressure and pressure in the tank respectively; U_{db} is the voltage proportional to the positive overlap of the valve; and C_v is the coefficient that accounts for the dependency between the 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 factor $(U \pm U_{db})^2$ is used to model the effect of valve grooves with triangular cross-sections. In order to introduce the pressure compensators to the valves in the model, the pressure drops $p_S - p_A$ and $p_S - p_B$ should be set to 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 a spool position.

The flow supplied by the pressure compensated pump can be calculated as:

$$\dot{Q}_P = \frac{k_P(p_{ref} - p_S) - Q_P}{\tau_P} \tag{2.51}$$

where k_P is the flow-pressure coefficient of the pump; τ_P is the pump time constant; and p_{ref} is the reference pressure of the pump. Finally, the pressures in the system can be calculated using continuity equation accounting for inlet and outlet volume flows for each considered volume as:

$$\dot{p}_S = \frac{B_e}{V}(Q_P - Q_S) \tag{2.52}$$

$$\dot{p}_A = \frac{B_{eA}}{V_0 + A_1 x_p}\left(Q_A - A_1 \dot{x}_p\right) \tag{2.53}$$

$$\dot{p}_B = \frac{B_{eB}}{V_0 + A_2\left(H - x_p\right)}\left(-Q_B + A_2 \dot{x}_p\right) \tag{2.54}$$

Here B_e is the bulk modulus of the volume V emerging between the pump and the corresponding directional valve. For each considered hydraulic cylinder B_{eA} and B_{eB} represent the corresponding bulk modulus of the volumes of the cylinder chambers; V_0 is the dead volume; x_p and $\dot{x}_p$ are the piston position and velocity respectively; H is the piston stroke.

2.3 Mobile crane modelling using commercial software

One of the types of the modelling implemented in the work is the crane modelling using MATLAB/Simulink commercial software. For the multibody system of the crane, a Simulink extension Simscape Multibody was used. In its turn the fluid power system was modelled with basic Simulink blocks using equations presented in the previous subsection. The Simscape Multibody was chosen since it provides an opportunity to use the CAD drawings of the crane components directly together with their size and weight characteristics calculated beforehand in other engineering CAD software. Moreover, automatically generated 3D animation by Simscape Multibody allowed the system dynamics to be visualised.

In Figure 2.9 and Figure 2.10 the resulting block diagram of the crane in Simscape Multibody and a screenshot of the generated 3D animation are presented. In Figure 2.10 the separate bodies are shown in different colours. The crane model was built using only revolute and prismatic joints. The prismatic joints were used only in the modelling of the sliding motion of the hydraulic cylinders and the extension boom with respect to the outer boom. In the resulting model the forces created by the cylinders are calculated in the fluid power model and then taken as inputs into the multibody model. In its turn, the positions and velocities of the cylinder rods are calculated in the multibody model and further supplied back to the fluid power model. The crane motion control in the model is realised through control voltages supplied to the directional valves.

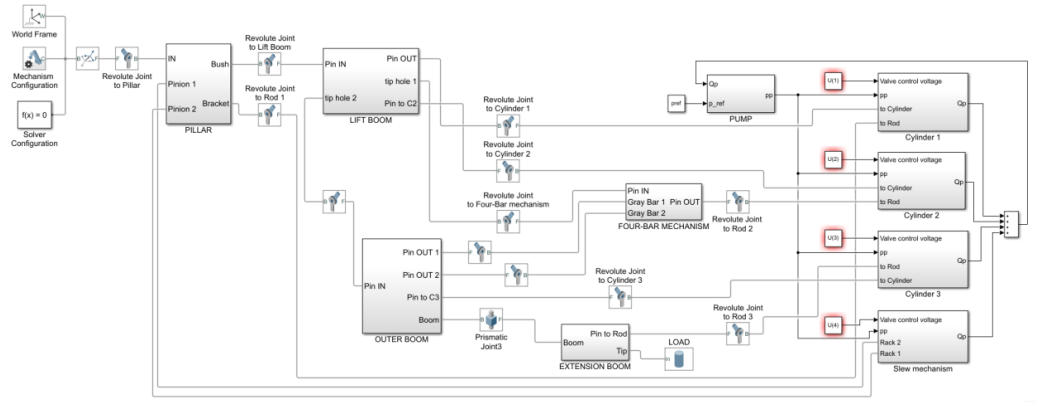

Figure 2.9: Mobile crane model in MATLAB/Simulink and Simscape Multibody.

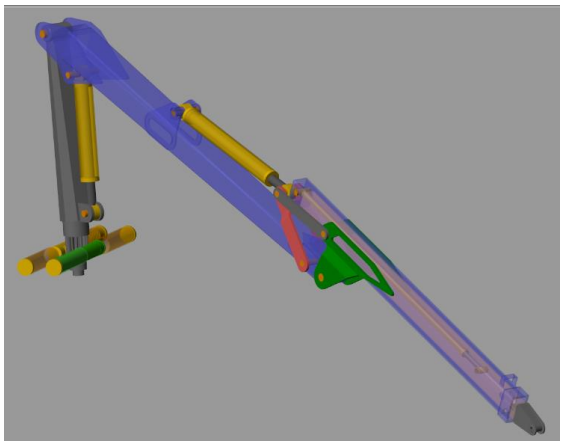

Figure 2.10: Screenshot of the generated 3D animation of mobile crane.

2.4 Mobile crane simulation: results and discussion

In the work, the model obtained using commercial software was taken as a reference of calculation speed and accuracy. The model was calculated using two fixed-step solvers *ode 1* and *ode 4*, which exploit the 1st order Euler numerical method and the 4th order Runge-Kutta method, respectively. In general, the Euler solver, due to its structure, provides faster but less accurate simulation results when compared with the Runge-Kutta solver. A time step as small as 10^{-4} s for both solvers was used. The reference models were translated to C source code using MATLAB Embedded Coder. The program was optimised for its execution efficiency. The code also enables the logging to the file of the main variables, such as coordinates of tip location and values of the forces produced by actuators. The derived mathematical model was directly implemented as C source code. All programs were executed on a personal computer running 2.26 GHz Intel(R) Core(TM) 2 Duo CPU and 4 GB of RAM.

The movement of the crane was modelled from the static initial position. The cylinders of the lifting boom and outer boom were actuated by providing the input voltage of 5 V to their control valves. All other cylinders were fixed in their initial positions. The movement during a time period of 5 s was simulated. In Figure 2.11 global coordinates of the crane tip produced by the simulation of the reference and mathematical INEF models are presented for comparison. The resulting coordinates of the crane tip calculated by the models differ less than 5 cm from each other. Thus, the accuracy of the models can be considered as comparable. The time spent for the execution of each program implementing different models is compared in Figure 2.12. The black line indicates the real-time period of 5 s. Any values of time below this line can be considered faster than real time, while the value above the line indicates that simulation overflow. The program implementing INEF method demonstrates the fastest execution. It consumes less than half of the real-time period that is simulated and provides the maximum amount of time for analysis of predicted system behaviour.

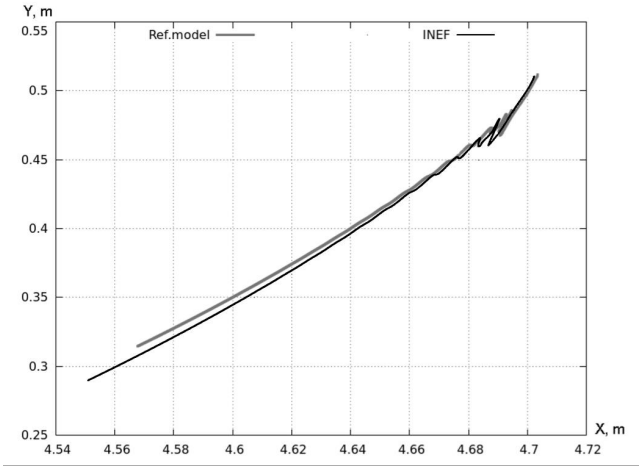

Figure 2.11: Trajectory of the boom tip: Accuracy comparison.

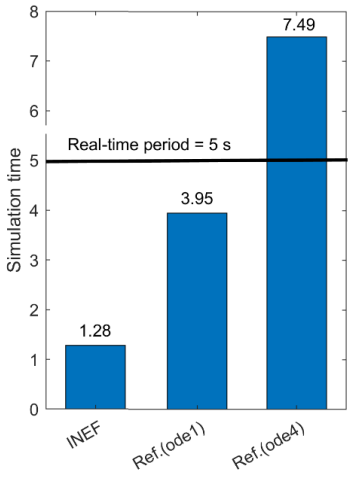

Figure 2.12: Performance comparison.

The advantage of the direct mathematical modelling built with the use of the INEF method can be seen in the ability of its fine tuning in terms of performance and portability. The implementation of such a model does not depend on any particular programming language, software library, operating system or hardware platform. At the same time the commercial software modelling ensures simplicity of implementation, and even the complex mechatronic systems in certain cases can also be used in real-time applications.

3 Fast simulation of hydraulic models using recurrent neural network

In this chapter the modelling and fast simulation of the fluid power system using an artificial neural network with recurrent architecture is studied. As an example of the fluid power system, a mathematical model of the HPS system (Circuit 3) is considered. The system was chosen since it includes the features that are typical for the wide variety of fluid power system models, such as stiff differential equations, strong nonlinearities and non-smooth dependencies. Such models are complex and very time-consuming to solve with conventional numerical integration methods.

3.1 Recurrent neural network architectures

A number of the RNN architectures can be used for dynamic system modelling, such as the NFIR, NARX and the NARMAX architectures. These three architectures are similar in their structure; however, they differ in how much previous information is recurrently supplied to the network input, which, in turn, affects the network prediction accuracy, stability, size and calculation time.

The NFIR architecture can be considered the simplest and can be presented by the following defining equation (Schram, Verhaegen, & Krijgsman, 1996):

$$\hat{\mathbf{y}}(t) = \psi_H\big(\mathbf{x}(t), \mathbf{x}(t-1), \dots, \mathbf{x}(t-n_x)\big) \tag{3.1}$$

where $\mathbf{x}(t) = [x_1(t), \dots, x_m(t)]^T$ is the network input vector at time *t*; n_x is the time delay order for input; and ψ_H is the nonlinear mapping performed by multilayer feedforward network with *H* layers. In (3.1) and in the following equations the hatted variables represent the network estimations of the states of the dynamic system to be modelled. According to (Schram, Verhaegen, & Krijgsman, 1996), the NFIR architecture ensures stable simulation as long as the time of dynamic response is fully covered by past inputs. This leads to a larger number of parameters to be estimated in comparison with the NARX architecture, for example. The NARX architecture is an extended version of the NFIR architecture where also the past outputs are recurrently supplied. The defining equation for the NARX architecture is (Siegelmann, Horne, & Giles, 1997):

$$\hat{\mathbf{y}}(t) = \psi_H\left(\hat{\mathbf{y}}(t-1), \dots, \hat{\mathbf{y}}\big(t-n_y\big), \mathbf{x}(t), \mathbf{x}(t-1), \dots, \mathbf{x}(t-n_x)\right) \tag{3.2}$$

where, in addition to described above parameters, $\mathbf{y}(t) = [\hat{y}_1(t), \dots, \hat{y}_l(t)]^T$is the network output vector at time *t* and n_y is the time delay order for output. In comparison to the NFIR, this architecture is inherently less stable, since it operates in a closed loop, i.e. it uses its predictions as additional input. At the same time, the NARX network is considered to be a more accurate approximator. The NARX network can be obtained from the basic multilayer feedforward network by delaying the input vector $\mathbf{x}$ and feeding back the output vector $\hat{\mathbf{y}}$ (Figure 3.1, Figure 3.2).

The predictive performance of the NARX network can be enhanced by also adding the error

information to the network input vector. This approach is used in the NARMAX architecture, whose defining equation can be written as:

$$\hat{\mathbf{y}}(t) = \psi_H\left(\hat{\mathbf{y}}(t-1), \dots, \hat{\mathbf{y}}(t-n_y), \mathbf{x}(t), \mathbf{x}(t-1), \dots, \mathbf{x}(t-n_x), \mathbf{e}(t-1), \dots, \mathbf{e}(t-n_e)\right) \quad (3.3)$$

where $\mathbf{e}(t-1) = \mathbf{y}(t-1) - \hat{\mathbf{y}}(t-1)$ is the network error at time t - 1 and n_e is the time delay order for error. In (3.3) elements presented by **x** are sometimes called "controlled input" and thus **e** elements can be considered as "uncontrolled input". From this point of view, the NARMAX architecture is the most beneficial in the case of the real-world datasets with the unobserved noise as it is explicitly modelled in the network structure. However, in order to identify this kind of architecture, the previous n_e prediction errors are needed. Thus, the need arises for another predictor that has to be used during the training. This makes the identification process for the NARMAX network much more complex than for the NFIR or NARX networks.

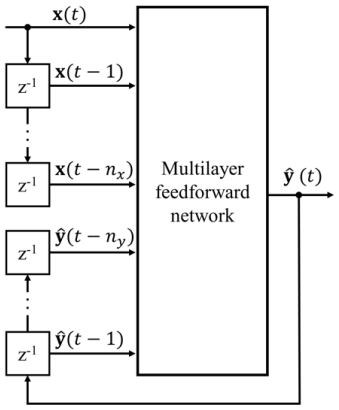

Figure 3.1: Training configuration of the NARX network.

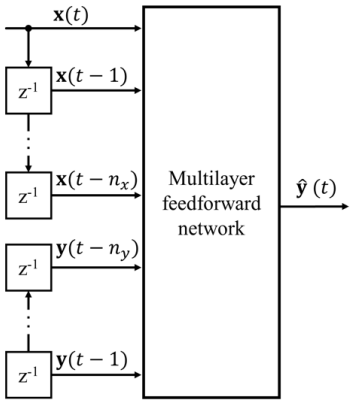

Figure 3.2: General configuration of the NARX network.

In our case, the used datasets do not include any noise as they are produced by a simulation of the mathematical model. Moreover, the smaller number of network parameters means faster calculations during operation, which is our goal. At the same time the prediction accuracy is still important. At this point, it can be concluded that the NARX architecture is seen as the most suitable for our application.

Network training is an iterative process of finding the set of network parameters (weights and biases) that satisfy some optimal criterion, such as mean square error minimum, using training data (a set of samples where each sample includes the input vector and the desired output vector). Using training data, a typical feedforward network can be trained by applying one of the backpropagation-based algorithms (Rojas, 1996). One of the most successful up-to-date training algorithms for this kind of network is the Levenberg-Marquardt (LM) algorithm (Yu & Wilamowski, 2011). In order for the trained network to be not only able to reproduce the training data but also to generalise well, Early Stopping (Demuth, Beale, de Jess, & Hagan, 2014) and Bayesian Regularisation (Demuth, Beale, de Jess, & Hagan, 2014) techniques should

be used. Both the techniques allow network overfitting to be avoided. The latter technique also provides additional information about which part of the neurons is used efficiently.

One of the main advantages of the NARX architecture is that it allows the network to be trained in an open loop configuration. In this configuration during the training, the real output vectors of a dynamic system **y** are fed instead of prone-to-errors estimations $\hat{\mathbf{y}}$, as shown in Figure 3.2. Thus, the accurate input is provided to the feedforward network. Moreover, the resulting network acquires a purely feedforward architecture providing the opportunity to apply standard backpropagation-based algorithms for training. However, during operation the trained network is used in a closed loop configuration (Figure 3.1).

3.2 Training data generation

The mathematical model of Circuit 3 described in detail in Section 2.2.4 was implemented in MATLAB Simulink. The training data was obtained by simulating the model for 4,000 seconds. Each sample of the resulting training data consisted of three values: the input voltage u (V) applied to the solenoid valve and two output values, namely the hydraulic cylinder spool position x_p (m) (with respect to its zero position when the cylinder is fully retracted) and spool velocity $\dot{x}_p$ (m/s). In order to produce adequate training data, which should cover all the regions of the input space where the RNN will be used in the future, a pseudo-random multilevel signal (PRMS) was employed. The PRMS was placed in the range -10…10 V and supplied as the input. The explicit Runge-Kutta solver with a fixed time step 10^{-4} s was used to solve the model. The integration time step of such a small size is usually considered safe for simulation, as it ensures the numerically stable solution of the hydraulic model.

3.3 Pre-processing technique

Let us examine the training data produced by the simulation of Circuit 3 more closely. Figure 3.3 shows a magnification of the areas within the circle. In Figure 3.3 each point represents the separate training example for the neural network. It is obvious that the closest neighbours of each point will represent the similar training examples. During the training process these examples do not bring much new information to the network while making the process longer. In order to speed up the training process and concentrate on the temporal information carried by the sequence, in this study we developed a technique according to which only each 100th example of the initial dataset was used for the training. Thus, the new training data slightly differed from the initial one in terms of shape, while they differed significantly in length (Figure 3.4). The absolute measure of fit of the new data to the initial data was with RMSE of 3.87×10^{-5}. On the other hand, the new training data comprised only 339,000 samples. Part of the training data is presented in Table 3.1. Further, for training purposes, the new data was divided into training, validation and test subsets as 70/15/15%.

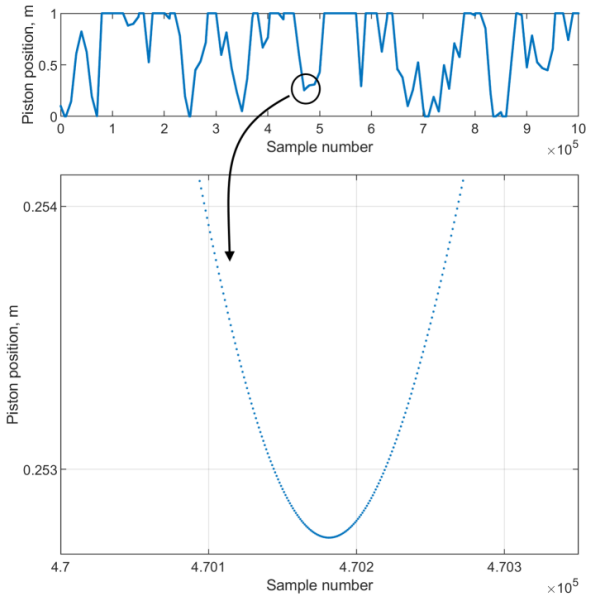

Figure 3.3: Closer look at the training data obtained by numerical integration of the mathematical model (time step 0.1 ms).

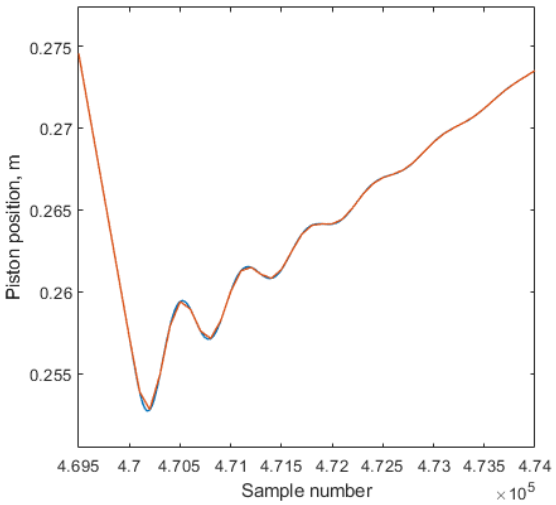

Figure 3.4: Difference between the initial (smoother line) and pre-processed training data.

Table 3.1: Training data layout.

Sample number	Input u, V	Output 1 x_p, m	Output 2 $\dot{x}_p$, m/s
195	1.1866	0.1328	0.1433
196	1.1866	0.1342	0.1433
197	1.1866	0.1357	0.1433
198	1.1866	0.1371	0.1433
199	1.1866	0.1385	0.1433
200	1.1866	0.1400	0.1433
201	4.0328	0.1414	0.1433
202	4.0328	0.1430	0.2023
203	4.0328	0.1460	0.4127
204	4.0328	0.1509	0.5361
205	4.0328	0.1564	0.5384

3.4 Results and discussion

The single input multiple outputs (SIMO) NARX network was implemented using the Neural Network Toolbox. The number of hidden layers, the number of neurons in each layer and the delay parameters were selected by trial and error until the desired accuracy was obtained. The resulting network included four fully connected hidden layers with sigmoid activation functions, and one output layer with a linear activation function. The number of input and output delays was determined by the experiment described further in the section.

One of the major differences between the NARX network presented in this paper and the networks studied in other works (for example in (Patel & Dunne, 2003)) is the network size. The present network has much deeper architecture that allows for the approximation of more complex functions with multiple inflections inherent to the hydraulic model.

Let us address the stability issues of the network. As was noted previously, adding global feedback to a network can lead to instability in an otherwise stable open loop network. One way to assess the stability of a trained neural network is to apply Lyapunov stability analysis. However, it can be considered a rather tedious approach (Barabanov & Prokhorov, 2002). In our work we implemented a more practical approach consisting of multiple training with random parameter initialisation (Patel & Dunne, 2003). Following the selected approach, 15 randomly initialised networks of the same structure were trained using the same training data. The networks were then ranked according to their posterior accuracy on the test dataset in closed loop configuration (Table 3.2). Experience shows that the network with minimum posterior error is stable (Patel & Dunne, 2003).

For all networks in Table 3.2 the LM algorithm supplemented by Bayesian Regularisation and Early Stopping were used for training. Training of the network with the highest rank took 2 hours and 57 minutes for 293 epochs and stopped by Early Stopping regularisation (the validation subset error continuously increased for 6 epochs). Figure 3.5 shows the evolution of the training, validation and test performances of the open loop network during the training. The best validation performance at epoch 287 was 7.94×10^{-6}, whereas training and testing performances were 4.62×10^{-6} and 7.36×10^{-6}, respectively. The number of parameters (weights

and biases) being effectively used by the network was equal to 956. The closeness of this value to the total number of parameters, i.e. 1,053, showed that the size of the network was appropriately chosen. After the training, the network was transformed to the closed loop architecture. A dataset that included 1,000 samples and corresponded to 10 seconds of real time was used for testing. The output accuracy of the trained NARX network with the highest rank was $MSE = 1.87\times10^{-4}$.

Table 3.2: Ranking of the 15 trained networks with Random Initialisation.

Rank	MSE_{Train} x 10^{-6}	MSE_{Valid} x 10^{-6}	MSE_{Test} x 10^{-6}	Valid. Stop	MSE_{Test} CL*
1	4.62	7.94	7.36	287	1.87×10^{-4}
2	6.11	11.30	6.87	181	4.83×10^{-4}
3	2.59	4.91	4.51	554	5.03×10^{-4}
4	10.80	17.10	11.60	144	6.04×10^{-4}
5	2.62	34.70	27.40	1000	6.93×10^{-4}
6	2.08	2.76	4.06	209	6.98×10^{-4}
7	6.09	10.80	8.73	415	7.01×10^{-4}
8	9.96	10.30	11.30	172	7.02×10^{-4}
9	5.95	10.40	7.17	327	0.25×10^{-2}
10	5.77	9.74	7.45	105	0.77×10^{-2}
11	3.98	10.40	12.10	373	0.69
12	3.54	8.31	8.02	453	1.09
13	4.16	7.29	6.16	110	1.31
14	8.61	12.50	10.50	159	2.38
15	5.66	10.40	7.59	422	3.59

* Closed loop.

The additional experiments to study the influence of the number of input and output time delays on the accuracy of the considered network were carried out. The results are summarised in Table 3.3. It should be noted that all seven networks were initialised with the same parameters. The visual inspection of the produced responses was also performed. The responses of the most accurate networks (Case 5 and Case 6 in Table 3.3) are presented in Figure 3.6. Although the overall accuracy of network Case 5 is the highest, network Case 6 represents the transition process in velocity in a more natural way. Thus, network Case 6 might be a better choice for the simulation.

Table 3.3: Network performances with different number of input/output time delays.

Case	Input TD**	Output TD**	MSE_{Train} × 10^{-6}	MSE_{Test} × 10^{-6}	MSE_{Test} CL*
1	0…5	1…6	2.74	7.44	7.71×10^{-4}
2	0…6	1…7	6.39	7.79	3.37×10^{-4}
3	0…7	1…8	3.49	7.04	8.39×10^{-2}
4	0…8	1…9	7.71	10.20	2.99×10^{-4}
5	0…9	1…10	4.66	6.20	4.70×10^{-4}
6	0...10	1…11	4.62	7.36	1.87×10^{-4}
7	0…11	1…12	4.63	7.88	0.12

* Closed loop.
** Time delay.

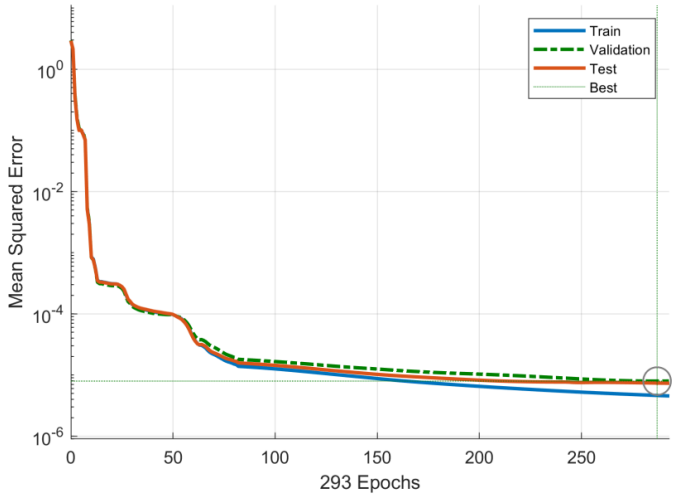

Figure 3.5: Training, validation and test performances of the open-loop NARX network during training.

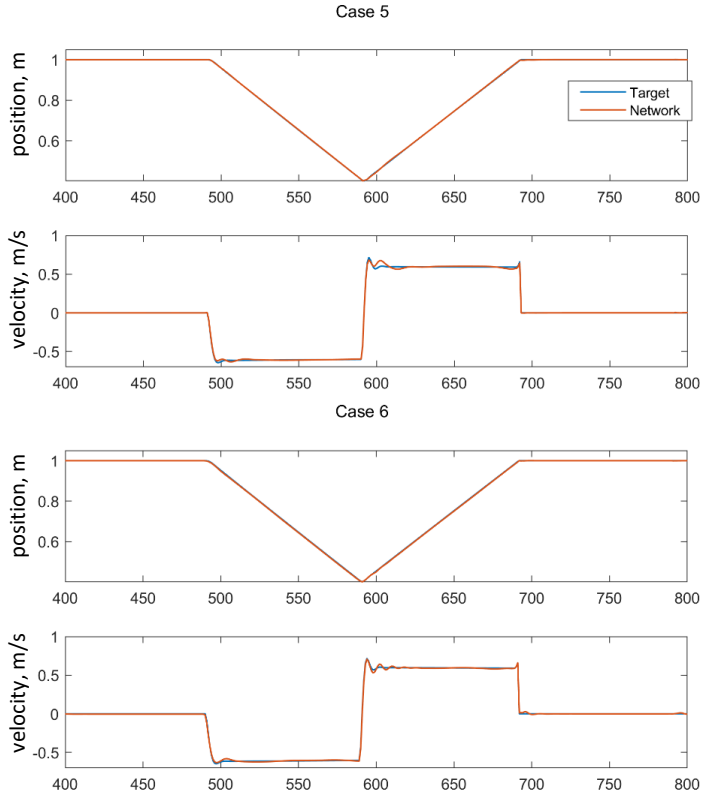

Figure 3.6: Position and velocity responses of the most accurate networks (Case 5 and Case 6 in Table 3.3).

In order to compare the calculation speed on the test dataset, the reference mathematical model implemented in Simulink and the trained NARX network (Case 6) were translated in stand-alone C code using MATLAB Embedded Coder 7.0. The C code was optimised for execution efficiency. Both codes were compiled and run on a personal computer that had a 2.26 GHz Intel(R) Core(TM) 2 Duo CPU and 4 Gb of RAM. It took 78 ms for the mathematical model to simulate an interval of 10 seconds of real time that was 128 times faster than real time, whereas for the NARX network model it took only 16 ms to simulate the same time interval that, in turn, was 625 times faster than real time (see Table 3.4). Thus, for the considered case of the hydraulic system the NARX network model is able to provide the solution 4.8 times faster than the reference mathematical model. Moreover, the visual compression of the NARX network model and mathematical model responses to a longer test dataset (30,000 samples) revealed very good generalisation capabilities of the developed NARX network model (Figure 3.7). For readability reasons only some of the samples are shown in Figure 3.7. By analysing the results presented in Figure 3.7, one can notice the regions with lower accuracy. However, these regions are transitional and often retain the form of the target model.

Table 3.4: Computational time of stand-along C codes of the mathematical model and the model produces by the proposed method.

Modelling approach	**Computational time, ms (corresponds to 10 s of real time)**	**Times faster than real time**
Mathematical	78	128
RNN network (+ pre-proc. tech.)	16	625

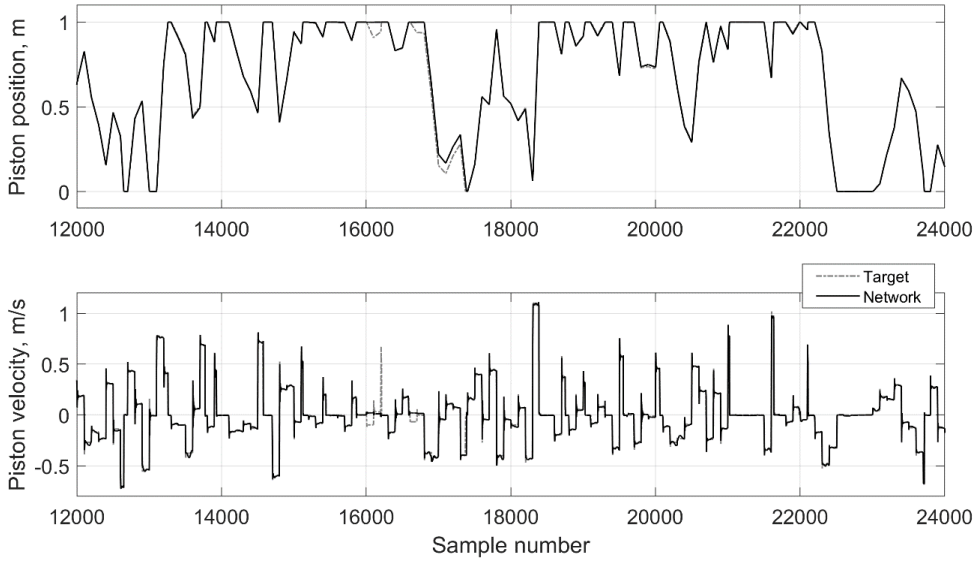

Figure 3.7: Comparison between the responses of the mathematical model (Target) and NARX network-based model (Network) using the test dataset.

The obtained results show that the NARX network with deep architecture and proper chosen size provides an accurate solution for the considered hydraulic system and can be used for simulation purposes. In addition to the benefits provided by the NARX network-based modelling approach, the pre-processing technique developed in this work allows the simulation of the hydraulic system to be sped up. In particular, for the considered HPS model the NARX network-based simulation was 4.8 times faster than the mathematical model-based simulation.

It should be noted that both simulations of the considered hydraulic system were performed faster than real time. However, with the increasing size of the system model (by adding more hydraulic components such hydraulic cylinders and valves and by adding the multibody model of the mobile machine) or with increasing model complexity (by taking into account the small volumes of the valves and/or the flexibility of the hydraulic components), the computational time increases for the mathematical model-based simulation much faster than for the NARX network model-based simulation. The reason for this is that the RNN calculation speed is mainly related to the number of network parameters and only partly depends on the complexity of the mathematical model, which is used for the training. Thus, it can be expected that the number of the network parameters will not increase much as the number considered in this work is already sufficient to represent rather complex dynamics.

Another aspect that deserves special attention is that any ANN with adequate training can achieve the good performance with interpolation problems and at the same time has quite poor behaviour with extrapolation problems. Here interpolation and extrapolation are considered with respect to the training set. Fortunately, the fluid power systems often have limited number of states that can be formulated as interpolation problem. In the work, the RNN is used only as a tool for simulation acceleration. All the dynamics of the fluid power system is provided by the mathematical model that is used for the RNN training. Thus, in the case of another fluid power system the dynamics of interest should be reflected in the mathematical model. Then the structure (inputs/outputs number and character, neurons number) of the RNN should be revised in a way that will help the network to reproduce the dynamics correctly.

4 An efficient method for solving the fluid power models with singularities

In this chapter on simulation acceleration, the method based on the developed advanced pseudo-dynamic solver with adaptive criterion (AdvPDS) is used. This solver artificially reduces the numerical stiffness of the mathematical model of the fluid power system, which is caused by the presence of a small volume in the system (Malysheva, Ustinov, & Handroos, 2020; Malysheva & Handroos, 2020). Also in the chapter, the effect of the three different fixed-step integration methods (Euler, Runge-Kutta of fourth order, and modified Heun's method) used within AdvPDS are considered. The numerical stability of the modified Heun's method was improved by substituting the purely turbulent orifice model with the two-regime orifice model. The two-regime orifice accounts for both the turbulent and laminar flows and thus allows the numerical problems related to the small pressure drops to be avoided.

4.1 Classical pseudo-dynamic solver

The classical pseudo-dynamic solver (PDS) was proposed and studied in (Åman & Handroos, 2008; Åman & Handroos, 2010; Åman R. , 2011; Pedersen H. , 2007). The main purpose of the PDS is to find a steady-state solution for the pressure building up in a small volume. The solver outputs the pressure value only after its steady state is reached using derivative convergence criterion. Then PDS includes two integration loops: the main loop, which contains algebraic and differential equations related to larger volumes, and the inner loop. The inner loop, using artificially enlarged fluid volumes, searches for the steady-state value of pressure passing by the transition process of pressure formation. The steady-state value of pressure is sought out during the single time step of the main loop. The PDS is used only for small volume presser integration. For all other parts of the hydraulic model the common numerical integration method is used.

The pressure inside the inner loop can be calculated from the classical continuity equation using an artificially enlarged fluid volume as follows:

$$\dot{p} = \frac{B_e}{V_{pseudo}}(Q_{in} - Q_{out}) \tag{4.1}$$

where V_{pseudo} is the artificial pseudo-volume, and Q_{in} and Q_{out} are the inlet and outlet volume flows, respectively. According to (Malysheva, Ustinov, & Handroos, 2020), it is recommended that the pseudo-volume is set at least 10 times higher than the actual volume. Inlet and outlet volume flows can be expressed as a function of pressure drop as follows:

$$Q = f(\Delta p) \tag{4.2}$$

The integration of the differential equation inside the inner loop of classical PDS occurs using an explicit fixed-step fourth order Runge–Kutta integration routine with an independent sufficiently small time step t_i. The integration routine continues until the convergence criterion is reached. The criterion is a predefined user parameter, which represents the first derivative of the pressure. It is important to note that the activation of the inner loop suspends the main loop

until the steady-state pressure value is found. In general, the PDS allows the hydraulic model overall stiffness to be reduced by neglecting the high order dynamics that do not play a major role in the dominant dynamic behaviour of the mobile machine.

4.2 Advanced pseudo-dynamic solver with adaptive criteria

To study the characteristics of the PDS described in Section 4.1, a fluid power system (Circuit 2) was employed. The three random signals in the form of pseudo-random multilevel signals (PRMSs) were supplied as the inputs: supply pressure p_s in the range 14–20 MPa, control voltage to the control throttle U_e in the range −10 to +10 V, and directional valve control signal U_d, which took either 1 when it is open or 0 when it is closed (Figure 4.1). The signals were supplied asynchronously with a period of 0.5 s. The system was simulated using a conventional fourth order Runge–Kutta integrator with a sufficiently small time step of 10^{-6} s for 100.5 s.

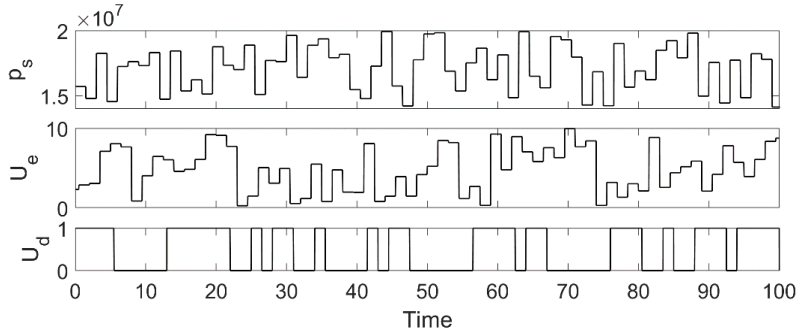

Figure 4.1: Input signals for Circuit 2.

During the simulation, the small volume V_1 is equal to 10^{-5} m^3. The simulation took about five hours using the following simulation environment: MATLAB 2018b, Intel Core i5-4590 3.30 GHz with 16 GB of RAM, running OS Windows 7 64-bit. The fourth order Runge–Kutta solver is considered further in this work as a reference solver and the solutions obtained with its help are thus also considered as a reference. The solution for the pressure p_1 was obtained under such conditions and was used as a reference in the case of Circuit 2. The classical pseudo-dynamic solver was introduced in the same simulation using recommended parameters. Unfortunately, it could not achieve a stable solution without compromising its speed and accuracy. While studying the reasons for such a behaviour, it was discovered that the solver becomes numerically unstable in areas of sudden pressure change, owing to fixed Q_{out} in (4.1) during integration in the inner loop. To stabilise the numerical solution, it was decided to move the calculation of Q_{out} into the inner loop. Thus, in terms of Circuit 2, the pressure build-up in the small volume (2.22) as well as inlet and outlet volume flows described by (2.24) and (2.25), respectively, are calculated in the inner loop. However, it was also found that the calculation of other system elements such as K and k_t inside the inner loop does not have much of an effect on the solution accuracy; moreover, it makes the simulation longer. These findings formed the basis for the AdvPDS.

In the classical PDS, the single convergence criterion was used. The criterion was based on the rate of pressure change between the iterations in the inner loop. The rate of pressure change between the iterations Δp_1 can be written as:

$$\Delta p_1 = p_1 - p_{1prev} \tag{4.3}$$

where p_1 is the pressure from the current iteration, and p_{1prev} is the pressure from the previous iteration of the inner loop. The rate of pressure change is compared with the convergence criterion value to detect the beginning of the steady-state process of the pressure. The captured steady-state value is further passed to the main loop. The effect of the single criterion value in the inner loop of the AdvPDS on the solution accuracy was also studied. It was discovered that applying a smaller convergence criterion in the inner loop produces a more numerically stable result when the pressure approaches its lower values during the simulation. In Figure 4.2 the effect of the criterion value on the calculation of the low pressures using the AdvPDS is shown. At the same time, it was noted that the computational time of the simulation increases with the criterion decrease, owing to the large number of iterations performed inside the inner loop. Thus, the adaptive convergence criterion was proposed. The idea behind the adaptive criterion is that depending on the pressure level, the criterion with the most suitable time efficient and numerically stable effect on the pressure is automatically selected during the fluid power circuit simulation.

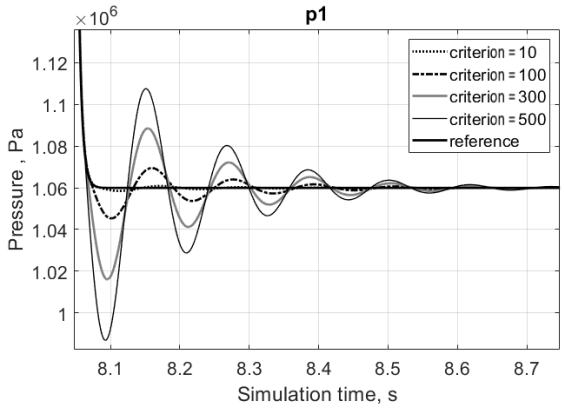

Figure 4.2: Effect of the criterion value on the low-pressure calculation using the AdvPDS.

When the pressure difference between the iterations Δp_1 is calculated using (4.3), the current pressure level p_1 is compared to the assigned low-pressure level p_{limit}. If the current pressure level is low, the smaller criteria is used, i.e. the inner loop continues to iterate until the change in the pressure is less than $p_{1\ tol\ low}$. If $p_1 > p_{limit}$ the inner loop proceeds with criteria $p_{1\ tol\ high}$. In other words, at low pressure levels in the system, the smaller criterion is implemented to achieve a more numerically stable result. At pressure levels higher than the low-pressure limit, the bigger criterion is used to reduce the computational time of the simulation. The low-pressure level was defined experimentally and for both Circuits 2 and Circuit 4 it was 22 bar. Both criteria have to be predefined by the user before the simulation, based on the recommendations given further in this work. When the criterion is satisfied, the value of pressure p_1 and flow Q_t are updated for subsequent calculations in the main loop. The main

loop further updates the pressure p_2, and flows Q_{e1} and Q_{e2} according to (2.23), (2.28), and (2.29), respectively. Then the calculation of flow coefficients K and k_t according to equations (2.26) and (2.27) is performed. The next iteration of the main loop begins at the next time step Δt. The process continues for all the specified simulation time.

In the work (Malysheva, Ustinov, & Handroos, 2020) the subtlety of the classical PDS was improved by modifying the original structure and introducing the second convergence criterion for the small pressures (AdvPDS). In order to show the operating principle of the AdvPDS, let us consider the simple Circuit 1 with two orifices presented in Section 2.2.1 by the equations (2.18) –(2.21).

An algorithm for the AdvPDS built in the structure of this fluid power circuit for pressure calculation in the small volume is presented below (Algorithm 4.1). It should be noted that inside the loop the Q_1, Q_2 and p_1 are being recalculated during the integration whereas all other parameters, like p_2, are constant.

In Algorithm 4.1 V_{pseudo} is the pseudo volume, $p_{1\ tol\ high}$ and $p_{1\ tol\ low}$ are the convergence criteria for the high-pressure and low-pressure areas, respectively, and t_{max} is the maximum allowed iteration time.

Algorithm 4.1. AdvPDS

Input: p_1^{prev}, p_2, V_{pseudo}, dt_i, t_{max},
Output: p_1^{next}, Q_2^{next}
Initialisation:
$t_i = 0$
LOOP :
for $t_i = 0$ to t_{max} **do**
  $t_i = t_i + dt_i$
  use (2.21) to calculate Q_2
  use (2.20) to calculate Q_1
  use (2.18) and $V_1 = V_{pseudo}$ to calculate $\dot{p}$
  calculate $p_1^{current}$ using the numerical integration method
  $\Delta p_1 = p_1^{current} - p_1^{prev}$
  if ($p_1^{current} < p_{limit}$) **then**
    if ($\Delta p_1 < p_{1\ tol\ low}$) **then** $p_1^{next} = p_1^{current}$, $Q_2^{next} = Q_2$
      return p_1^{next}, Q_2^{next}
      break *LOOP*
    else $p_1^{prev} = p_1^{current}$ **goto** *LOOP*
    end if
  else
    if ($\Delta p_1 < p_{1\ tol\ high}$) **then** $p_1^{next} = p_1^{current}$, $Q_2^{next} = Q_2$
      return p_1^{next}, Q_2^{next}
      break *LOOP*
    else $p_1^{prev} = p_1^{current}$ **goto** *LOOP*
    end if
  end if
end for

There are two main differences in the AdvPDS when compared with the classical pseudo-dynamic solver. First, the calculation of the outlet volume flow rate related to the small volume is included in the solver, which allowed the numerical stability of the solution to be increased.

Second, the adaptive convergence criterion is introduced, which allowed the simulation time to be decreased and the calculation accuracy to be increased.

4.3 Numerical Integration Methods for the AdvPDS

In this section, four numerical integration methods that are further used in the framework of the AdvPDS are presented. The first three methods (Euler method, Runge-Kutta method and Heun's method) are the general mathematical numerical integration methods, while the fourth is an ad hoc method developed specially for pressure integration in hydraulic circuits.

4.3.1 Euler Method

Euler's numerical integration method is the simplest and most computationally efficient, since only a single function calculation is performed at each time step. Let us consider an initial value problem of the form:

$$\dot{y} = f\big(t, y(t)\big), y(t_0) = y_0 \tag{4.4}$$

where *t* is the time flow. The governing equation of the Euler method can be written as:

$$y_{n+1} = y_n + dtf(t_n, y_n) \tag{4.5}$$

where *dt* is the integration time step, and y_{n+1} and y_n are respectively the next and the previous values (estimations) of the integrated parameter *y*. However, the applicability of the method for the stiff problem solution is quite limited due to its narrow stability performances (Hairer & Wanner, 1996). The global error of Euler method is proportional to the maximum step size (Hairer & Wanner, 1996).

4.3.2 Runge-Kutta Method

The well-known Runge-Kutta method of the fourth order has a larger stability area and provides a more accurate approximation for the integrating parameter compared with the Euler method. However, it is more computationally expensive as it requires four function calculations at each time step. The method can be written in the form:

$$s_1 = f(t_n, y_n) \tag{4.6}$$

$$s_2 = f\left(t_n + \frac{dt}{2}, y_n + \frac{dt}{2}s_1\right) \tag{4.7}$$

$$s_3 = f\left(t_n + \frac{dt}{2}, y_n + \frac{dt}{2}s_2\right) \tag{4.8}$$

$$s_4 = f(t_n + dt, y_n + dts_3) \tag{4.9}$$

$$y_{n+1} = y_n + dt\left(\frac{s_1}{6} + \frac{s_2}{3} + \frac{s_3}{3} + \frac{s_4}{6}\right) \tag{4.10}$$

4.3.3 Heun's Method

The third numerical integration method considered in this work is Heun's method. This is a two-stage method that first uses Euler's equation for the calculation of the intermediate value for the integrating parameter (predictor) and then the implicit trapezoidal method for the final approximation (corrector). The procedure for Heun's predictor-corrector method can be written as:

$$\hat{y}_{n+1} = y_n + dt * f(t_n, y_n) \tag{4.11}$$

$$y_{n+1} = y_n + \frac{dt}{2}[f(t_{n+1}, \hat{y}_{n+1}) + f(t_n, y_n)] \tag{4.12}$$

The accuracy of the solution ensured by the Euler method improves linearly with the step size decrease (the error is *O(dt)*). However, the solution accuracy obtained using Heun's method improves quadratically (the error is $O(dt^2)$). In its turn the accuracy of the Runge-Kutta method is $O(dt^4)$.

4.3.4 Modified Heun's Method with improved stability

In the work (Ellman A. , 1992) the above-described Heun's method was modified and adopted for the pressure integration in fluid power circuits. The convergence of the method was improved by means of Newton's single iteration method. For the simple fluid power circuit (Circuit 1) the predictor equation of the modified Heun's method can be written as:

$$\hat{p}_1^{next} = \frac{\hat{p}_1^{prev} + dt\frac{Be}{V_1}\left(Q_\Sigma^{prev} - \frac{\partial Q_\Sigma^{prev}}{\partial p_1^{prev}} p_1^{prev}\right)}{1 - dt\frac{Be}{V_1}\frac{\partial Q_\Sigma^{prev}}{\partial p_1^{prev}}} \tag{4.13}$$

where $\hat{p}_1^{next}$ is the estimate of the next pressure value, Q_Σ^{prev} is the total volume flow calculated with respect to the previous pressure value, and *dt* is the integration time step. In its turn, the corrector equation can be presented as:

$$p_1^{next} = \frac{p_1^{prev} + \frac{dt}{2}\frac{Be}{V_1}\left(Q_\Sigma^{prev} + \hat{Q}_\Sigma^{next} - \frac{\partial \hat{Q}_\Sigma^{next}}{\partial p_1^{next}} \cdot \hat{p}_1^{next}\right)}{1 - \frac{dt}{2}\frac{Be}{V_1}\frac{\partial \hat{Q}_\Sigma^{next}}{\partial \hat{p}_1^{next}}} \tag{4.14}$$

where p_1^{next} is the resulting pressure value, and $\hat{Q}_\Sigma^{next}$ is the total volume flow calculated with respect to the estimate of the next pressure value. It should be noted that both the predictor (4.13) and corrector (4.14) equations include a partial derivative of the total flow with respect to the pressure. The derivatives originate from the employed Newton single iteration method. The use of the conventional turbulent orifice model (2.19) inside the derivatives may have led to a singularity problem, as the model has singularity in the Jacobian at zero pressure difference (Piché & Ellman, 1994). In particular, when the pressure drop across the turbulent orifice model

nears zero, the first derivative of flow with respect to pressure drop approaches infinity and the second derivative becomes discontinuous (Åman, Handroos, & Eskola, 2008). This model property can cause arithmetical overflow in the numerical integration process. However, this property related only to the mathematical representation of the volume flow. Model (2.19) describes only the turbulent flow through the orifice, whereas in reality the transition from turbulent to laminar flow occurs when the orifice pressure drop is small. Thus, in order to avoid the singularities in the integration process, a more accurate model of the flow through the orifice is needed. For this reason, the two-regime orifice model (Åman, Handroos, & Eskola, 2008) is employed in the work. This model includes both the turbulent flow model as (2.19) and the laminar flow model for the pressure drops close to zero. The laminar flow model is based on the cubic spline approximation and can be written as:

$$Q = a_1\Delta p + a_2\Delta p^2 + a_3\Delta p^3, |\Delta p| < |\Delta p_0| \tag{4.15}$$

where Δp is the pressure difference across the orifice, p_0 is the boundary pressure of transition, and a_1, a_2, a_3 are the coefficients of the cubic spline that can be calculated as:

$$\begin{aligned} a_1 &= \frac{5k_i}{4\sqrt{\Delta p_0}} \\ a_2 &= 0 \\ a_3 &= -\frac{k_i}{4(\Delta p_0)^{5/2}} \end{aligned} \tag{4.16}$$

The physical adequate value for the boundary pressures can be found as (Åman, Handroos, & Eskola, 2008):

$$\Delta p_0 = \pm\frac{Re_{tr}^2 v^2 \pi\sqrt{\rho}}{5.657 C_d k_i} \tag{4.17}$$

where Re_{tr} is the transition Reynolds number, C_d is the value of the discharge coefficient in the turbulent region, v is the kinematical viscosity of hydraulic fluid, and k_i is the semi-empirical volume flow coefficient.

4.4 Simulation results using AdvPDS with the fourth order Runge-Kutta solver

In this section, the results of the simulation of the two fluid power systems Circuit 2 and Circuit 4 are presented. The results are represented through a comparison of the responses of the considered fluid power circuits obtained using the referenced fourth order Runge-Kutta solver and the AdvPDS. The results demonstrate the features of the proposed method and its advantages compared with the traditional method of fluid power system modelling and simulation.

4.4.1 Circuit 2 simulation

Circuit 2 was simulated for 100.5 s using the AdvPDS and the reference solver with the inputs presented in Figure 4.1. The presence of the small volume ($V_1 = 10^{-5}$ m^3) in the circuit between the pressure compensator and control throttle increased the stiffness of the whole system, and also determined the selection of the integration time step for the reference solver. The integration time step of the reference system was set to the largest possible value of 1×10^{-6} s, at which point the solution for the pressure appeared numerically stable. The use of the AdvPDS with Circuit 2 allows the stiffness of the system to be reduced, owing to substitution of the small volume by the larger artificial volume. In the mathematical model of Circuit 2 the artificial volume V_{pseudo} substitutes the real volume V_1 in (2.22). This volume directly affects the resulting pressure p_1. In order to analyse how the size of the artificial volume affects the respective pressure solution, a simple sensitivity analysis was carried out. The sensitivity analysis was performed in a way that Circuit 2 is simulated five times with the same input signals and parameter values except for the pseudo-volume value. In the experiment, V_{pseudo} took the following values: 0.5×10^{-2} m^3, 1×10^{-3} m^3, 0.5×10^{-3} m^3, 1×10^{-4} m^3, and 0.5×10^{-4} m^3. The upper bound of the pseudo-volume range was limited by the system stability while holding the condition $\Delta t_i = 10^{-5}$ s. In Figure 4.3 the results of five simulations are presented. For clarity, only a short range of the simulation time is shown in the figure. It should be noted that the biggest difference in the pressure solutions of five simulations is observed in the transition areas, when one of the control signals was changed. One such area is shown in Figure 4.3. It can be seen from the figure that the four solutions that refer to the smaller pseudo-volumes are rather close to the reference one. Only the solution obtained using the biggest volume 0.5×10^{-2} m^3 compromised the accuracy. Taking into account the obtained results, the artificial volume was set to $V_{pseudo} = 1 \times 10^{-3}$ m^3. On the one hand, the pseudo-volume of this size ensured quite high accuracy of the solution. On the other hand, it allowed the integration time steps for the main and inner loops to increase significantly and to be set to the values of 10^{-4} s and 10^{-5} s, respectively. As previously mentioned, the number of iterations performed in the inner loop at each time step also has a direct effect on the simulation time. The transition process is more oscillatory, and the larger the pressure changes, the more iterations are performed in the inner loop. At the same time, the number of iterations is dependent on the chosen convergence criterion. It was found experimentally that the larger criterion is associated with the smaller number of iterations. Thus, to speed up the simulation of the AdvPDS-based system, the adaptive convergence criterion 300 Pa/10 Pa was selected based on experimental results. In Figure 4.4, the number of iterations performed by the AdvPDS using a single convergence criterion in comparison with the use of the adaptive criterion is shown for the first 20 seconds of the simulation. It can be seen from the figures that the AdvPDS executed a higher number of iterations in transition areas with the single criterion than with the adaptive criterion, which resulted in a shorter simulation time. Figure 4.5 shows the pressure responses p_1 of Circuit 2, obtained with the reference solver and the AdvPDS. One can observe that the two curves are highly coincidental with each other. Now the high accuracy of the AdvPDS-based system was also achieved on the low-pressure areas. The accuracy of the system was represented through root-mean-square error (RMSE). The overall error was RMSE = $1.12\cdot10^4$Pa, which is insignificant for such high-pressure levels in the system. Thus, the use of larger integration time steps together with the adaptive convergence criteria allowed the computational time of the simulation to be reduced compared with the reference system. The simulation time with the reference solver was about five hours, whereas only 147.983 seconds were spent for the same

simulation using the AdvPDS. Moreover, it should be noted that the system with the AdvPDS (in contrast to the use of the classical pseudo-dynamic solver) is numerically stable during the whole 100.5 seconds of simulation (i.e. the solver kept the same pressure level as the reference system).

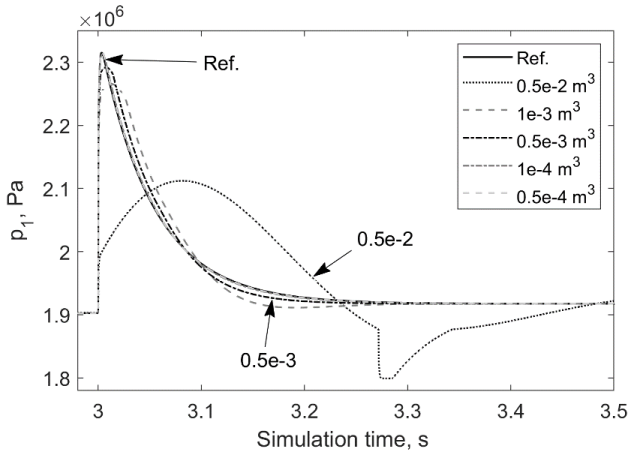

Figure 4.3: Sensitivity analysis of p_1 solution to the changes in V_{pseudo}.

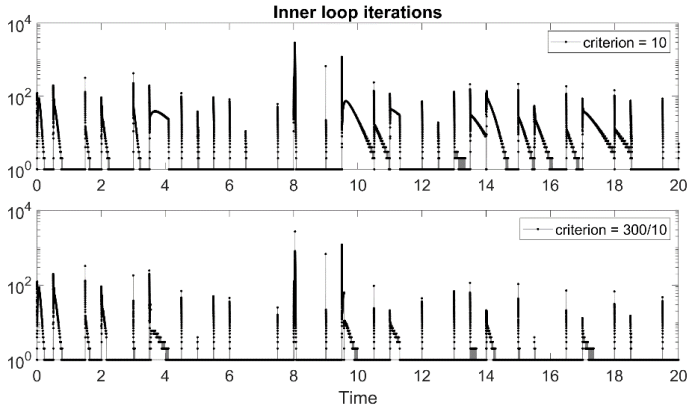

Figure 4.4: Number of iterations performed by the AdvPDS in the inner loop during the first 20 s of the simulation with 10 (upper plot) and 300/10 (lower plot) criteria.

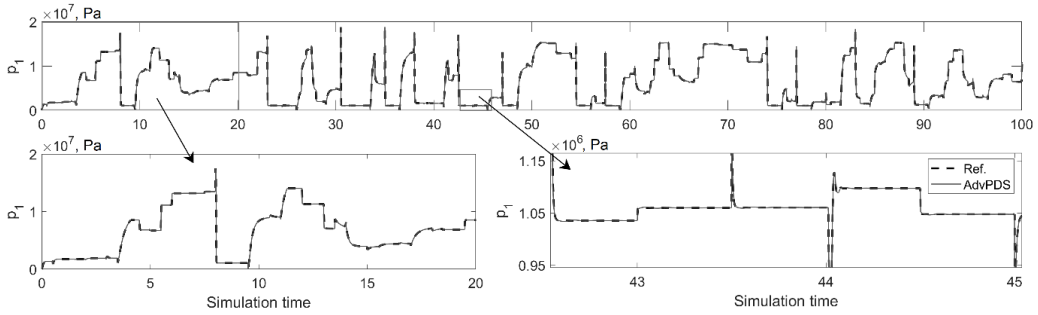

Figure 4.5: Pressure responses of Circuit 2 obtained using the reference solver and AdvPDS (with enlarged areas).

4.4.2 Circuit 4 simulation

Circuit 4 was simulated for 10 seconds with input signals, which are a constant supply pressure of 14 MPa and voltage signal for the directional control valve that varies from −5 to 8 V within a one-second period. The simulation of the system in the presence of the small volume between the pressure compensator and directional control valve using the reference solver was run with the safe integration time step of 10^{-5} s. Such a time step ensured a numerically stable solution for the system. The adaptive criterion values for the AdvPDS under the condition of trade-off between the accuracy and simulation time was experimentally chosen using Circuit 2. In order to verify the applicability of the chosen criterion values to other fluid power circuits, which also include small volumes, Circuit 4 with the AdvPDS was used in another experiment. In this experiment the circuit was simulated 14 times with the different values of the criterion of the AdvPDS, while the simulation times and solution accuracy for the cylinder position piston x_p (against the responses obtained with the reference solver) were measured. The use of AdvPDS for the solution of the system allowed the integration time step to be increased to 10^{-4} seconds for both the main and inner loops. The single criteria value was used in order for the dependency (criterion value/accuracy vs. simulation time) to show itself more clearly. The experimental results are summarised in Table 4.1 and graphically illustrated in Figure 4.6. It can be seen from the figure that the calculation accuracy and simulation time have exponential dependency. Thus, it can be concluded that a larger criterion reduces the simulation time but decreases the calculation accuracy, which is expressed by an increased RMSE. In this case, the criterion equal to 100 can be considered optimal. However, according to the results, the increase in overall accuracy was not significant in contrast with the decrease in simulation time, which in our work is the more advantageous system performance. While also taking into account the solution problems in the low-pressure areas, which were solved by use of a smaller criterion, it became clear that the adaptive criteria 300/20 Pa was the most suitable choice. Consequently, the simulation time was 27.572 s, which is a better result compared with the reference system and with systems having a single convergence criterion.

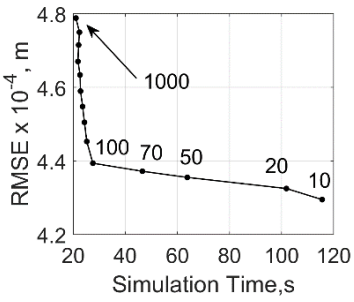

Figure 4.6: Dependency between simulation time and RMSE using AdvPDS with a single criterion.

The response of the pressure p_3 built up in the small volume as well as the cylinder position piston x_p against the responses obtained with the reference solver can be observed in Figure 4.7. The obtained responses of the AdvPDS-based system in the pressure and cylinder piston position were accurate and differed from the reference responses with RMSEs of 1.12×10^5 Pa and only 4.24×10^{-4} m for the pressure and piston position, respectively. The obtained accuracy of the responses was ensured, in particular, by the adaptive criteria, which provided a more precise solution in the low-pressure areas. In Table 4.2, the resulting simulation times for both circuits using reference solver and AdvPDS are presented. The appropriateness of the adaptive criterion chosen was confirmed by a number of experiments that were also carried out with Circuit 4.

Table 4.1: Relationship between criteria value, simulation time, and calculation accuracy for the AdvPDS with a single criterion.

Criteria, Pa	**Simulation time, s**	**RMSE $\times10^{-4}$, m**
10	115.593	4.2950
20	101.847	4.3251
50	63.766	4.3554
70	46.558	4.3720
100	27.546	4.3939
200	25.222	4.4529
300	24.307	4.5051
400	23.609	4.5480
500	22.815	4.5897
600	22.638	4.6339
700	21.829	4.6704
800	22.084	4.7151
900	22.400	4.7496
1000	21.043	4.7881

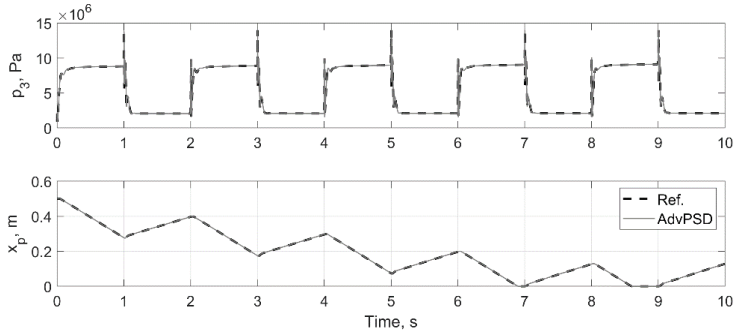

Figure 4.7: Circuit 4 responses in pressure p_3 and cylinder piston position x_p using the reference solver and the AdvPDS.

Table 4.2: Simulation time of Circuit 2 and Circuit 4 using the reference solver and the AdvPDS.

Circuit	Solver	Real - time, s	Time Step, s (main/inner)	Simulation Time	Adaptive Criterion	RMSE w.r.t. Ref.
2	Ref. AdvPDS	100.5 100.5	10^{-6}/- 10^{-4}/10^{-5}	~5 h 147.983 s	- 300/10	- $RMSE_{p1}$ = 1.12×10^{4} Pa
4	Ref. AdvPDS	10 10	10^{-5}/- 10^{-4}/10^{-4}	200.350 s 27.572 s	- 300/20	- $RMSE_{p3}$ = 1.89×10^{5} Pa $RMSE_{xp}$ = 4.24×10^{-4} m

4.4.3 Real-time implementation

To investigate the possibilities of the use of the developed method in real-time and faster than real-time implementations, MATLAB codes for Circuit 4 with the reference solver and the AdvPDS were translated into standalone C code using MATLAB Coder 4.1. Both codes were compiled and run outside MATLAB on a PC with an Intel Core i5-4590 3.30 GHz with 16 GB RAM. As a result, to simulate an interval of 10 s of real time, it took 219 ms for the reference system, whereas for the AdvPDS-based system it took only 47 ms to simulate the same time interval. Thus, the introduction of the developed AdvPDS solver allowed Circuit 4 to be simulated 4.7 times faster in comparison with the reference solver. It should be noted that in our case, both implementations were calculated much faster than real time. However, in virtual prototypes the fluid power system is usually employed in conjunction with mechanical components (i.e. multibody dynamic representation of the mobile machine structure). Thus, the mechanical component should also be calculated at each time step of the real-time simulation. Based on the results, it can be concluded that the use of the AdvPDS for the solution of real-time and faster than real-time systems, which include fluid power components with the small volumes, can be more beneficial than the reference solver application.

4.5 Simulation results using AdvPDS with the improved modified Heun's method

In order to maximise the simulation speed of the developed simulation model (Circuit 2) an implementation and algorithmic perspectives are considered. From an implementation perspective, the compiled C language and procedural programming approach allows for a higher simulation speed in comparison with, for example, MATLAB or Python languages. Thus, the mathematical model of the hydraulic circuit with the two-way flow control valve described in the previous section was implemented in C code.

From an algorithmic perspective, four integration approaches were developed, implemented and compared in solution accuracy and simulation time. In Approach 1 the conventional Runge-Kutta method was employed for the whole system integration. The solution obtained in this way for p_1 was used as a reference solution. For the pressure integration in the three other approaches the AdvPDS was used. The difference between these three approaches was that inside the AdvPDS the Runge-Kutta method (in Approach 2), Euler method (in Approach 3), and the modified Heun's method with improved stability (in Approach 4) were used. It should be noted that the integration outside the AdvPDS loop was still carried out using the Runge-Kutta method.

For the simulation the internal parameters of the AdvPDS, such as the pseudo-volume, two criteria, and maximum iteration time were assigned as recommended in (Malysheva, Ustinov, & Handroos, Computationally Efficient Practical Method for Solving the Dynamics of Fluid Power Circuits in the Presence of Singularities, 2020): $V_{pseudo} = 10^{-3}$ m^3, $p_{1\ tol\ high} = 300$ Pa, $p_{1\ tol\ low} = 20$ Pa, and $t_{max} = 10$ s. Thus, the hydraulic circuit with the two-way flow control valve was simulated for 40.5 s using four described above approaches. As the input, randomly generated signals p_s, U_e and U_d with the respective ranges of 14…20 MPa, 0…10 V and [0,1] V were used. All the signals had periods equal to 1.5 s and were shifted in 0.5 s with respect to each other. For the simulation, the different integration step sizes were chosen for each approach. In each case the step size was as large as the one that provides the numerically stable solution. In addition, inside the AdvPDS loop the local time step was used.

The simulation results for all four approaches are presented in Figure 4.8 and Table 4.3. If we assume that the solution provided by Approach 1 is the reference one, then from the upper plot of Figure 4.8 one can see that in general, all three approaches that use the AdvPDS provided very good approximations of the model solution. In particular, Approaches 2 and 3 were close in accuracy with the errors 0.0539% and 0.0531%, respectively, of the p_1 operating range. Figure 4.8 (bottom plots) reveals that these two approaches provide the solutions that deviated from the reference solution mostly in the transition areas. At the same time, Approach 4 was more accurate with an error as small as 0.0341% of the pressure operating range. Although, the difference in accuracy of the considered approaches was not very significant, as the simulation time varied dramatically from one approach to another. Thus, it took 5.687 s for Approach 1 to simulate 40.5 s of real time, whereas it took 0.115 s and 0.099 s for Approach 2 and Approach 4, respectively. However, Approach 3 was able to handle the same simulation within 0.054 s.

Table 4.3: Simulation results of the four integration approaches.

Approach	Description	Integration step sizes (main/AdvPDS), s	Simulation time (of real time = 40.5 s), ms	$RMSE_{p1}$ (wrt Reference), Pa
1	Reference	10^{-6}/-	5687	-
2	AdvPDS RK4	$10^{-4}/10^{-5}$	115	1.0774×10^{4}
3	AdvPDS Euler	$10^{-4}/10^{-5}$	54	1.0616×10^{4}
4	AdvPDS modif. Heun	$10^{-4}/10^{-4}$	99	0.6816×10^{4}

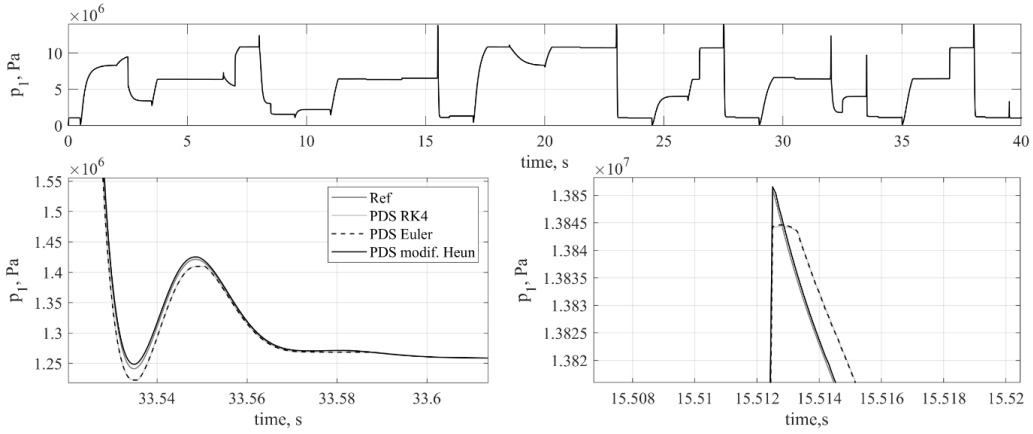

Figure 4.8: Simulation results of the four approaches: Approach 1 – Reference; Approach 2 – AdvPDS with RK4 method; Approach 3 – AdvPDS with Euler method; Approach 4 – AdvPDS with modified Heun's method with improved stability.

Analysing the obtained results, it can be concluded the following. In general, in comparison with the conventional integration approaches applied to the stiff hydraulic model, the employment of AdvPDS allows a noticeable increase in the model's simulation speed, no matter which integration approach is used inside the AdvPDS loop. For the considered numerical example with a single small volume in the circuit, a speed-up of 49.4 with Approach 2, of 57.4 with Approach 4 and of 105.3 with Approach 3 were achieved. However, it is important to note that the speed-up obtained in Approach 3 was due to the reduced number of function calculations needed by the Euler method, whereas the speed-up achieved with Approach 4 was due to the more numerically stable solution provided by the modified Heun's method inside the AdvPDS loop. This fact is also confirmed by the higher error level of Approach 3 in comparison with the error level of Approach 4 (Table 4.3).

5 Conclusions

In this thesis the modelling methods that allow the simulation of the mechatronic machines to accelerate to faster than real-time computational speeds have been proposed. The main findings highlighted in the publications and chapters above are listed as follows:

- Simulation results showed that the application of the computationally efficient dynamic topological formulation of computational complexity *O(n)* (similar to INEF) together with the reasonable simplification of the mechanical component of the simulation model of a mechatronic machine allow the faster than real-time simulation of the acceptable accuracy to be achieved under the condition that the fluid power model of moderate stiffness is used (integration time step should not be less than 10^{-4} s). At the same time, the faster than real-time simulation model of a mechatronic machine of worse computation efficiency can be obtained using commercial software (like MATLAB/Simulink Simscape). However, the integrator of the lower accuracy (similar to the first order Euler method) and obligatory translation to the lower lever programming language (similar to C) should be applied. Thus, the advantage of the direct mathematical modelling built with the use of the computational efficient multibody dynamic method can be seen in the ability of its fine tuning in terms of performance and portability. The implementation of a such model does not depend on any particular programming language, software library, operating system or hardware platform. At the same time, the commercial software modelling can ensure the simplicity of implementation, and even the complex mechatronic systems can in certain cases be used in faster than real-time applications.

- In order to achieve faster than real-time simulation for the fluid power systems, which include features such as stiff differential equations and strong nonlinearities and thus are complex and very time-consuming to solve with numerical integration methods, the RNN with NARX architecture can be used as modelling approach. According to the developed approach, the mathematical model was used for the training data generation. The training data was intended for network training. The pre-processing technique, which concentrates on the temporal information carried by the sequence, was developed and applied to the training data. This technique allowed both the training and simulation processes to speed up. In the considered case of Circuit 3, a calculation speed-up of factor 4.8 was obtained in comparison to the mathematical model-based simulation. Analysing the obtained results, it can be concluded that compared to mathematical model-based simulation, the utilisation of the RNN in combination with the developed pre-processing technique allows simulation speed-up to be obtained at the expense of a minor decrease in accuracy.

- The AdvPDS with adaptive criterion has been proposed for the efficient solution of fluid power systems with singularities originating (in particular) from the presence of small volumes in the system. Based on the results of the experiments performed with two test fluid power circuits, which contained small volumes in their structure, the model for the AdvPDS was formulated. There are two main differences in the AdvPDS in comparison with the classical pseudo-dynamic solver. First, the calculation of the outlet volume flow rate related to the small volume is included in the solver, which allowed the

numerical stability of the solution to be increased. Second, the adaptive convergence criterion is introduced, which allowed the simulation time to be decreased and the calculation accuracy to be increased. Side-by-side simulation results confirmed that the proposed solver is much more efficient in the solution of the fluid power circuits than the conventional method, as well as the classical pseudo-dynamic solver. The main advantage of the proposed solver is that it produces fewer errors than the classical pseudo-dynamic solver with single criteria. In addition, the AdvPDS-based model can be calculated faster than the conventional model of the fluid power circuit with small volumes, due to the possibility of the application of a larger integration time step. Moreover, the AdvPDS solver may be the preferable method in the modelling of more detailed fluid power circuits, especially in such cases when the classical pseudo-dynamic solver may show a numerically unstable and slow response. The described advantages in the solution of the fluid power systems with small volumes of the developed solver allow AdvPDS to be used in simulations of mobile machines in real-time and faster than real-time applications.

- The effect of the three numerical integration methods (Euler, Runge-Kutta of fourth order, and modified Heun's method with improved stability) used inside the AdvPDS loop on the solution efficiency of the stiff mathematical model was studied. The simulation of the fluid power model was carried out using four approaches. The first approach was based on a conventional integration procedure (Rung-Kutta method). The other three approaches included the AdvPDS for the small volume pressure integration and were based on the different numerical integration methods: the Euler method, the Runge-Kutta method of fourth order, and the modified Heun's method with improved numerical stability. The stability of the modified Heun's method was improved by the use of the two-regime orifice model.

- Analysis of the obtained simulation results showed that, in general, the harnessing of the power of the AdvPDS allows numerically stiff hydraulic models to be solved in a very efficient way, ensuring accelerated simulation with high solution accuracy. It was also shown that the simulation speed-up can be achieved not only by the complexity reduction of the numerical integration method inside the AdvPDS (as in the AdvPDS with Euler method), but also by increasing the numerical stability of the employed numerical integration method (as in the AdvPDS with modified Heun's method with improved stability).

The above-mentioned findings and methods can be directly implemented in real-time and faster than real-time simulations of mechatronic machines.

References

Andersson, S., Söderberg, A., & Björklund, S. (2007). Friction models for sliding dry, boundary and mixed lubricated contacts. *Tribology international*, *40*(4), 580-587.

Andrade, F. X., Feucht, M., Haufe, A., & Neukamm, F. (2016). An Incremental Stress State Dependent Damage Model for Ductile Failure Prediction. *International Journal of Fracture*, *200*(1), 127-150.

Baharudin, E., Rouvinen, A., Korkealaakso, P., & Mikkola, A. (2014). Real-time Multibody Application for Tree Harvester Truck Simulator. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, *228*(2), 182-198.

Barabanov, N. E., & Prokhorov, D. V. (2002). Stability Analysis of Discrete-Time Recurrent Neural Networks. *IEEE Transactions on Neural Networks*, *13*(2), 292-303.

Bianchi, F. M., Maiorino, E., Kampffmeyer, M., Rizzi, A., & Jenssen, R. (2017). *Recurrent Neural Networks for Short-Term Load Forecasting.* Springer International Publishing.

Boschert, S., & Rosen, R. (2016). Digital Twin - The Simulation Aspect. In *Mechatronic Futures* (pp. 59-74). Cham: Springer.

Bowns, D. E., & Wang, L. M. (1990). The digital computation of pressures in hydraulic pipes with small volume using an iterative technique. *Proceedings of the Institution of Mechanical Engineers, Part C: Mechanical Engineering Science*, *204*(1), 29-36.

Canudas de Wit, C., Olsson, H., Astrom, K. J., & Lischinsky, P. (1995). A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 419-425.

Craig, J. (2005). *Introduction to Robotics: Mechanics and Control.* New Jersey: Pearson/Prentice Hall.

Demuth, H., Beale, M. H., de Jess, O., & Hagan, M. T. (2014). *Neural Network Design.* Stillwater: Martin Hagan.

Dormand, J., & Prince, P. (1980). A Family of Embedded Runge-Kutta Formulae. *Journal of Computational and Applied Mathematics*, *6*(1), 19-26.

Ellman, A. (1992). *Proposals for utilizing theoretical and experimental methods in modelling two-way cartridge valve circuits.* Doctoral thesis, Tampere University of Technology, Tampere.

Ellman, A., & Piché, R. (1996). A modified orifice flow formula for numerical simulation of fluid power systems. *Fluid power Systems and Technology 1996, Collected papers presented at the 1996 ASME International Mechanical Engineering Congress and Exposition* (pp. 59-62). Atanta: ASME Press.

Ellman, A., & Piché, R. (1999). A two regime orifice flow formula for numerical simulation. *Journal of Dynamic Systems, Measurements, and Control, 121*(4), 721-724.

Esqué, S. (2008). *A new approach for numerical simulations of fluid power circuits using Rosenbrock methods.* Doctoral thesis, Tampere University of Technology, Tampere.

Esqué, S., Raneda, A., & Ellman, A. (2003). Techniques for studying a mobile hydraulic crane in virtual reality. *International Journal of Fluid Power, 4*(2), 25-35.

Ferreira, J. A., Almeida, F. G., Quintas, M. R., & de Oliveira, J. P. (2004). Hybrid models for hardware-in-the-loop simulation of hydraulic systems part 2: Experiments. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 210*(6), 475-486.

Hairer, E., & Wanner, G. (1996). Stability Analysis for Explicit RK Methods. In *Solving Ordinary Differential Equations II. Springer Series in Computational Mathematics, vol 14.* (pp. 15-39). Berlin, Heidelberg: Springer.

Handroos, H., & Vilenius, M. (1991). Flexible semi-empirical models for hydraulic flow control valves. *Journal of Mechanical Design, 113*(3), 232–238.

Hornik, K. (1991). Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks, 4*(2), 251-257.

Jalon, J. G., & Bayo, E. (1994). *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge.* New York: Springer-Verlag New York.

Jelai, M., & Kroll, A. (2003). *Hydraulic Servo-systems: Modelling, Identification and Control.* London: Springer-Verlag .

Karpenko, M., Anderson, J., & Sepehri, N. (2006). Coordination of hydraulic manipulators by reinforcement learning. *American Control Conference. 12*, pp. 3221-3226. Minneapolis: IEEE.

Khalil, W., & Dombre, E. (2002). *Modeling, Identification and Control of Robots.* London: Hermes Penton.

Kiani Oshtorjani, M., Mikkola, A., & Jalali, P. (2019). Numerical Treatment of Singularity in Hydraulic Circuits Using Singular Perturbation Theory. *IEEE/Asme Transactions on Mechatronics, 24*(1), 114-153.

Korkealaakso, P. (2009). *Real-time simulation of mobile and industrial machines using the multibody simulation approach.* Doctoral thesis, Lappeenranta University of Technology, Lappeenranta.

Krus, P. (2011). Robust modelling using bi-lateral delay lines for high speed simulation of complex systems. *The International Symposium on Dynamic Problems of Mechanics, March, 13 to 18, 2011 Maresias Beach Hotel São Sebastião, São Paulo, Brazil*.

Łacny, L. (2012). Modelling of the Dynamics of a Gyroscope Using Artificial Neural Networks. *Journal of Theoretical and Applied Mechanics, 50*(1), 85-97.

Liu, J., Wu, H., Handroos, H., & Haario, H. (2012). Parameter estimation of electrohydraulic servo system using a Markov Chain Monte Carlo method. *Journal of Dynamic Systems, Measurment, and Control, 135*, 011009.

Liu, K., Zhang, C., & Sun, Z. (2019). Independent pressure and flow rate control enabled by hydraulic free piston engine. *IEEE/ASME Transactions on Mechatronics, 24*(3), 1282-1293.

Luh, J. Y., Walker, M. W., & Paul, R. P. (1980). On-Line Computational Scheme for Mechanical Manipulators. *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control, 102*(2), 69-76.

Malysheva, I., Handroos, H., Zhidchenko, V., & Kovartsev, A. (2018). Faster than real-time simulation of a hydraulically actuated log crane. *Global Fluid Power Society PhD Symposium (GFPS)* (pp. 1-6). Samara: IEEE .

Malysheva, J., & Handroos, H. (2020). Fast Calculation of Stiff Hydraulic Models Using the Modified Pseudo-Dynamic Solver. *BATH/ASME 2020 Symposium on Fluid Power and Motion Control.*

Malysheva, J., Li, M., & Handroos, H. (2020). Hydraulic System Modeling with Recurrent Neural Network for the Faster Than Real-Time Simulation. *International Review on Modelling and Simulations, 13*(1), 16-25.

Malysheva, J., Ustinov, S., & Handroos, H. (2020). Computationally Efficient Practical Method for Solving the Dynamics of Fluid Power Circuits in the Presence of Singularities. *IEEE/ASME Transactions on Mechatronics*, 1-1.

Merritt, H. E. (1967). *Hydraulic control systems.* New York: Wiley.

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

Mikkola, A. (1997). Using The Simulation Model for Identification of the Fatigue Parameters of Hydraulically Driven Log Crane. *Journal of Mechanical Design, 123*(1), 125-131.

Mikkola, A., & Handroos, H. (1996). Modelling and simulation of a flexible hydraulic driven log crane. *9th Bath Fluid Power Workshop* (pp. 9-11). Taunton: Wiley-Blackwell.

Ogunmolu, O. P., Gu, X., Jiang, S. B., & Gans, N. R. (2017). Nonlinear systems identification using deep dynamic neural networks. *arXiv:1610.01439 [cs.NE]*, 1-8.

Park, C.-G., Yoo, S., Ahn, H., Kim, J., & Shin, D. (2020). A coupled hydraulic and mechanical system simulation for hydraulic excavators. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 234*(4), 527-549.

Pastorino, R., Cosco, F., Naets, F., Desmet, W., & Cuadrado, J. (2016). Hard real-time multibody simulations using ARM-based embedded systems. *Multibody System Dynamics, 37*(1), 127-143.

Patel, A., & Dunne, J. (2003). NARX Neural Network Modelling of Hydraulic Suspension Dampers for Steady-State And Variable Temperature Operation. *Vehicle System Dynamics, 40*(5), 285-328.

Pedersen, H. (2007). *Automated hydraulic system design and power management in mobile hydraulic applications.* Doctoral thesis, Aalborg University , Aalborg.

Pedersen, M. M., Hansen, M. R., & Ballebye, M. (2010). Developing a Tool Point Control Scheme for a Hydraulic Crane Using Interactive Real-Time Dynamic Simulation. *Modeling, Identification and Control, 31*(4), 133-143.

Petlenkov, E. (2007). *Neural networks based identification and control of nonlinear systems: A NARX model based approach.* Doctoral thesis, Tallinn University of Technology, Tallinn.

Piché, 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*(3), 167-175.

Rahikainen, J., Kiani, M., Sopanen, J., Jalali, P., & Mikkola, A. (2018). Computationally efficient approach for simulation of multibody and hydraulic dynamics. *Mechanism and Machine Theory, 130*, 435-446.

Rojas, R. (1996). *Neural Networks: A Systematic Introduction* . Berlin Heidelberg: Springer-Verlag.

Schram, G., Verhaegen, M., & Krijgsman, A. (1996). System identification with orthogonal basis functions and neural networks. *13th World Congress of IFAC, 29(1)*, pp. 4150 – 4155. San Francisco.

Siegelmann, H. T., Horne, B. G., & Giles, C. L. (1997). Computational Capabilities of Recurrent NARX Neural Networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 27(2)*, pp. 208-215. Haifa: IEEE.

Siegelmann, H., Horne, B. G., & Giles, C. L. (1997). Computational Capabilities of Recurrent NARX Neural Networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 27*(2), 208–215.

Sinha, N. K., Gupta, M. M., & Rao, D. H. (2000). Dynamic neural networks: an overview. *IEEE International Conference on Industrial Technology* (pp. 491-496). Goa: IEEE.

Xu, B., Ding, R., Zhang, J., Sha, L., & Cheng, M. (2016). Multiphysicscoupled modeling: Simulation of the hydraulic-operating mechanism for a sf6 high-voltage circuit breaker. *IEEE/ASME Transactions on Mechatronics, 21*(1), 379–393.

Yu, H., & Wilamowski, B. M. (2011). Levenberg-Marquardt Training. Teoksessa *Industrial Electronics Handbook: Intelligent Systems* (ss. 1-15). CRC Press.

Zheng, Y., Ge, T., & Liu, J. (2015). Kinematics modeling and control simulation for a logging harvester in virtual environments. *Advances in Mechanical Engineering, 7*(10), 1-10.

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

Åman, R. (2011). *Methods and models for accelerating dynamic simulation of fluid power circuits*. Doctoral thesis, Lappenranta University of Technology, Lappeenranta.

Åman, R., & Handroos, H. (2008). Pseudo-dynamic solution of pressures in small volumes in fluid power circuit simulation. *5th Fluid Power Net International PhD Symposium*, (pp. 406-416). Krakow.

Åman, R., & Handroos, H. (2009). Comparison of numerical effectiveness of three methods for modelling 2-way flow control valves. *7th International Conference on Fluid Power Transmission and Control* (pp. 711-715). Hangzhou: Beijing World Publishing Corporation.

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), 945-961.

Åman, R., & Handroos, H. (2010). Optimization of parameters of pseudodynamic solver for real-time simulation of fluid power circuits. *7th International Fluid Power Conference* (ss. 495-507). Aachen: Apprimus.

ACTA UNIVERSITATIS LAPPEENRANTAENSIS

922. KORHONEN, SATU. The journeys of becoming ang being an international entrepreneur: A narrative inquiry of the "I" in international entrepreneurship. 2020. Diss.

923. SIRKIÄ, JUKKA. Leveraging digitalization opportunities to improve the business model. 2020. Diss.

924. SHEMYAKIN, VLADIMIR. Parameter estimation of large-scale chaotic systems. 2020. Diss.

925. AALTONEN, PÄIVI. Exploring novelty in the internationalization process - understanding disruptive events. 2020. Diss.

926. VADANA, IUSTIN. Internationalization of born-digital companies. 2020. Diss.

927. FARFAN OROZCO, FRANCISCO JAVIER. In-depth analysis of the global power infrastructure - Opportunities for sustainable evolution of the power sector. 2020. Diss.

928. KRAINOV, IGOR. Properties of exchange interactions in magnetic semiconductors. 2020. Diss.

929. KARPPANEN, JANNE. Assessing the applicability of low voltage direct current in electricity distribution - Key factors and design aspects. 2020. Diss.

930. NIEMINEN, HARRI. Power-to-methanol via membrane contactor-based CO2 capture and low-temperature chemical synthesis. 2020. Diss.

931. CALDERA, UPEKSHA. The role of renewable energy based seawater reverse osmosis (SWRO) in meeting the global water challenges in the decades to come. 2020. Diss.

932. KIVISTÖ, TIMO. Processes and tools to promote community benefits in public procurement. 2020. Diss.

933. NAQVI, BILAL. Towards aligning security and usability during the system development lifecycle. 2020. Diss.

934. XIN, YAN. Knowledge sharing and reuse in product-service systems with a product lifecycle perspective. 2020. Diss.

935. PALACIN SILVA, VICTORIA. Participation in digital citizen science. 2020. Diss.

936. PUOLAKKA, TIINA. Managing operations in professional organisations – interplay between professionals and managers in court workflow control. 2020. Diss.

937. AHOLA, ANTTI. Stress components and local effects in the fatigue strength assessment of fillet weld joints made of ultra-high-strength steels. 2020. Diss.

938. METSOLA, JAAKKO. Good for wealth or bad for health? Socioemotional wealth in the internationalisation process of family SMEs from a network perspective. 2020. Diss.

939. VELT, HANNES. Entrepreneurial ecosystems and born global start-ups. 2020. Diss.

940. JI, HAIBIAO. Study of key techniques in the vacuum vessel assembly for the future fusion reactor. 2020. Diss.

941. KAZARNIKOV, ALEXEY. Statistical parameter identification of reaction-diffusion systems by Turing patterns. 2020. Diss.

942. SORMUNEN, PETRI. Ecodesign of construction and demolition waste-derived thermoplastic composites. 2020. Diss.

943. MANKONEN, ALEKSI. Fluidized bed combustion and humidified gas turbines as thermal energy conversion processes of the future. 2020. Diss.

944. KIANI OSHTORJANI, MEHRAN. Real-time efficient computational approaches for hydraulic components and particulate energy systems. 2020. Diss.

945. PEKKANEN, TIIA-LOTTA. What constrains the sustainability of our day-to-day consumption? A multi-epistemological inquiry into culture and institutions. 2021. Diss.

946. NASIRI, MINA. Performance management in digital transformation: a sustainability performance approach. 2021. Diss.

947. BRESOLIN, BIANCA MARIA. Synthesis and performance of metal halide perovskites as new visible light photocatalysts. 2021. Diss.

948. PÖYHÖNEN, SANTERI. Variable-speed-drive-based monitoring and diagnostic methods for pump, compressor, and fan systems. 2021. Diss.

949. ZENG, HUABIN. Continuous electrochemical activation of peroxydisulfate mediated by single-electron shuttle. 2021. Diss.

950. SPRINGER, SEBASTIAN. Bayesian inference by informative Gaussian features of the data. 2021. Diss.

951. SOBOLEVA, EKATERINA. Microscopy investigation of the surface of some modern magnetic materials. 2021. Diss.

952. MOHAMMADI ASL, REZA. Improved state observers and robust controllers for non-linear systems with special emphasis on robotic manipulators and electro-hydraulic servo systems. 2021. Diss.

953. VIANNA NETO, MÁRCIO RIBEIRO. Synthesis and optimization of Kraft process evaporator plants. 2021. Diss.

954. MUJKIC, ZLATAN. Sustainable development and optimization of supply chains. 2021. Diss.

955. LYYTIKÄINEN, JOHANNA. Interaction and barrier properties of nanocellulose and hydrophobically modified ethyl(hydroxyethyl)cellulose films and coatings. 2021. Diss.

956. NGUYEN, HOANG SI HUY. Model based design of reactor-separator processes for the production of oligosaccharides with a controlled degree of polymerization. 2021. Diss.

957. IMMONEN, HEIKKI. Application of object-process methodology in the study of entrepreneurship programs in higher education. 2021. Diss.

958. KÄRKKÄINEN, HANNU. Analysis of theory and methodology used in determination of electric motor drive system losses and efficiency. 2021. Diss.

959. KIM, HEESOO. Effects of unbalanced magnetic pull on rotordynamics of electric machines. 2021. Diss.

LUT University

ISBN 978-952-335-652-8
ISBN 978-952-335-653-5 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491
Lappeenranta 2021